

Universidad de Valladolid
Departamento de Electrónica

Simulación atomística de procesos para Microelectrónica



Tesis Doctoral
Ignacio Martín Bragado
2004

Lista de contenidos condensada

| | |
|---|-----------|
| Índice general | v |
| 1. Introducción | 1 |
| I Simulaciones de Dinámica Molecular | 9 |
| 2. Dinámica Molecular | 11 |
| 3. Optimización de clusters usando algoritmos genéticos | 31 |
| II El método de Monte Carlo | 47 |
| 4. Los métodos de Monte Carlo | 49 |
| 5. El método de Monte Carlo con red | 59 |
| 6. La aproximación de colisiones binarias | 71 |
| III El método de Monte Carlo cinético (sin red) | 77 |
| 7. Procesos en Silicio: Mecanismos | 79 |
| 8. Implementación atomística de los modelos físicos. | 101 |
| 9. Especies implementadas en DADOS | 183 |
| 10. Comprobación de los modelos de DADOS | 217 |
| 11. Resultados recientes | 249 |

| | |
|---|------------|
| IV Conclusiones | 313 |
| 12. Conclusiones | 315 |
| V Apéndices | 321 |
| A. Deducción de expresiones utilizadas en DADOS | 323 |
| B. Arquitectura del proyecto UVAS | 345 |
| VI Índices y bibliografía | 401 |
| Bibliografía | 403 |
| Índice de figuras | 419 |
| Índice de tablas | 435 |
| Índice de parámetros de DADOS | 438 |
| Índice general | 442 |

Índice general

| | |
|--|-----------|
| Índice general | v |
| 1. Introducción | 1 |
| 1.1. Introducción | 3 |
| 1.2. Fabricando un transistor NMOS: Breve revisión del proceso de fabricación monolítico | 3 |
| 1.3. Simulación de los procesos de fabricación | 4 |
| 1.4. Estructura de la tesis | 7 |
| | |
| I Simulaciones de Dinámica Molecular | 9 |
| | |
| 2. Dinámica Molecular | 11 |
| 2.1. Introducción | 13 |
| 2.2. Las ecuaciones de movimiento | 14 |
| 2.3. Esquema de integración | 14 |
| 2.3.1. El integrador de Verlet | 15 |
| 2.4. Potencial | 16 |
| 2.4.1. Potencial de Stillinger–Weber | 17 |
| 2.5. Configuraciones microcanónica y canónica | 19 |
| 2.5.1. Configuración canónica | 19 |
| 2.6. Detalles de la implementación del código | 19 |
| 2.6.1. DynAtomSet: El objeto principal de MD | 20 |
| 2.6.2. AtomicConfiguration: Un objeto para gestionar configuraciones atomísticas | 21 |
| 2.6.3. Integrator | 22 |
| 2.6.4. Potential | 23 |
| 2.6.5. Locator | 24 |
| 2.7. Un ejemplo de aplicación: La migración del I_2 y su captura en superficie | 25 |
| 2.8. Conclusiones | 27 |

| | |
|---|-----------|
| 3. Optimización de clusters usando algoritmos genéticos | 31 |
| 3.1. Introducción | 33 |
| 3.2. La técnica de los algoritmos genéticos | 33 |
| 3.2.1. Un ejemplo como introducción | 35 |
| 3.3. Optimización de clusters aislados | 37 |
| 3.3.1. Introducción al modelo | 38 |
| 3.3.2. El operador mutación | 38 |
| 3.3.3. El operador de cruce. | 39 |
| 3.3.4. Reemplazando los individuos viejos o menos aptos | 41 |
| 3.4. Resultados | 42 |
| 3.5. Conclusiones | 42 |
| | |
| II El método de Monte Carlo | 47 |
| | |
| 4. Los métodos de Monte Carlo | 49 |
| 4.1. Introducción | 51 |
| 4.2. Algunos ejemplos básicos | 51 |
| 4.2.1. Una dimensión | 51 |
| 4.2.2. Dos dimensiones | 52 |
| 4.2.3. Convergencia | 53 |
| 4.3. Aplicaciones del método MC a la física estadística | 54 |
| 4.4. Diferentes esquemas MC | 55 |
| 4.4.1. Metropolis MC | 55 |
| 4.4.2. Monte Carlo cinético (kMC) | 56 |
| 4.4.3. La aproximación de colisiones binarias (BCA) | 56 |
| 4.5. Conclusiones | 57 |
| | |
| 5. El método de Monte Carlo con red | 59 |
| 5.1. Introducción | 61 |
| 5.2. Difusión atomística para el modelo LMC | 62 |
| 5.3. Detalles de la implementación | 64 |
| 5.3.1. Diferentes conjuntos de átomos | 65 |
| 5.3.2. Gestionando el tiempo | 65 |
| 5.4. Un ejemplo: Recocido de cristales con bordes afilados. | 67 |
| 5.5. Conclusiones | 68 |
| | |
| 6. La aproximación de colisiones binarias | 71 |
| 6.1. Introducción | 73 |
| 6.2. La aproximación de colisiones binarias | 74 |
| 6.3. Potenciales interatómicos | 75 |
| 6.4. Pérdidas inelásticas de energía | 75 |
| 6.5. Conclusión | 76 |

| | | |
|------------|--|------------|
| III | El método de Monte Carlo cinético (sin red) | 77 |
| 7. | Procesos en Silicio: Mecanismos | 79 |
| 7.1. | Introducción | 81 |
| 7.2. | Breve revisión sobre la difusión y los defectos en Silicio | 82 |
| 7.3. | Defectos propios del silicio | 83 |
| 7.4. | Los defectos puntuales I y V | 83 |
| 7.4.1. | Configuración atómica | 83 |
| 7.4.2. | Difusividades y concentraciones en equilibrio | 85 |
| 7.4.3. | Efectos debidos al nivel de Fermi | 85 |
| 7.5. | Defectos extensos de I s y V s | 86 |
| 7.5.1. | Clusters de intersticiales | 86 |
| 7.5.2. | Los defectos de intersticiales $\{311\}$ | 87 |
| 7.5.3. | Discos de dislocación | 88 |
| 7.5.4. | Defectos extendidos de V en el Silicio | 90 |
| 7.5.5. | Voids (huecos) | 90 |
| 7.6. | Amorfización y recristalización | 91 |
| 7.6.1. | Regiones desordenadas (amorphous pockets) | 92 |
| 7.6.2. | Recristalización | 94 |
| 7.7. | Intercaras y volumen | 95 |
| 7.7.1. | Condiciones de equilibrio | 95 |
| 7.7.2. | Condiciones de no equilibrio: Oxidación y nitridación | 95 |
| 7.8. | Átomos de impurezas | 97 |
| 7.8.1. | Procesos de atrapamiento y reemplazo (kick-out) | 97 |
| 7.8.2. | Formación y precipitación de impurezas | 98 |
| 7.9. | Procesamiento de dispositivos nanoscópicos | 98 |
| 8. | Implementación atomística de los modelos físicos. | 101 |
| 8.1. | Introducción al funcionamiento de DADOS. | 103 |
| 8.1.1. | Unidades | 104 |
| 8.1.2. | Partículas y defectos puntuales y extensos en DADOS | 105 |
| 8.1.3. | Tratamiento del espacio en DADOS | 105 |
| 8.1.4. | Distintos materiales en DADOS | 106 |
| 8.1.5. | Gestión del tiempo en DADOS | 107 |
| 8.2. | Defectos puntuales I y V | 109 |
| 8.2.1. | Migración | 109 |
| 8.2.2. | Interacciones | 110 |
| 8.3. | Defectos extendidos de I s y V s | 114 |
| 8.3.1. | Pequeños clusters de I y V | 114 |
| 8.3.2. | Defectos $\{311\}$ | 118 |
| 8.3.3. | Bucles de dislocación | 121 |
| 8.3.4. | Void | 125 |
| 8.4. | Acumulación de dañado | 126 |

| | | |
|-----------|---|------------|
| 8.4.1. | Pockets amorfos | 127 |
| 8.4.2. | Modelo de amorfización | 131 |
| 8.4.3. | Regiones y defectos amorfos | 133 |
| 8.5. | Recristalización | 134 |
| 8.5.1. | Suavizando las zonas amorfas | 136 |
| 8.5.2. | Depositando el exceso de defectos puntuales | 136 |
| 8.5.3. | Las impurezas durante la recristalización | 137 |
| 8.6. | Impurezas | 139 |
| 8.6.1. | Las impurezas como defectos puntuales | 139 |
| 8.6.2. | Clusters de impurezas | 143 |
| 8.7. | Intercaras entre materiales | 149 |
| 8.7.1. | Modelo de intercaras para los defectos puntuales propios del Silicio | 149 |
| 8.7.2. | Impurezas | 153 |
| 8.7.3. | Oxidación y nitridación | 157 |
| 8.8. | Modelo de carga | 157 |
| 8.8.1. | Hipótesis | 158 |
| 8.8.2. | Energías de formación para especies cargadas | 159 |
| 8.8.3. | Dependencia con la temperatura | 160 |
| 8.8.4. | Atracciones y repulsiones de carga | 161 |
| 8.8.5. | Probabilidades y estados de carga | 162 |
| 8.8.6. | Deriva eléctrica | 165 |
| 8.8.7. | Implementación | 166 |
| 8.8.8. | Difusividades efectivas de dopantes | 168 |
| 8.8.9. | Dependencia de la solubilidad con el nivel de Fermi | 173 |
| 8.8.10. | Estrechamiento del gap | 174 |
| 8.9. | Comandos de DADOS | 175 |
| 8.9.1. | Insertando partículas en la simulación | 175 |
| 8.9.2. | Lectura de perfiles | 176 |
| 8.9.3. | Implantación iónica | 177 |
| 8.9.4. | Anneal (recocido o difusión) | 178 |
| 8.9.5. | Generación de salidas | 179 |
| 9. | Especies implementadas en DADOS | 183 |
| 9.1. | El intersticial propio del silicio | 185 |
| 9.1.1. | Defectos puntuales del I | 185 |
| 9.1.2. | Cómo se introducen I s en la simulación | 185 |
| 9.1.3. | Interacciones del I | 185 |
| 9.1.4. | Disolución de “pockets” amorfos | 187 |
| 9.2. | Vacante | 187 |
| 9.2.1. | Defectos puntuales de V | 187 |
| 9.2.2. | Cómo se introducen V s en la simulación | 187 |
| 9.2.3. | Interacciones con V | 188 |

| | | |
|--------|--|-----|
| 9.2.4. | Disolución de pockets amorfos | 189 |
| 9.3. | Boro | 189 |
| 9.3.1. | Defectos puntuales con B | 189 |
| 9.3.2. | Introduciendo B en la simulación | 189 |
| 9.3.3. | Interacciones del boro | 190 |
| 9.3.4. | Disolución de pockets amorfos | 194 |
| 9.3.5. | El B en los procesos de recristalización | 194 |
| 9.3.6. | Salto largo del boro | 194 |
| 9.3.7. | Clusters de impureza de B | 195 |
| 9.4. | Arsénico | 196 |
| 9.4.1. | Defectos puntuales con As | 196 |
| 9.4.2. | Introduciendo As en la simulación | 196 |
| 9.4.3. | Interacciones del As | 197 |
| 9.4.4. | Recristalización y As | 200 |
| 9.4.5. | Clusters de impurezas de As | 200 |
| 9.5. | Flúor | 201 |
| 9.5.1. | Defectos puntuales de F | 201 |
| 9.5.2. | Introduciendo F en la simulación | 202 |
| 9.5.3. | Interacciones del F | 202 |
| 9.5.4. | Recristalización y F | 203 |
| 9.5.5. | Clusters de impurezas con F | 203 |
| 9.6. | Carbono | 204 |
| 9.6.1. | Defectos puntuales con C | 205 |
| 9.6.2. | Introduciendo C en la simulación | 205 |
| 9.6.3. | Interacciones con C | 205 |
| 9.6.4. | Clusters de impurezas de C | 206 |
| 9.7. | Fósforo | 207 |
| 9.7.1. | Defectos puntuales con P | 207 |
| 9.7.2. | Introduciendo P en la simulación | 207 |
| 9.7.3. | Interacciones del fósforo | 207 |
| 9.7.4. | El fósforo durante la recristalización | 210 |
| 9.7.5. | Clusters de impurezas con P | 210 |
| 9.8. | Oxígeno | 211 |
| 9.8.1. | Defectos puntuales con O | 211 |
| 9.8.2. | Introduciendo O en la simulación | 212 |
| 9.8.3. | Interacciones del oxígeno | 212 |
| 9.8.4. | Cluster de impurezas con O | 213 |
| 9.9. | Indio | 213 |
| 9.9.1. | Defectos puntuales con In | 213 |
| 9.9.2. | Introduciendo indio en la simulación | 214 |
| 9.9.3. | Interacciones del indio | 214 |

| | |
|---|------------|
| 10. Comprobación de los modelos de DADOS | 217 |
| 10.1. Introducción | 219 |
| 10.2. Concentraciones y difusividades en equilibrio | 219 |
| 10.2.1. I y V | 219 |
| 10.2.2. Dopantes | 221 |
| 10.3. Defectos extendidos | 224 |
| 10.3.1. Clusters de intersticiales y evolución temporal de la super- saturación de I_s | 224 |
| 10.3.2. Disolución de los $\{311\}$ s | 225 |
| 10.3.3. Bucles de dislocación | 227 |
| 10.4. El modelo de dañado | 229 |
| 10.4.1. Acumulación de dañado | 229 |
| 10.4.2. Amorfización y recristalización | 231 |
| 10.4.3. Interacción del dañado con los dopantes | 232 |
| 10.5. Activación y desactivación de dopantes | 236 |
| 10.5.1. Difusión del boro y formación de clusters de boro intersti- cial (BIC)s | 236 |
| 10.5.2. Desactivación del arsénico | 238 |
| 10.5.3. Implantación y recocido del boro | 239 |
| 10.5.4. Implantación y recocido del arsénico | 240 |
| 10.6. Intercaras entre materiales | 241 |
| 10.6.1. Longitud de recombinación y TED del boro | 241 |
| 10.6.2. Solubilidad de dopantes | 242 |
| 10.7. Modelo de carga | 244 |
| 10.7.1. Difusión extrínseca | 244 |
| 10.7.2. Redistribución de dopantes en las uniones PN | 246 |
| 10.8. Conclusiones | 248 |
| 11. Resultados recientes | 249 |
| 11.1. El di-intersticial móvil del Silicio | 251 |
| 11.1.1. Introducción | 251 |
| 11.1.2. Comprobando los distintos modelos. | 253 |
| 11.1.3. Modelo propuesto | 257 |
| 11.1.4. Conclusiones | 261 |
| 11.2. Cálculos para la movilidad del I_2 | 262 |
| 11.2.1. Difusividad del I_2 | 262 |
| 11.2.2. Equilibrio y emisión desde la superficie | 262 |
| 11.2.3. Comparación entre $D(I_2)[I_2]^*$ y $D(I)[I]^*$ | 263 |
| 11.2.4. Relación entre el I_2 y los clusters de intersticiales | 264 |
| 11.2.5. Emisión de I_2 por parte de los clusters de impurezas | 266 |
| 11.2.6. El I_2 y su supersaturación | 267 |
| 11.2.7. Difusión del boro y movilidad del I_2 | 267 |
| 11.2.8. Disolución los $\{311\}$ e I_2 | 269 |

| | |
|---|-----|
| 11.3. Importancia de la correlación de los pares IV | 270 |
| 11.3.1. Introducción | 270 |
| 11.3.2. Modelo empleado | 271 |
| 11.3.3. Temas a tratar | 272 |
| 11.3.4. Simulación empleada | 272 |
| 11.3.5. Correlación espacial de los pares <i>IV</i> | 272 |
| 11.3.6. Implantaciones pequeñas | 274 |
| 11.3.7. Importancia de los clusters de vacantes | 274 |
| 11.3.8. Presencia de impurezas | 276 |
| 11.3.9. Conclusiones | 277 |
| 11.4. Modelizado físico de los efectos del nivel de Fermi | 278 |
| 11.4.1. Introducción | 278 |
| 11.4.2. Modelos físicos | 279 |
| 11.4.3. Implementación | 287 |
| 11.4.4. Ejemplos de simulación | 289 |
| 11.4.5. Conclusiones | 290 |
| 11.5. Modelizado físico del dañado, amorfización y recristalización | 291 |
| 11.5.1. Introducción | 291 |
| 11.5.2. Descripción del modelo | 292 |
| 11.5.3. Resultados | 295 |
| 11.5.4. Conclusiones | 298 |
| 11.6. Un modelo físico y extenso para el arsénico en Silicio | 301 |
| 11.6.1. Introducción | 301 |
| 11.6.2. El modelo | 302 |
| 11.6.3. Discusión de los resultados | 308 |
| 11.6.4. Conclusiones | 310 |

IV Conclusiones 313

| | |
|---|-----|
| 12. Conclusiones 315 | |
| 12.1. Construcción de un entorno de simulación | 317 |
| 12.2. Desarrollo de diversos simuladores atomísticos | 317 |
| 12.3. Desarrollo y puesta a punto del simulador KMC DADOS | 317 |
| 12.4. Estudios con resultados originales | 319 |
| 12.5. Conclusión | 320 |

V Apéndices 321

| | |
|---|-----|
| A. Deducción de expresiones utilizadas en DADOS 323 | |
| A.1. Ecuaciones de difusividad | 325 |
| A.1.1. Definiciones de flujo y difusividad en una dimensión | 325 |

| | |
|--|------------|
| A.1.2. Ley de Fick en una dimensión | 325 |
| A.2. Impurezas | 326 |
| A.2.1. Concentración de impurezas en equilibrio | 326 |
| A.2.2. Cálculo de $[A_i]$ en condiciones de casi-equilibrio | 328 |
| A.2.3. Prefactor y energía de activación de $[A_i]^*$ | 329 |
| A.2.4. Difusión de A | 329 |
| A.2.5. Un ejemplo: energías de difusión del boro | 330 |
| A.3. La superficie | 331 |
| A.3.1. La distancia de recombinación | 331 |
| A.3.2. Coeficiente de captura superficial | 332 |
| A.3.3. Concentración y capacidad de transporte de I | 333 |
| A.3.4. Supersaturación | 334 |
| A.4. Equilibrio en el volumen interno | 334 |
| A.5. Cálculos del modelo de carga | 335 |
| A.5.1. Energías de formación | 335 |
| A.6. Implementación del modelo de carga | 336 |
| A.6.1. Actualización estática de carga | 337 |
| A.6.2. Actualización dinámica de carga | 338 |
| A.7. Un ejemplo: Difusión del boro y estados de carga | 339 |
| A.7.1. Difusividad del boro y saltos largos | 342 |
| A.8. Energías y prefactores de activación efectivos para gráficas de Arrhenius | 342 |
| A.9. Notación | 343 |
| B. Arquitectura del proyecto UVAS | 345 |
| B.1. Introducción | 347 |
| B.2. Propósitos | 347 |
| B.2.1. Propósitos del entorno | 347 |
| B.2.2. Propósitos de los simuladores | 349 |
| B.3. Implementación | 349 |
| B.4. Opciones globales | 353 |
| B.5. Gestionando los comandos: El “Script Wizard” | 357 |
| B.5.1. Introducción | 357 |
| B.5.2. La clase de comandos básica: <code>UVCmd</code> | 358 |
| B.5.3. Comandos de entrada de parámetros: La clase <code>CmdIO</code> | 361 |
| B.5.4. Comandos de ejecución: <code>CmdExec</code> | 367 |
| B.5.5. Una clase que contiene a todos los comandos: <code>UVScript</code> | 370 |
| B.6. Los datos de salida: <code>DataBrowser</code> | 371 |
| B.6.1. Introducción | 371 |
| B.6.2. Clases generales para manipular los datos: <code>UVData</code> | 372 |
| B.7. Gestionando las gráficas | 385 |
| B.7.1. Introducción | 385 |
| B.7.2. Gráficos en dos dimensiones: <code>UVGraph2D</code> | 386 |

| | |
|--|-----|
| B.7.3. Gráficas en tres dimensiones: <code>UVGraph3D</code> | 388 |
| B.7.4. Clase común para las gráficas en 2 y 3 dimensiones: <code>UVGraph391</code> | |
| B.7.5. Gestionando un área de dibujo rectangular: <code>UVDrawRgn</code> . . | 392 |
| B.7.6. Un sistema de dibujo genérico: <code>GenericDrawer</code> | 393 |
| B.8. Extrayendo los datos en un modo no gráfico | 396 |

| | |
|----------------------------------|------------|
| VI Índices y bibliografía | 401 |
| Bibliografía | 403 |
| Índice de figuras | 419 |
| Índice de tablas | 435 |
| Índice de parámetros de DADOS | 438 |
| Índice general | 442 |

A Espe,
mi compañera de viaje
en este misterio
que llamamos vida.

Agradecimientos

No es fácil incluir aquí una lista completa de todas las personas a las que debería agradecer la feliz conclusión de esta tesis, y tampoco me resulta tarea trivial el cómo ordenarlas y priorizarlas, pero en cualquier caso debo empezar agradeciendo al director del Departamento de Electrónica, Dr. D. Juan Barbolla, el haberme dado la oportunidad de trabajar en este Departamento y este grupo, y después al Dr. D. Martín Jaraíz por haber dirigido mi tesis y haberme ofrecido su tiempo y conocimientos durante estos cuatro años. Tampoco quiero olvidarme de agradecer al Dr. D. Emiliano Rubio, sin cuya llamada de teléfono no habría existido nunca esta tesis.

Después quiero dirigirme en particular a los compañeros de mi grupo de investigación. Ya he nombrado a Martín Jaraíz y José Emiliano Rubio, y quiero también nombrar a Ruth Pinacho y Pedro Castrillo, por haber peleado juntos para que DADOS funcione, y de los que puedo decir que han sido excelentes compañeros de trabajo. Sin el apoyo de Ruth, el optimismo de Emiliano, las energías de Pedro y la experiencia y vista al frente de Martín, nada de esto habría sido posible. Han sido excelentes compañeros, es cierto, y también quiero decir que han sido amigos fuera del trabajo, dedicándome su tiempo, su comedor ¡y hasta su furgoneta!.

Posteriormente tengo que acordarme de mis compañeros de Departamento, D. Luis Bailón, D. Salvador Dueñas, D^a Elena Castán, D. Luis A. Marqués, D. Luis Quintanilla, D. Jesús Arias, D. Jesús M. Hernández, D^a Lourdes Pelaz, D. José V. Antón, D^a Lourdes Enríquez, que me han ofrecido su apoyo, comprensión, conocimientos y tiempo siempre que la ocasión lo ha requerido: gracias.

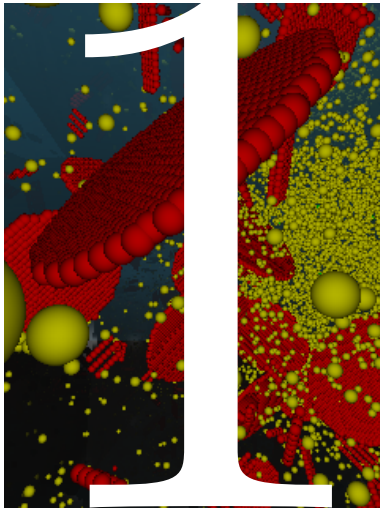
Dejo para el final al “grupo de jóvenes”, Jacinto, David, María, Héctor, Pedro, Caroline: ánimo, que ya va quedando menos para vuestras tesis, y a Ángel Vegas, para que aún tenga que soportarme muchos lunes más.

Me gustaría dedicar también una breve línea a Victor Moroz y Norbert Strec-ker, por darme la oportunidad de conseguir experiencia en el Silicon Valley, a George Gilmer, que me enseñó San Jose, y a Merlyne de Souza, quien me introdujo en el mundo del I_2 .

Por último quiero agradecer a mis familiares, padres, hermanos, tíos: sin ellos, para bien y para mal, no sería lo que soy, a mis amigos, que tan estoicamente me han aguantado, a mi mujer, que ha ido siguiendo el progreso de esta tesis

fielmente reflejado en los cambios de mi humor y estrés.

Y ahora, creo que es menester acabar mencionando a todos aquellos que no he nombrado en esta lista, ya algo larga, y así mismo al sufrido lector que esté enfrentándose a esta tesis: a todos vosotros, muchas gracias.



Introducción

| | |
|--|---|
| 1.1. Introducción | 3 |
| 1.2. Fabricando un transistor NMOS: Breve revisión del proceso de fabricación monolítico | 3 |
| 1.3. Simulación de los procesos de fabricación | 4 |
| 1.4. Estructura de la tesis | 7 |

El número de técnicas que existen para simular los diferentes procesos de fabricación de dispositivos desde la oblea de silicio hasta la caracterización y comprobación de las curvas I-V de un circuito integrado (IC) es enorme. Por ello, en este trabajo nos vamos a centrar únicamente en los procesos necesarios para fabricar los dispositivos, es decir, los pasos de implantación iónica, depósito de capas delgadas y recocido de la muestra, prestando una especial atención, sobre todo, a la técnica de Monte Carlo cinético que permite simular la difusión de los dopantes en silicio durante el recocido, sin por ello menoscabar el aspecto global del proceso como un todo.

▷ 1.1

Introducción

En los últimos años la producción de dispositivos microelectrónicos se ha convertido en algo cada vez más importante para la sociedad, abarcando campos tan diversos como el hogar, el tiempo de ocio, o las investigaciones científicas. Sin duda alguna podemos afirmar que la microelectrónica tiene un impacto enorme en la vida diaria y en la economía mundial del mundo en que vivimos.

Comenzando con el primer transistor fabricado en los años 50 y terminando con las modernas CPUs que incluyen millones de dispositivos integrados en un tamaño minúsculo, la microelectrónica se ha convertido en una industria que mueve miles de millones de euros cada año, y fabrica componentes más rápidos, precisos y pequeños cada día que pasa.

Los procesos físicos relacionados con la fabricación microelectrónica son bastante complejos, pero no obstante, la necesidad de fabricar circuitos baratos, rápidos y flexibles, exige su claro entendimiento. Para ello, la industria microelectrónica requiere de avanzadas herramientas de simulación, que incluyan los modelos y procesos involucrados en la fabricación de dispositivos microelectrónicos, para así poder predecir cómo va a funcionar un circuito antes de fabricarlo físicamente.

▷ 1.2

Fabricando un transistor NMOS: Breve revisión del proceso de fabricación monolítico

Un diagrama del procesado posible para la fabricación de un MOS de canal n básico se representa en la figura 1.1. Dicho proceso es como sigue:

1. Se comienza limpiando una oblea de tipo p , y posteriormente se genera una fina capa de dióxido de silicio (SiO_2) por oxidación, que proteja la superficie del silicio. Por encima de esta capa se deposita, mediante un depósito de vapor químico a baja temperatura (LPCVD), otra capa muy fina (80 nm) de nitruro de silicio (Si_3N_4). A continuación, mediante una primera máscara, se delimita el área activa del transistor. Posteriormente se elimina el nitruro y óxido de todas partes, excepto en las regiones en las que la máscara protegía el nitruro, que será donde se forme el transistor.
2. Posteriormente se implanta boro y se realiza un proceso de oxidación local (LOCOS). Esto crea dos regiones p^+ que aíslan el transistor (especialmente

las regiones de fuente y drenador que habrá que crear más tarde). El nitruro restante se utiliza como máscara de implantación y oxidación.

3. Se eliminan las capas de nitruro y óxido y se vuelve a crecer una nueva capa de óxido, que formará el óxido de puerta, necesario para el transistor NMOS.
4. Se deposita polisilicio sobre todo el transistor, el cual se quitará posteriormente excepto en la zona de la puerta. En este momento tenemos ya creada la puerta con su estructura típica de Metal (donde en vez de metal tenemos polisilicio, que también es un buen conductor) Óxido (el recientemente depositado SiO_2) y Semiconductor (la oblea p). A continuación se implanta arsénico para formar las regiones n^+ de fuente y drenador, donde la puerta y el SiO_2 que estaban del proceso LOCOS previo actúan a modo de máscara para la implantación. Esta implantación se hace más profunda a la vez que se activa el dopante y se elimina el dañado mediante un recocido (proceso de difusión).
5. Se vuelve a depositar óxido en la superficie y con otra máscara se definen las ventanas de contacto. Una vez que se han creado los huecos del contacto se deposita metal (típicamente aluminio), por evaporación o *sputtering*.
6. Por último se utiliza otra máscara que defina las pistas de conexión que serán cortadas en el metal y se deposita una capa de pasivación que garantice el correcto aislamiento del dispositivo entero, tras lo cual con una máscara final se aseguran las conexiones externas al transistor.

Se puede encontrar una información más completa sobre la fabricación VLSI y esta tecnología en general en los libros (Jaeger, 1997; Plummer et al., 2000; Sze, 1983).

▷ 1.3

Simulación de los procesos de fabricación

Como se ha visto en el apartado anterior, la fabricación de un transistor NMOS requiere, si bien no se limita a, *etching*, implantación iónica, difusión, depósito de capas delgadas, metalización y litografía. Estos procesos tienen un nivel distinto de complejidad de realización y simulación, y algunos de ellos están, a día de hoy, perfectamente caracterizados y conocidos. Otros, sin embargo, suponen grandes retos para la tecnología actual.

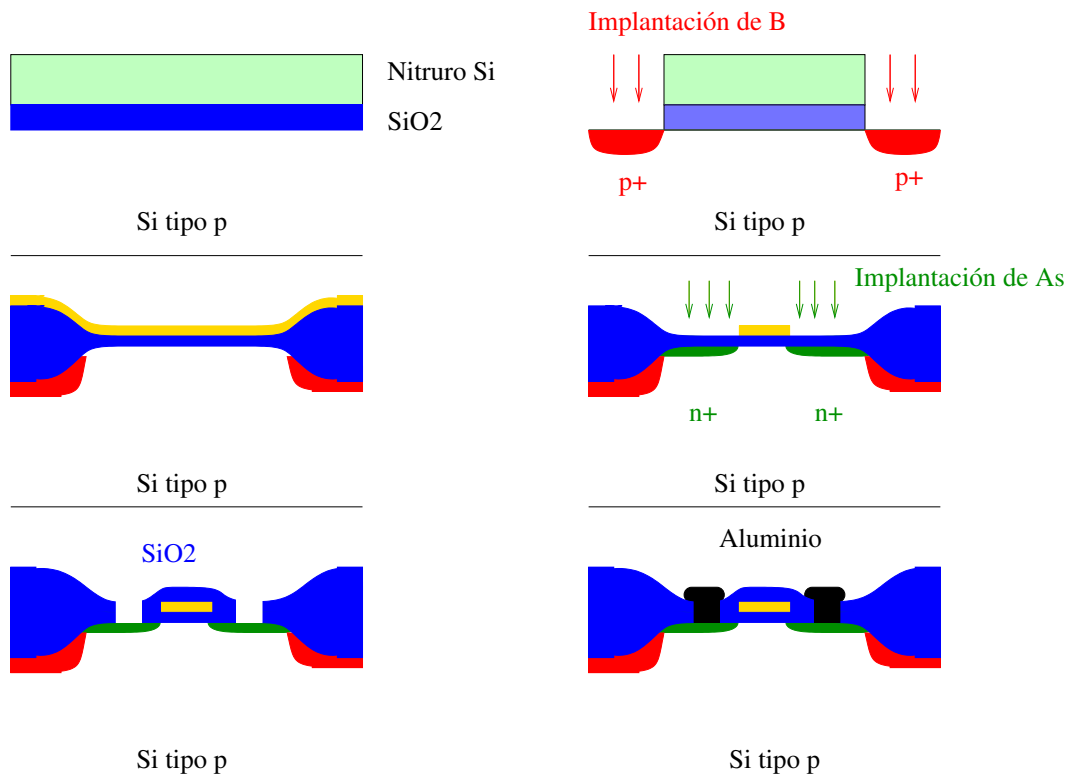


Figura 1.1: Pasos de fabricación de un transistor NMOS. Su descripción se encuentra en el texto.

Siguiendo esta línea, *THE INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS, 2003* (Plan internacional para la tecnología de semiconductores)(Several, 2003), en su sección *Grand challenges* (grandes retos) afirma que aun hay dos de estos retos para resolver hasta el año 2009 en el campo del modelizado y simulación, que son:

1. Modelizado de circuitos de alta frecuencia para aplicaciones de 5 a 40 GHz.
2. Modelizado de procesos para estructuras nanométricas.

De estos dos retos, el último está relacionado con los procesos de fabricación tecnológica y, por tanto, con esta tesis. Concretamente, el *International Roadmap* lo describe como sigue:

MODELADO DE PROCESOS PARA ESTRUCTURAS NANOMÉTRICAS [MODELIZADO y SIMULACIÓN]

El modelado de procesos para estructuras nanométricas supone la piedra angular para la predicción de resultados en la fabricación de dispositivos. En cierta medida, se solapa con el difícil reto de la capacidad de simular CMOS nanométricos que, no obstante, también incluye la simulación de materiales y dispositivos. Mucho más importante y estimulante en el área de simulado de procesos es el modelizado de la formación de uniones poco profundas, que comienzan con las implantaciones a muy baja energía, y se centra especialmente en el **recocido térmico y difusión de dopantes**. Debido a la exigencia de muy reducidos fenómenos térmicos requerida para estas uniones poco profundas, dichos procesos son muy transitorios, y están gobernados por la difusión y reacción de átomos y defectos de dopantes, y especialmente por el clustering dinámico de éstos. El **daño de implantación, amorfización, recristalización y silicidación** tienen que ser también simulados de forma precisa. A la vista de la necesidad de incrementar la movilidad de los portadores en el canal, el modelizado del estrés y su influencia en la difusión y activación se ha vuelto vital, especialmente para silicio estresado, SiGe y para estructuras SOI. La investigación de modelos, calibración y evaluación, así como los procesos de caracterización, requieren de numerosas actividades experimentales y de amplios progresos en la forma de medir los dopantes, defectos y estrés, sobre todo para medidas en dos y tres dimensiones.

FRONT-END PROCESS MODELING FOR NANOMETER STRUCTURES [MODELING AND SIMULATION]

Front-end process modeling for nanometer structures is the key challenge for the prediction of result from device fabrication. It overlaps to some

*extent with the Difficult Challenge–Ultimate nanoscale CMOS simulation capability, which however also includes materials and device simulation. Most important and challenging in the area of front-end process modeling is the modeling of ultra-shallow junction formation, which starts from very low energy implant and especially focuses on the **thermal annealing and diffusion of dopants**. Due to the strongly reduced thermal budgets needed for shallow junctions that process is highly transient and is governed by the diffusion and reaction of dopant atoms and defects, and especially by the dynamics of clusters of these two. **Implantation damage, amorphization, recrystallisation, and silicidation** must be accurately simulated. In view of the need to increase carrier mobilities in the channel, the modeling of stress and strain and their influence on diffusion and activation has become vital, especially for strained silicon, SiGe, and for SOI structures. Model development, calibration and evaluation as well as process characterization require numerous experimental activities and large progress in the metrology for dopants, defects and stress, especially regarding two- and three-dimensional measurements.*

En el desarrollo de esta tesis doctoral nos hemos centrado en los procesos resaltados en negrita en el párrafo anterior.

▷ 1.4

Estructura de la tesis

A lo largo de esta tesis, y con diferente profundidad y extensión, se han ido estudiando diferentes técnicas de simulación. Se empieza con una introducción a la técnica de Dinámica Molecular (MD) en el capítulo 2 para poder después aplicar esta técnica junto con Algoritmos Genéticos (GA) a la investigación de estructuras óptimas de intersticiales en el capítulo 3. Estas técnicas se utilizan para dilucidar los mecanismos de difusión del I_2 y su recombinación en la superficie, así como para calcular la formación de clusters de átomos de silicio en el vacío. La dinámica molecular se utiliza con éxito para caracterizar los mecanismos de difusión, formación de dañado y otros, pero tiene como problema estar limitada a escalas de tiempo pequeñas y pequeña cantidad de partículas.

Posteriormente, se introduce de una forma general el método de Monte Carlo (MC) en el capítulo 4. Este método aporta descripciones para escalas temporales mucho más largas que las que aporta la MD, pero en general, necesita partir de datos que se pueden obtener de experimentos o simulaciones de ab-initio o MD. Hay diferentes variaciones del método MC:

- El capítulo 5 explica como implementar y usar Monte Carlo con red (LMC)

para simular el depósito y/o crecimiento de capas delgadas metálicas.

- En el capítulo 6 se comenta muy brevemente la aproximación de colisiones binarias (BCA), método estándar para simular la implantación iónica.
- Por último se trata muy profundamente el método de Monte Carlo cinético sin red (kMC), que supone una herramienta “state-of-the-art” para simular los procesos de difusión de dopantes, recocido del dañado, formación de defectos puntuales y extendidos, amorfización y recristalización. La mayor parte de esta tesis está dedicada al estudio de este método, y a la exposición del funcionamiento del simulador kMC DADOS.

Para estudiar el método kMC se ha centrado la tesis en diferentes aspectos:

- El capítulo 7 da un repaso general y teórico a los fenómenos físicos que aparecen durante el procesado del silicio.
- La descripción de todos los modelos implementados en DADOS y los parámetros usados se encuentra en el capítulo 8.
- El capítulo 9 incluye los modelos partículas y parámetros usados de forma ordenada para cada uno de los dopantes concretos utilizados.
- Se muestra una colección exhaustiva de comprobaciones y ejemplos de buen funcionamiento del simulador en el capítulo 10.
- Por último, el capítulo 11 incluye los más recientes y punteros estudios realizados con DADOS.

La tesis finaliza en el capítulo 12 con la exposición de las conclusiones.

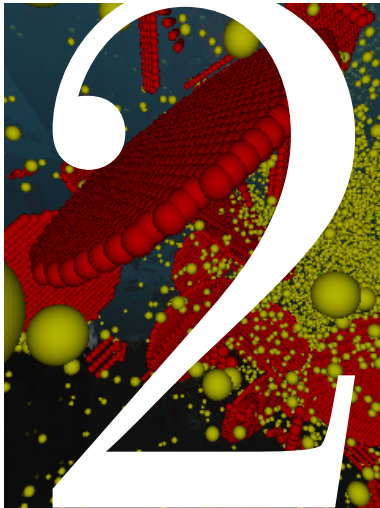
En los apéndices se ofrece información adicional sobre los modelos de DADOS, con demostraciones y deducciones matemáticas extra en el apéndice A.

El apéndice B se dedica al Simulador Atomístico de la Universidad de Valladolid, UVAS (*University of Valladolid Atomistic Simulator*), que supone la realización de un esfuerzo para conseguir integrar todas las técnicas de simulación anteriores en un único entorno de simulación. UVAS ha sido diseñado para ser fácil de usar, mantener y portar a otras plataformas, pero también para ofrecer un entorno gráfico común a todas las simulaciones atomísticas que permita estudiar y analizar los datos de éstas aunque usen diferentes técnicas.

Antes de terminar, decir que puesto que el objetivo principal de esta tesis es el estudio del desarrollo y modelizado de un simulador atomístico de procesos en silicio, no debe extrañar que aparezcan algunos pequeños fragmentos de código en C++ diseminados en la tesis, cuyo objetivo es ayudar a clarificar y dar, a la par, una visión más profunda de la estructura y funcionamiento interno de estos simuladores.

Parte I

**Simulaciones de Dinámica
Molecular**



Dinámica Molecular

| | |
|--|----|
| 2.1. Introducción . . . | 13 |
| 2.2. Las ecuaciones de movimiento . . . | 14 |
| 2.3. Esquema de integración | 14 |
| 2.4. Potencial | 16 |
| 2.5. Configuraciones microcanónica y canónica | 19 |
| 2.6. Detalles de la implementación del código | 19 |
| 2.7. Un ejemplo de aplicación: La migración del I_2 y su captura en superficie | 25 |
| 2.8. Conclusiones . . . | 27 |

El punto de partida de la Dinámica Molecular (MD) consiste en una descripción conocida del sistema físico bajo consideración. Dicha descripción se puede expresar directamente como las ecuaciones de movimiento de Newton, o puede ser también una expresión lagrangiana o hamiltoniana, de la que se obtengan las ecuaciones de movimiento aplicando los formalismos apropiados a cada caso. De esta manera el método de la Dinámica Molecular calcula las propiedades de un sistema resolviendo numéricamente sus ecuaciones de movimiento. En estos sistemas, debido a una conservación de la energía inherente a la técnica MD, se suele emplear una configuración microcanónica.

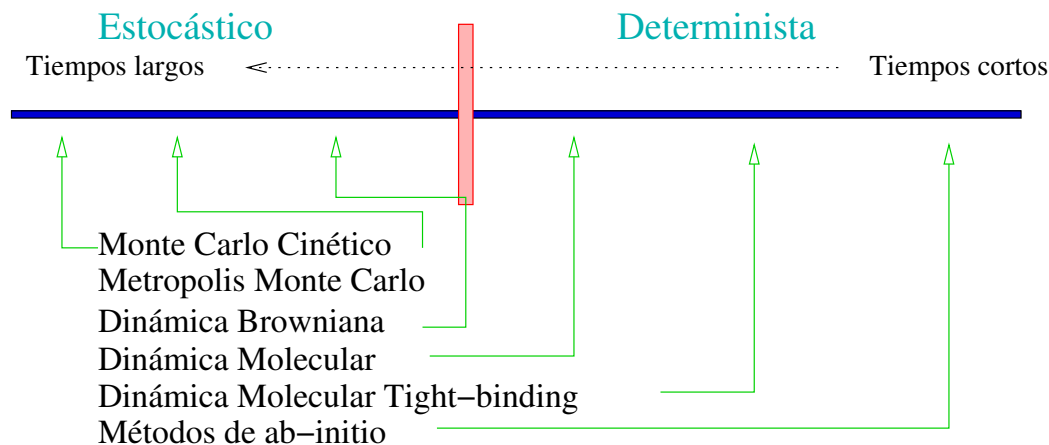


Figura 2.1: Diferentes métodos de simulación, teniendo en cuenta sus rangos de tiempos típicos y su ámbito determinista o estocástico.

▷ 2.1

Introducción

La dinámica molecular (MD) es un sistema de simulación determinista. En los sistemas en los que se aplica todos los grados de libertad son tomados en consideración explícitamente. La dinámica molecular se basa en las ecuaciones de Newton o Lagrange dentro del marco de la mecánica clásica, y posteriormente se intentan calcular las magnitudes físicas del sistema bajo consideración, como pueden ser la temperatura, energía potencial, etc. Dichas ecuaciones serán discretizadas para así generar una trayectoria en el espacio de fases, donde se puedan calcular las propiedades requeridas. En este método la trayectoria en el espacio de fases se va generando “paso a paso”, a través de un esquema iterativo, en el que, en cada paso, se simula una cantidad de tiempo Δt , que típicamente se encuentra en torno a 1×10^{-15} s.

Este valor típico de incremento de tiempo sitúa a la dinámica molecular en un rango de simulación que está representado comparativamente con otros métodos en la figura 2.1, junto con una indicación de su situación en el ámbito de la simulación determinista o estocástica.

En este capítulo se describe brevemente el método de dinámica molecular: comenzaremos con una descripción de las ecuaciones de movimiento para proseguir con los diferentes esquemas de integración de éstas y los diferentes potenciales aplicables, prosiguiendo con una introducción a las configuraciones microcanónica y canónica. Terminaremos con un ejemplo: La aplicación de la técnica MD al silicio cristalino para mostrar como los di-intersticiales propios del Si (I_2) migran y son atrapados por la superficie libre del cristal.

▷ 2.2

Las ecuaciones de movimiento

En general, el sistema a tratar tendrá N partículas y estará descrito por el siguiente hamiltoniano:

$$\mathcal{H} = \frac{1}{2} \sum_i \frac{p_i^2}{m_i} + \sum_{i < j} u(r_{ij}) \quad (2.1)$$

donde r_{ij} es la distancia entre las partículas i y j . Supondremos que la energía potencial será la suma de todos los potenciales interatómicos, donde se necesita limitar esta suma a $i < j$ para evitar duplicar los potenciales entre i, j y j, i . Con el anterior hamiltoniano se derivan las ecuaciones del movimiento como:

$$m_i \frac{d\vec{r}_i}{dt} = \vec{p}_i, \quad \frac{d\vec{p}_i}{dt} = \sum_{i < j} \vec{F}(r_{ij}).$$

Vamos a restringir el problema a las propiedades en un volumen con una densidad concreta ρ , de manera que se introduce un volumen, denominado celda de simulación, que generalmente es un simple cubo con $L^3 = V$, o bien un octaedro cuyos lados miden $L_x * L_y * L_z = V$.

Para reducir el efecto de las superficies se suelen imponer condiciones periódicas de contorno, es decir, se repite la celdilla básica a izquierda y derecha, arriba y abajo, etc., de manera que cada componente de los vectores de posición pueda representarse con un número entre 0 y L_i , (siendo L_i la longitud máxima en la dirección considerada), de tal forma que existan partículas en todas las posiciones $r_i + \vec{n}L_i$, donde \vec{n} es un vector de números enteros.

Al introducir las condiciones periódicas, puesto que los potenciales presentan un radio de corte (r_c) a partir del cual no hay interacciones entre partículas, es decir, potencial cero), se evita que una partícula interaccione con su homóloga en otra celdilla eligiendo un L_i tal que $r_c < L_i/2$.

▷ 2.3

Esquema de integración

Se denomina integrador a aquel algoritmo cuyo fin es resolver el conjunto de ecuaciones diferenciales del movimiento. Debido a la naturaleza numérica del proceso, los integradores introducen ciertos errores en dicho proceso. Para generar esta solución, los integradores almacenan un número determinado de estados pasados, y calculan el nuevo estado “presente” a partir de ellos (ver figura 2.2).

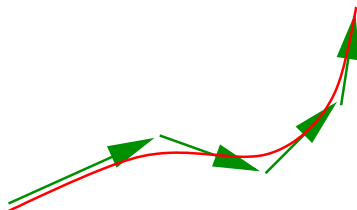


Figura 2.2: Un integrador resuelve las ecuaciones de movimiento calculando el paso siguiente en función de la trayectoria pasada y las fuerzas en el momento presente (y/o pasado).

La resolución de un integrador es mejorable, siempre a costa de consumir más recursos de memoria y CPU.

En nuestro caso se ha implementado el integrador de Verlet (Heermann, 1990), que es bastante sencillo a la vez que rápido y suficientemente preciso.

▷ 2.3.1

El integrador de Verlet

Desde un punto de vista estrictamente matemático, el problema de la dinámica molecular consiste en resolver un conjunto de ecuaciones diferenciales dadas por:

$$\frac{d^2\vec{r}_i}{dt^2} = \frac{1}{m_i}\vec{F}_i.$$

Analíticamente, las soluciones de un sistema diferencial de segundo orden se obtienen integrando dos veces desde un tiempo cero hasta t , para obtener las velocidades en primer lugar, y después las posiciones. Esto requiere las posiciones y velocidades iniciales (como constantes de integración), y produce que el sistema se vaya moviendo a lo largo de una trayectoria de energía constante en el espacio de fases.

Para resolver las ecuaciones diferenciales numéricamente se utiliza la siguiente discretización para la derivada segunda:

$$\frac{d^2\vec{r}_i}{dt^2} = \frac{1}{\Delta t^2}(\vec{r}_i(t + \Delta t) - 2\vec{r}_i(t) + \vec{r}_i(t - \Delta t)) = \frac{\vec{F}_i(t)}{m_i},$$

que suministra un algoritmo con el que obtener las posiciones de las partículas en un tiempo $t + \Delta t$ a partir de las posiciones inmediatamente anteriores en t y $t - \Delta t$ y las fuerzas en t . Resolviendo para $t + \Delta t$ tenemos:

$$\vec{r}_i(t + \Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t - \Delta t) + \vec{F}_i(t)\frac{\Delta t^2}{m_i}.$$

Y, teniendo en cuenta que las velocidades son

$$\vec{v}_i = \frac{d\vec{r}_i}{dt}$$

se puede escribir que:

$$\vec{v}_i(t) = \frac{\vec{r}_i(t + \Delta t) - \vec{r}_i(t - \Delta t)}{2\Delta t}.$$

Por último, si denominamos a $t = n \times \Delta t = t_n$, $\vec{r}_i^n = \vec{r}_i(t_n)$ y $\vec{F}_i^n = \vec{F}_i(t_n)$, y reorganizamos las fórmulas, se llega a la siguiente forma del algoritmo de Verlet, que es la que utilizamos en nuestro simulador:

$$\vec{r}_i^{n+1} = \vec{r}_i^n + \Delta t \times \vec{v}_i^n + \frac{\Delta t^2}{2m_i} F_i^n \quad (2.2)$$

$$\vec{v}_i^{n+1} = \vec{v}_i^n + \frac{\Delta t(\vec{F}_i^{n+1} + \vec{F}_i^n)}{2m_i}. \quad (2.3)$$

$$(2.4)$$

▷ 2.4

Potencial

Para implementar un algoritmo MD se necesita un potencial que describa el sistema. El potencial puede ser de índole fundamental, como la interacción de Coulomb, o bien un potencial empírico. Los potenciales fundamentales dan una descripción extraordinariamente precisa y realista del sistema, pero generalmente resultan bastante complejos de implementar a no ser que se hagan aproximaciones, las cuales reducen, a veces de manera bastante significativa, la exactitud del resultado final. A veces, incluso con estas aproximaciones, este tipo de potenciales resultan extraordinariamente costosos, hablando en términos de memoria y ciclos de CPU consumidos.

Por otra parte, los potenciales empíricos están limitados a ciertos sistemas, y a veces la descripción de estos sistemas no es completa o muy precisa, pero generalmente la descripción resulta lo suficientemente profunda para ser útil. A cambio los potenciales empíricos resultan muy rápidos y sencillos de implementar. Su problema principal radica en que usan parámetros que deben ser ajustados muy cuidadosamente, y que la utilidad de estos parámetros suele encontrarse limitada al sistema para el que fueron extraídos.

En las últimas décadas se han desarrollado varios potenciales para el silicio cristalino, de los cuales nosotros hemos implementado y vamos a utilizar el denominado potencial de Stillinger–Weber (Stillinger and Weber, 1985).

▷ 2.4.1

Potencial de Stillinger–Weber

Stillinger y Weber propusieron un potencial para las interacciones silicio–silicio. Supusieron que el potencial que describe las interacciones entre las distintas partículas puede ser dividido en diferentes contribuciones para uno, dos, tres... cuerpos, de forma que para un sistema con N cuerpos:

$$\Phi(1, 2, \dots, N) = \sum_i v_1(i) + \sum_{i < j} v_2(i, j) + \dots + \sum_{i < j < k} v_3(i, j, k) + \dots + v_n(1, 2, \dots, N)$$

El potencial v_1 es debido a fuerzas externas, y vamos a suponer que en nuestro caso tales fuerzas no existen. Respecto a las diferentes contribuciones Stillinger y Weber despreciaron las mantenidas entre cuatro o más cuerpos, y se limitaron a las de dos y tres.

Para desarrollar las contribuciones v_2 y v_3 se introducen unas unidades para la distancia y la energía llamadas ϵ y σ

$$v_2(r_{ij}) = \epsilon f_2(r_{ij}/\sigma)$$

$$v_3(\vec{r}_i, \vec{r}_j, \vec{r}_k) = \epsilon f_3(\vec{r}_i/\sigma, \vec{r}_j/\sigma, \vec{r}_k/\sigma)$$

donde ϵ y σ han sido escogidas para que el mínimo de f_2 sea -1 y $\frac{d}{dt} f_2(2^{1/6}) = 0$. (Stillinger and Weber, 1985)

Con esta elección, la función del potencial normalizado para dos cuerpos v_2 es

$$f_2(r) = \begin{cases} A(Br^{-p} - r^{-q}) * \exp[(r - a)^{-1}], & r < a \\ 0, & r \geq a \end{cases}, \quad (2.5)$$

donde A, B, p, q y a son mayores que 0. Además se impone un radio de corte en esta función para $r = a$, con lo que el potencial será nulo para distancias mayores.

La contribución a tres cuerpos resulta ser:

$$f_3(\vec{r}_i, \vec{r}_j, \vec{r}_k) = h(r_{ij}, r_{ik}, \theta_{jik}) + h(r_{ji}, r_{jk}, \theta_{ikj}) + h(r_{ki}, r_{kj}, \theta_{ikj}) \quad (2.6)$$

donde θ_{jik} es el ángulo comprendido entre \vec{r}_{ij} y \vec{r}_{ik} , y la función h se define a continuación:

$$h(r_{ij}, r_{ik}, \theta_{jik}) = \lambda (\gamma(r_{ij} - a)^{-1} + \gamma(r_{ik} - a)^{-1}) \left(\cos(\theta_{jik}) + \frac{1}{3} \right)^2, \quad r_{ij}, r_{ik} < a \quad (2.7)$$

si r_{ij} o r_{ik} son más grandes que a , entonces $h = 0$ (al igual que para v_2). El ángulo θ_t característico para las estructuras tipo diamante es

$$\cos(\theta_t) = -\frac{1}{3}$$

de tal forma que este término tenderá a situar los átomos en una estructura tipo diamante, característica de la estructura cristalina del silicio.

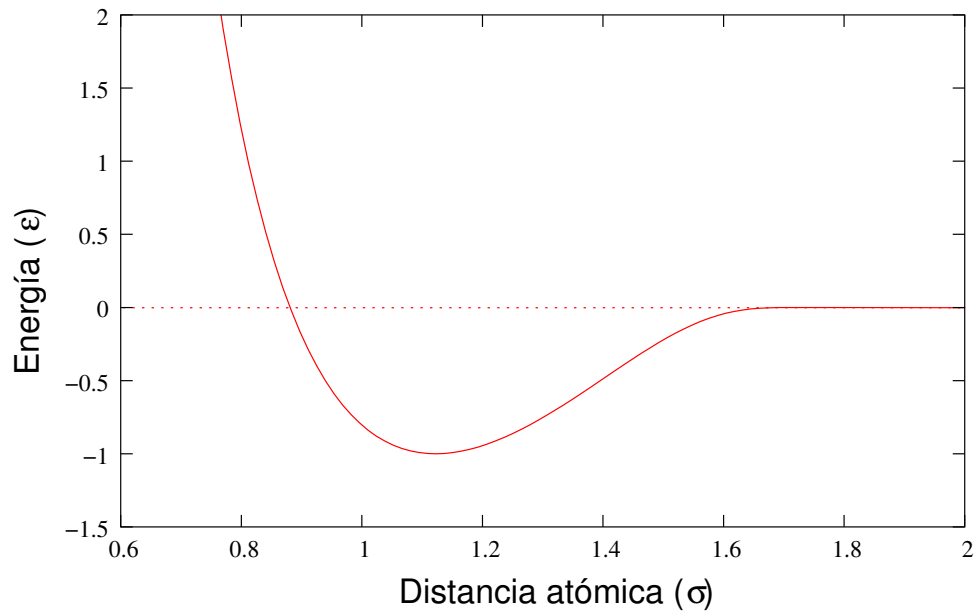


Figura 2.3: Contribución a dos cuerpos del potencial de Stillinger–Weber.

Stillinger y Weber, ajustando las constantes de su potencial para obtener una conducta adecuada para éste, publicaron el siguiente conjunto de valores: (Stillinger and Weber, 1985)

$$\begin{aligned}
 A &= 7,049556277 & B &= 0,6022245584 \\
 p &= 4 & q &= 0 \\
 a &= 1,80 & \lambda &= 21,0 \\
 \gamma &= 1,20 & &
 \end{aligned} \quad (2.8)$$

Además, para obtener una longitud de enlace correcta, y el adecuado punto de fusión, σ y ϵ resultaron ser:

$$\begin{aligned}
 \sigma &= 0,20951 \text{ nm} \\
 \epsilon &= 50 \text{ Kcal/mol} = 2,167 \text{ eV}
 \end{aligned} \quad (2.9)$$

Una representación de la función f_2 para este potencial se muestra en la gráfica 2.3. El potencial es mínimo para $r_{\min} = 1,1225\sigma$ con una energía por átomo de -2ϵ .

▷ 2.5

Configuraciones microcanónica y canónica

Hemos introducido la técnica MD como un mecanismo numérico para resolver las ecuaciones newtonianas o hamiltonianas dentro de un marco de dinámica clásica. En dicho marco la energía total del sistema se conserva, por lo que las simulaciones MD se realizan de manera natural en una configuración microcanónica.

▷ 2.5.1

Configuración canónica

Anteriormente hemos visto como la dinámica molecular resuelve las ecuaciones de movimiento numéricamente, dando lugar a un sistema conservativo en el cual la energía total es constante. Ahora bien, en muchas circunstancias es mucho más deseable investigar las propiedades de un sistema a temperatura constante, o bien poner el sistema en conjunto a cierta temperatura y ver como evoluciona después.

Una forma natural de conseguir lo anterior es manipular la energía cinética del sistema, que es proporcional a la temperatura, y escalarla de vez en cuando para asegurarse que el sistema evoluciona a la temperatura pedida, multiplicando todas las velocidades por un factor β .

La forma de obtener el factor de escalado β para un sistema con $3N$ grados de libertad es:

$$\beta = \sqrt{\frac{(3N - 4)k_B T}{\sum_i m v_i^2}},$$

donde hemos quitado cuatro grados de libertad, siendo uno la limitación de temperatura constante, y otros tres al forzar a que el momento lineal total del sistema sea 0.

▷ 2.6

Detalles de la implementación del código

En UVAS se ha implementado el módulo de dinámica molecular siguiendo las anteriores directrices. Han sido generados tres grandes clases: `Integrator`, `Potential` y `DynAtomSet`, que resulta ser una clase de índole general que manipula las anteriores. Además de éstas ha sido necesario añadir una `AtomicConfiguration` definida para ayudar a agrupar las configuraciones atómicas.

El integrador de Verlet y el potencial de Stillinger–Weber no han sido defi-

nidos directamente en UVAS, sino que más bien se ha definido primero una clase genérica que manipule los “integradores en general” y otra para los “potenciales en general”, de las que el integrador de Verlet y el potencial de Stillinger–Weber son derivaciones concretas. Con ello se pretende construir un código lo más flexible y general posible, donde el uso de diferentes potenciales y/o integradores sea sencillo.

▷ 2.6.1

DynAtomSet: El objeto principal de MD

`DynAtomSet` es la clase que UVAS asocia con las simulaciones de dinámica molecular. Se encarga de coordinar el potencial y el integrador pertinentes para llevar a buen puerto la simulación completa. Así, una `DynAtomSet` necesita de un objeto `Potential`, un `Integrator` y una `AtomicConfiguration` para comenzar a funcionar. Además necesita de un objeto especial (de uso interno) al que denominamos `Locator`, para gestionar las coordenadas espaciales de manera eficiente.

Este objeto `DynAtomSet` es capaz de:

Recocer, es decir, ejecutar la simulación un tiempo o número de iteraciones determinado a energía total constante. Para hacer esto `DynAtomSet` llama a `Potential` para actualizar las fuerzas en cada átomo en el momento presente y después invoca a `Integrator` para calcular las nuevas posiciones. Este proceso es repetido una y otra vez.

Poner a una temperatura, cosa que hace escalando la velocidad de los átomos integrantes de la simulación, tal y como se describe en 2.5.1, y haciendo una serie de recocidos entre dichos escalados.

Relajar. “Relajar” una simulación de dinámica molecular consiste en que `DynAtomSet` intente minimizar en lo posible la energía potencial del sistema. Para ello `DynAtomSet` permite a cada átomo que se mueva hasta que encuentre un mínimo local de energía potencial, momento en que para dicho átomo (pone su velocidad a 0). `DynAtomSet` repite este proceso hasta que ha pasado una cantidad determinada de iteraciones o todos los átomos están en un mínimo local de energía.

Calcular la energía potencial. `DynAtomSet` calcula y devuelve la energía potencial total del sistema en consideración, limitándose a preguntar al objeto `Potential` adecuado.

Más detalles en sec. B, pág. 347.

`DynAtomSet` está derivado de `UVObject`, que aporta toda la interfaz necesaria para manipular el objeto usando la estructura gráfica del programa UVAS. La estructura interna de `DynAtomSet` se representa en la figura 2.4. Como ya se ha comentado, este objeto contiene a su vez otros objetos `Potential`, `Integrator`

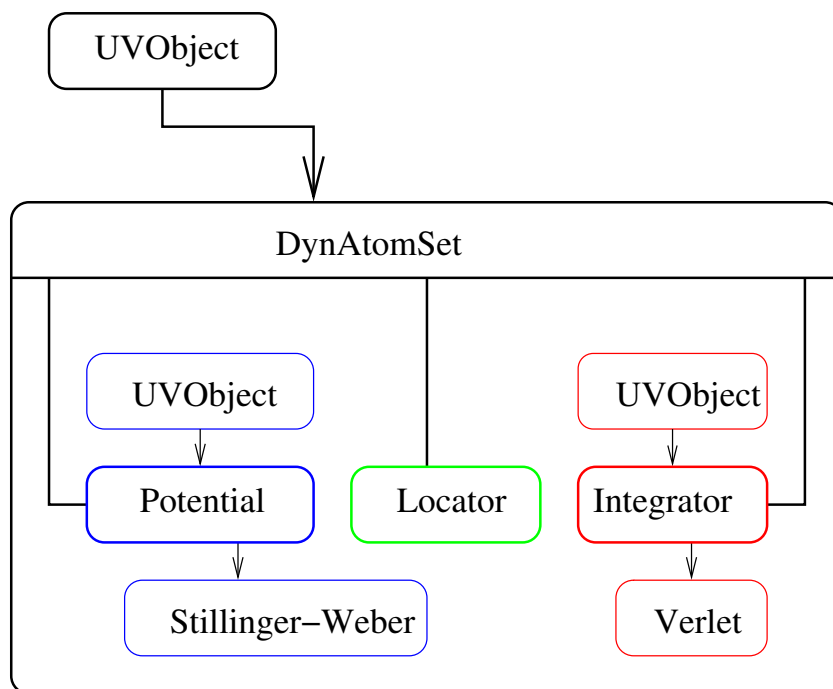


Figura 2.4: Estructura interna de DynAtomSet.

y un Locator. El Integrator y el Potential van a ser descritos en las secciones 2.6.3 y 2.6.4 respectivamente, y el Locator es una herramienta importante cuya misión es gestionar las posiciones de los átomos en el espacio, y será ligeramente descrita en el apartado 2.6.5.

El objeto que queda por describir, AtomicConfiguration, se limita a ser un contenedor de un conjunto de átomos. Hemos creado esta clase para conseguir leer y grabar conjuntos de átomos desde ficheros de manera sencilla, y su implementación será tratada en el apartado 2.6.2.

▷ 2.6.2 AtomicConfiguration: Un objeto para gestionar configuraciones atómicas

Un objeto AtomicConfiguration es un conjunto de átomos, de los cuales queremos guardar su posición y tipo de elemento (silicio, aluminio...). Estos objetos son bastante útiles para pasar este tipo de información desde unos módulos a otros, así como para poder fácilmente grabar y obtener átomos de ficheros. No obstante, la variedad de información asociada a los átomos es enorme, ya que un simulador de dinámica molecular necesitará las fuerzas interatómicas, pero en cambio uno de Monte Carlo requeriría más bien las frecuencias de salto, etc. Por ello AtomicConfiguration se limita a almacenar la información básica y común de los átomos: su elemento y posición. La información extra se pierde cuando

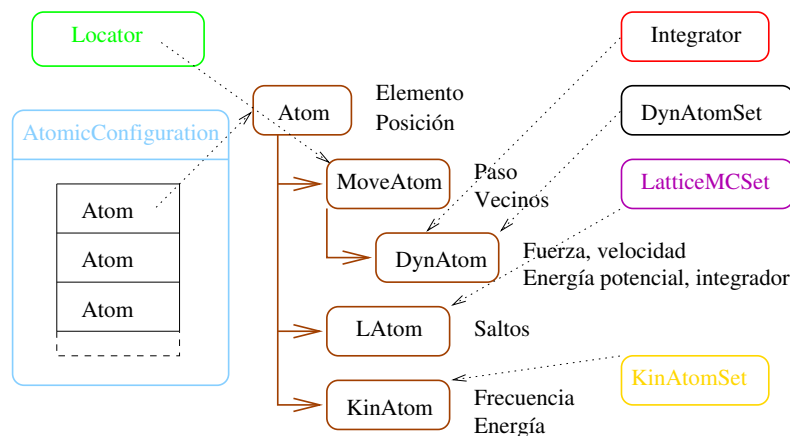


Figura 2.5: Diferentes tipos información atómica en UVAS.

se graba una **AtomicConfiguration**. Esta información extra es gestionada por UVAS a través de la clase base **Atom** (que sólo contiene posición y elemento) y una serie de clases derivadas de ella que añaden lo que cada simulación en particular necesite, siendo competencia de cada módulo de simulación el grabar dicha información extra. Un gráfico que muestra esta “jerarquía de átomos” se encuentra en la figura 2.5.

Estas estructuras **AtomicConfiguration** son muy útiles para intercambio entre módulos por las anteriores razones. Uno puede generar una configuración específica con el simulador MD, pero al grabarla como **AtomicConfiguration** la información particular de MD se pierde. Posteriormente, si esta configuración es leída por un simulador MC (por ejemplo), éste puede añadir la información específica que necesite, por lo que **AtomicConfiguration** se convierte en la herramienta idónea para dichas transformaciones.

► 2.6.3

Integrator

Integrator es una clase base que contiene los elementos comunes de todos los integradores posibles. Se define como:

```
// Integrator.h: interface for the Integrator class.
//
#if !defined(_INTEGRATOR_H_)
#define _INTEGRATOR_H_

class DynAtom;
class Potential;
class Locator;
#include "../Common/UVTypes.h"
#include "../Common/UVObject.h"
#include "../Common/CmdConstructor.h"
```



```

class Integrator : public UVObject
{
public:
    Integrator() : TimeStep(1e-15) {}
    virtual ~Integrator() {}
    virtual Energy StepAllAtoms(Locator *, /*const*/ Potential *) = 0;
    virtual void Setup(DynAtom &) const = 0;
    virtual std::string getBaseName() const { return "Integrator"; }
    virtual void SetStep(SimulTime time) { TimeStep = time; }
    virtual SimulTime GetStep() const { return TimeStep; }
    virtual bool IsOK2Construct(const UVObject *pObj) const
    { return dynamic_cast<const Integrator *>(pObj) != 0; }
    virtual Integrator * Clone() const = 0;
protected:
    SimulTime TimeStep;

public:
    class IntegratorData
    {
    public:
        virtual ~IntegratorData() {}
    };
};
#endif

```

Su función más importante es `StepAllAtoms`, cuya misión consiste en calcular la nueva posición de los átomos para un tiempo que pasa de t a $t+\Delta t$. Esta función es una función virtual pura (abstracta), lo que significa que no define ninguna rutina en sí misma, sino sólo declara cómo deben comportarse las rutinas que deriven de ella. De esta forma, es en los integradores concretos derivados de éste, donde se codifica la forma particular en que cada integrador calcula las nuevas posiciones.

El integrador de Verlet

El integrador de Verlet se deriva de `Integrator`, y se limita a implementar el método de cálculo ya explicado anteriormente en la sección 2.3.1.

▷ 2.6.4

Potential

`Potential` es una clase genérica de UVAS que define las operaciones y requerimientos básicos que todo potencial debería cumplir. Posteriormente se deberán definir los potenciales concretos, que derivarán todos de esta clase base. El fichero de cabecera de `Potential` es:

```

// Potential.h: interface for the Potential class.
//
#if !defined(_POTENTIAL_H_)
#define _POTENTIAL_H_

#include "../Common/Locator.h"
#include "../Common/UVObject.h"
#include "../Common/CmdConstructor.h"

```

```

class DynAtom;

class Potential : public UVObject
{
public:
    virtual Energy GetPotentialEnergy(const Locator &, const Atom &) const = 0;
    virtual Energy GetPotentialEnergy(const Locator &) = 0; //Total energy.
    virtual Energy UpdateForces(Locator &) = 0;
    virtual ExtCoord GetCutOff() const = 0;
    virtual std::string getBaseName() const { return "Potential"; }
    void ComputeVirial(bool flag) { bVirial = flag; if(flag) Virial=0; }
    ExtCoord GetVirial() const { /*assert(bVirial);*/ return Virial; }
    void AtomsNotVisited(std::vector<DynAtom *> &) const;
    void EraseForcesPotsAndVisits(std::vector<DynAtom *> &) const;
    virtual bool IsOK2Construct(const UVObject *pObj) const
    { return dynamic_cast<const Potential *>(pObj) != 0; }
    virtual Potential* Clone() const = 0;

protected:
    ExtCoord Virial; //To contain the virial: Sum rF
    bool bVirial;
};

#endif // !defined(_POTENTIAL_H_)

```

Las funciones más importantes de `Potential` son `UpdateForces` y `GetPotentialEnergy`. `UpdateForces` calcula las fuerzas experimentadas por cada átomo y las almacena en un espacio reservado para tal fin en dicho átomo. Además devuelve el valor de la energía total del sistema, ya que este valor se obtiene de manera sencilla en el proceso de calcular las fuerzas, y resulta más efectivo que calcularlo por separado.

El potencial de Stillinger–Weber

El potencial de Stillinger–Weber es implementado en una clase que deriva de `Potential`, la cual define e implementa las rutinas necesarias para calcular este tipo de potencial empírico, como se comentó en la sección 2.4.1

▷ 2.6.5

Locator

`Locator` es una clase especial que ha sido creada para gestionar las posiciones de los átomos en el espacio y, más concretamente, para poder obtener de forma sencilla y rápida los vecinos de un átomo dado, aspecto que es vital para el cálculo de las fuerzas y energías por parte de los potenciales.

`Locator` necesita un radio de corte, (`CutOff`), que es la distancia máxima en la que se van a buscar vecinos (esta distancia generalmente coincide con el radio de corte del potencial). Una vez que el `Locator` dispone de ese dato, pasa a dividir el espacio en cajas o celdillas pequeñas, y genera una especie de plantilla o sello (ver figura 2.6.5) de las cajas vecinas que hay que tener en cuenta a la hora de buscar átomos vecinos. Gracias a esta plantilla `Locator` no se ve obligado a

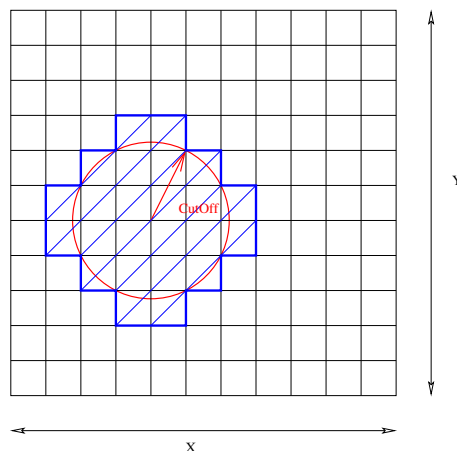


Figura 2.6: Forma en que se divide el espacio en el Locator para facilitar la búsqueda de vecinos.

buscar por todas las cajas de la simulación, lo que acelera bastante el proceso de obtención de vecinos.

▷ 2.7

Un ejemplo de aplicación: La migración del I_2 y su captura en superficie

El módulo de dinámica molecular explicado más arriba ha sido usado para determinar si el di-intersticial del silicio (I_2) es móvil y si, caso de serlo, es atrapado y recombinado o no en una superficie libre. Algunos estudios recientes (ver referencias (Gilmer et al., 1995; Marques et al., 2001; Hane et al., 2000; Estreicher et al., 2001; Chichkine et al., 2002; Chichkine and de Souza, 2002)) encuentran que el I_2 es móvil y con una energía de migración baja, por lo que toda la información que se pueda obtener sobre el I_2 y sus interacciones con la superficie y otras partículas es de gran interés para su inclusión en simuladores de procesos (Martin-Bragado et al., 2003). El estudio de la interacción con otras partículas (como boro), necesitaría del desarrollo de potenciales mixtos de silicio con dopantes, pero en cambio el estudio del intersticial I y el I_2 , su movilidad y su interacción con la superficie puede hacerse partiendo del típico potencial Stillinger-Weber a través de simulaciones de MD.

Una forma de comprobar si nuestro módulo de dinámica molecular funciona, antes de pasar a las consideraciones del I_2 , consiste en simular la difusión y posterior captura superficial de un intersticial sencillo, para lo cual se procede

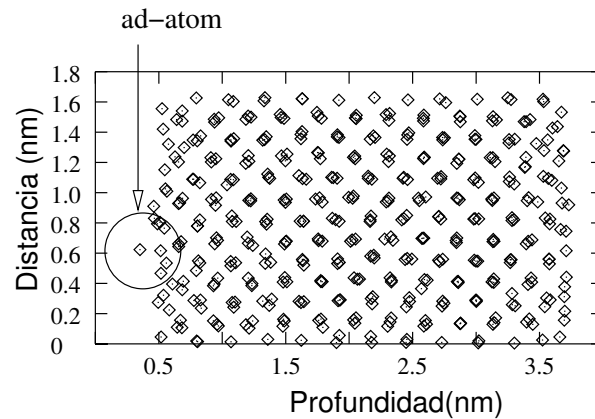


Figura 2.7: Un ad-atom, que fue un intersticial que se dirigió hacia la superficie y fue atrapado por ésta.

como se explicita a continuación:

1. Creamos un cristal con N átomos en posiciones del cristal perfecto, y dejamos una región en blanco (desde $x = 0$ hasta $x = 5 \text{ \AA}$) para generar la superficie libre.
2. Introducimos un átomo extra (el intersticial) en la mitad de la caja.
3. Relajamos la estructura completa con el nuevo átomo.
4. Recocemos el cristal a 1400°C (cerca de su punto de fusión) para asegurarnos de que la difusividad sea lo más grande posible.
5. Comprobamos si el intersticial (átomo o grupo de átomos con energías potenciales bastante mayores que el resto) se mueve y desaparece cuando llegue a la superficie.

Las simulaciones muestran, sin género de duda, que el intersticial sencillo se va moviendo al azar y, cuando se acerca a la superficie libre, es capturado por ésta y desaparece. La figura 2.7 muestra este intersticial, que al ser atrapado se convirtió en un ad-atom.

De esta manera, después de comprobar que el módulo de MD desarrollado funciona de manera adecuada con un intersticial, procedimos a crear una simulación con un di-intersticial:

1. Creamos un cristal con N átomos en posición de cristal perfecto, y dejamos una región (desde $x = 0$ hasta $x = 5 \text{ \AA}$) para generar una superficie libre.
2. Introducimos dos intersticiales extra en la mitad de la celda de simulación.

3. Relajamos el cristal con los nuevos átomos.
4. Recocemos el cristal a 1400 °C (cerca del punto de fusión) para intentar conseguir una difusividad lo más alta posible.
5. Comprobamos si los intersticiales (átomos o grupos de átomos con una energía potencial más grande que el resto) desaparecen o no en la superficie libre.
6. Comprobamos si el grupo de intersticiales se mueve o no, y si se mueven juntos, formando un I_2 , o separados como 2 I_s .

La figura 2.8 muestra una proyección 2D de una simulación en tres dimensiones. En ella se han destacado los átomos con una energía potencial más grande que el resto. Puede observarse que este grupo de átomos con más energía (el I_2 y algunos de sus vecinos que, al estar ligeramente comprimidos tienen una energía potencial algo más alta), se mueven juntos hacia la superficie la cual, finalmente, los captura.

La figura 2.9 muestra esta misma simulación representando la profundidad x frente a la energía potencial de cada átomo. Se puede observar como el grupo de átomos más energéticos (con una energía superior al resto) van marchando hacia la superficie (izquierda de la gráfica) y finalmente desaparecen en ella.

▷ 2.8

Conclusiones

Se ha desarrollado e implementado un algoritmo para realizar dinámica molecular en silicio, utilizando un potencial de Stillinger–Weber para las interacciones Si–Si y un integrador de Verlet para resolver numéricamente las ecuaciones de movimiento implicadas en el problema. Este algoritmo ha sido integrado en UVAS y programado en código C++. La rutina ha sido utilizada para mostrar que el di-intersticial propio del silicio es móvil, y que la superficie es capaz de atraparlo, confirmando otros estudios recientes al respecto.

Para más información sobre el método de la dinámica molecular remitimos al lector a las referencias (Heermann, 1990; Hockney and Eastwood, 1988).

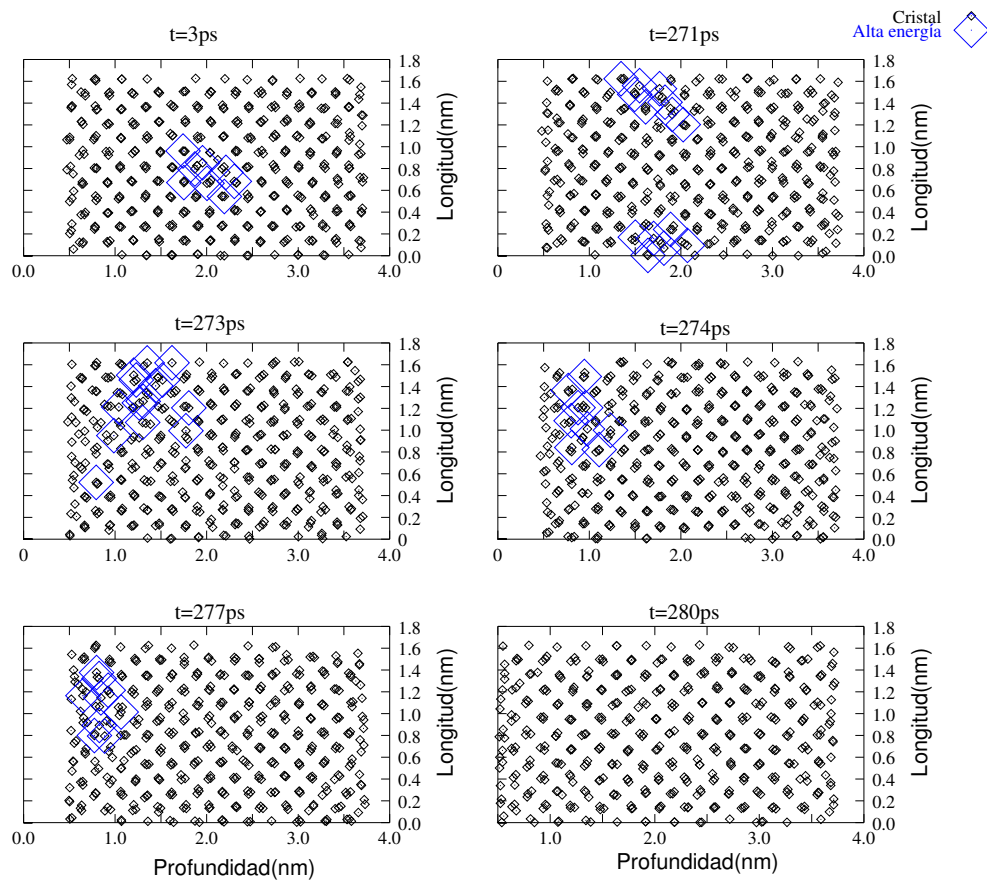


Figura 2.8: Proyección bidimensional de átomos (cuadrados pequeños), en la que se muestra el di-intersticial y su entorno (cuadrados grandes) marchando hacia la superficie y siendo recombinado por ésta.

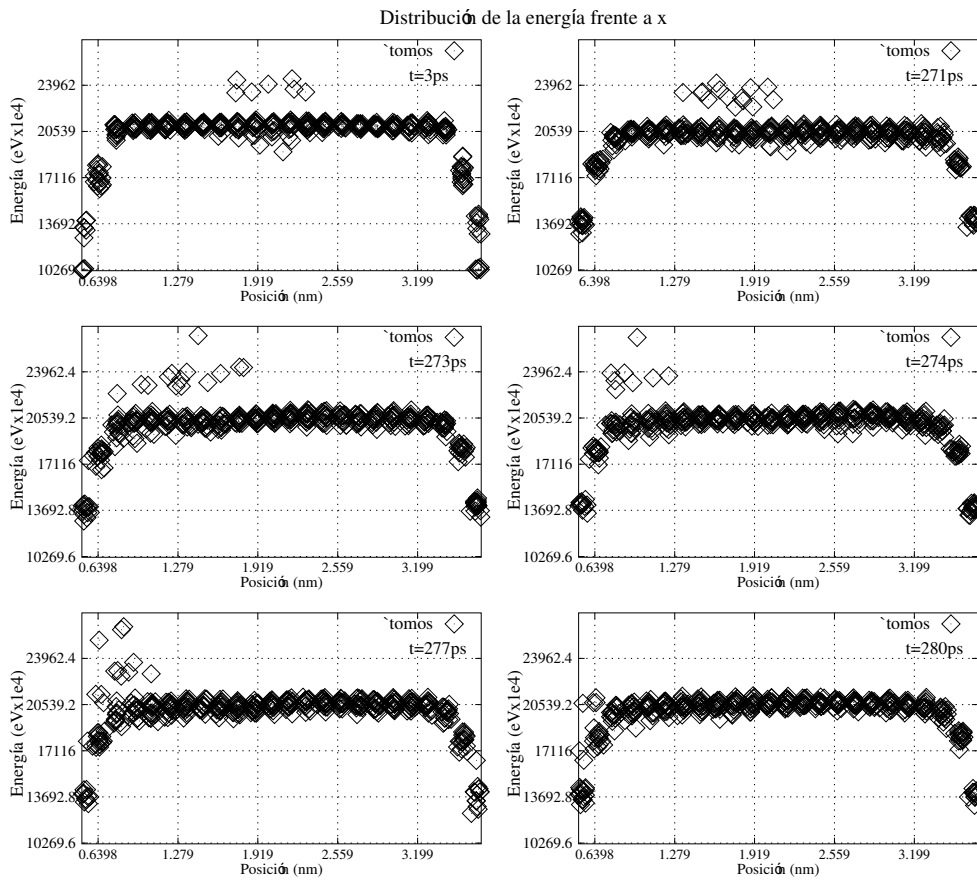
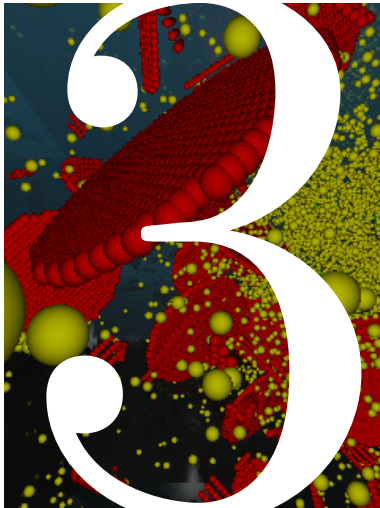


Figura 2.9: Evolución temporal de la energía de los átomos frente al eje x . El I_2 está rodeado por un conjunto de átomos bastante energéticos, que se van moviendo hacia la izquierda y terminan desapareciendo en la superficie.



Optimización de clusters usando algoritmos genéticos

| | |
|---|----|
| 3.1. Introducción . . . | 33 |
| 3.2. La técnica de los algoritmos genéticos | 33 |
| 3.3. Optimización de clusters aislados . | 37 |
| 3.4. Resultados | 42 |
| 3.5. Conclusiones . . . | 42 |

La búsqueda de configuraciones de mínima energía en sistemas físicos es una de las más habituales tareas de las simulaciones físicas. En este capítulo se presenta una técnica en la que, los algoritmos genéticos (GA) junto con dinámica molecular, han sido usados para buscar configuraciones de clusters de mínima energía en el vacío. Se muestran también los resultados para algunos de estos clusters, con tamaños $n = 4, 5, 6, 7, 8, 9, 10$ y 14 , que han sido encontrados utilizando potenciales de Stillinger–Weber y Lennard–Jones.

▷ 3.1

Introducción

La búsqueda de configuraciones de mínima energía en sistemas físicos es una de las más habituales tareas de la física computacional. La configuración de mínima energía tiene propiedades importantes, debido a que es la forma primordial en la que los sistemas se presentan en la naturaleza. No obstante, pese a su importancia y utilidad práctica, el proceso de minimizar la energía de los sistemas físicos es una tarea abrumadoramente compleja, debida a los muchos grados de libertad que suelen presentar estos sistemas, así como a la presencia de abundantes estados meta-estables (mínimos locales) que se presentan en el proceso.

El desarrollo de algoritmos de minimización eficientes es uno de los campos de investigación más relevantes de la física computacional, donde se están continuamente descubriendo nuevos métodos, muy en relación con los descubrimientos en el campo de la inteligencia artificial. Algunos de estos métodos recientes, pero no obstante de amplio impacto y aceptación, son las redes neuronales (Rao and Rao, 1993), Algoritmos Genéticos (Holland, 1975; Man et al., 1999), algoritmos “de hormiga” (Dorigo et al., 2000; Sützle and Hoos, 2000) y otros.

▷ 3.2

La técnica de los algoritmos genéticos

Los principios básicos en los que se sustenta la técnica de los algoritmos genéticos (GA) fueron propuestos por primera vez por (Holland, 1975). Esta técnica, como se intenta mostrar en la figura 3.1, está inspirada en los mecanismos de selección natural, donde los individuos más aptos resultan ser los “ganadores” en entornos de supervivencia competitivos. GA usa una analogía directa con dicha evolución natural: a través de un método de evolución genética se encuentra una solución óptima, que viene dada por el ganador de una selección genética.

La técnica GA supone que las posibles soluciones de cualquier problema son unos “individuos” que pueden representarse con un conjunto de parámetros, que a su vez están relacionados con los genes de un cromosoma. A través de una evolución de estos genes, el cromosoma que más adaptado esté al entorno tendrá una tendencia mayor de dejar una descendencia más abundante y cualificada, lo que supone una mejor solución al problema de la adaptación.

En las aplicaciones prácticas de GA se crea un conjunto inicial de cromosomas, generalmente al azar, que constituirán la población del problema. El tamaño de esta población dependerá del problema que se quiera resolver. Posteriormente, en cada iteración, considerada una evolución, se construye una nueva generación

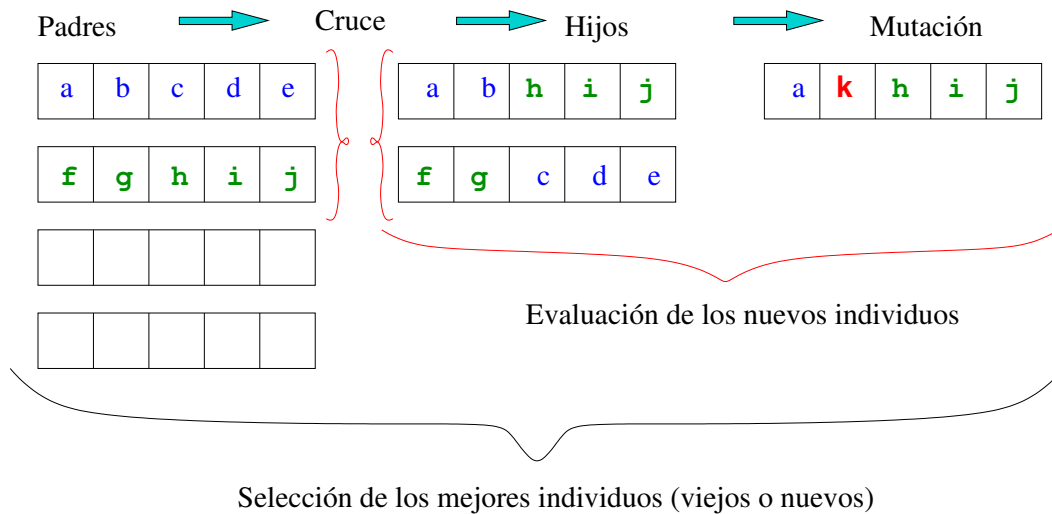


Figura 3.1: Los algoritmos genéticos seleccionan los mejores individuos de una población después de que se han permitido cruces y mutaciones entre sus miembros.

a partir de los cromosomas de la actual. Ello se hace seleccionando un grupo de cromosomas de la generación actual (los “padres”), y cruzándolos.

Como resultado de este proceso se espera que el “mejor” cromosoma genere una descendencia mayor y, por tanto, tenga una mayor probabilidad de sobrevivir en la siguiente generación, emulando de esta manera la supervivencia de los más aptos que puede observarse en la naturaleza.

Así pues, continuando la analogía con la naturaleza, cada individuo es un conjunto de parámetros, y el más apto será aquel que mejor se ajuste (dé un valor más adecuado) a una cierta función objetivo. Ésta será la función que queremos optimizar buscando su máximo o mínimo, y puede incluir cualquier tipo de parámetros continuos o discretos. Esta función produce como salida un número que evalúa el “cómo de buenos” son los parámetros de entrada, es decir, cada individuo concreto de la población. Por último habrá que construir algunos operadores genéticos: los operadores de cruce y mutación que, a partir de un conjunto inicial de individuos (los padres), crea otros nuevos y los modifica ligeramente, y el operador de reemplazo, que se encarga de seleccionar aquellos individuos (en principio poco adaptados), que van a desaparecer de la población.

Como resumen, los algoritmos genéticos convencionales operan de la siguiente manera:

1. Se marca la función objetivo, junto con los esquemas de reemplazo, cruce y mutación.
2. Se inicializa la población inicial (por ejemplo al azar).

3. Se evalúa cada individuo de la población inicial (utilizando para ello la función objetivo).
4. Para cada generación.
 - a) Se incrementa el número de la generación (como contador del proceso).
 - b) Se seleccionan los “padres”.
 - c) Se crean “hijos” a partir de esos padres (usando el operador de cruce).
 - d) Se mutan los hijos con una cierta probabilidad (usando el operador de mutación).
 - e) Se evalúan los nuevos individuos (función objetivo).
 - f) Se reemplazan algunos de los individuos viejos con los nuevos, dependiendo de lo bien adaptados que estén (usando el operador de reemplazo).
5. Se muestra el resultado de la población resultante en la última generación.

▷ 3.2.1

Un ejemplo como introducción

A modo de ejemplo simple para introducir el método GA, vamos a exponer un pequeño algoritmo cuyo fin sea minimizar la función de dos variables

$$f(x, y) = (x - 1)^2(y - 2)^2,$$

es decir, obtener $x = 1$ e $y = 2$.

La población de genes

Definimos los genes como las variables x e y y de paso, asociamos el valor de adaptación a cada gen, cosa que nos facilitará las posteriores manipulaciones. De esta forma vamos a construir la población como un array¹ de genes.

```
struct GENE
{
    float x, y, fitness;
    bool operator < (const GENE &g) const
    { return fitness < g.fitness; }
};
std::vector<GENE> popul;
```

¹Un vector de C++

La función objetivo

La función objetivo debe limitarse a evaluar los genes y asociarlos el valor de “cómo de bien se adaptan”. Este valor será la salida de la función, y cuanto más cercano esté a 0 mejor serían los genes.

```
void evaluation(GENE &g)
{
    g.fitness = (g.x-1)*(g.x-1)+(g.y-2)*(g.y-2);
}
```

Los operadores genéticos

El operador de mutación toma al azar el gen x o el y del cromosoma y lo cambia también de forma aleatoria. El operador de cruce creará los hijos a partir de los padres tomando los cromosomas x e y de éstos últimos e intercambiándolos.

```
const float mut_prob = 0.5;
void mutation(GENE &g)
{
    if (rand() > RAND_MAX/2)
        g.x = ((float) rand()) / RAND_MAX * 20. -10.;
    else
        g.y = ((float) rand()) / RAND_MAX * 20. -10.;
}
void crossover(const GENE &p1, const GENE &p2, GENE &c1, GENE &c2)
{
    c2.x = p1.x; c2.y = p2.y;
    c1.x = p2.x; c1.y = p1.y;
    if (rand() < RAND_MAX*mut_prob)
        mutation( (rand() < RAND_MAX/2)? c1:c2);
}
```

El bucle principal

El bucle principal de iteración comienza inicializando la población, y evaluando esta primera estimación.

```
int main()
{
    for(unsigned i=0; i<10; ++i)
    {
        popul.push_back(GENE());
        mutation(popul.back());
        evaluation(popul.back());
    }
    std::sort(popul.begin(), popul.end());
```

Posteriormente, mientras el valor de ajuste sea mayor que el requerido (o cualquier otro criterio que usemos para delimitar cuanto tiempo debe funcionar el proceso)

```
const float PREC = 1e-5;
while(popul.back().fitness > PREC)
{
```

tomamos dos individuos al azar, los “padres”, y los cruzamos.

```

unsigned p1 = unsigned((float)rand() / (RAND.MAX+1) * 10);
unsigned p2 = unsigned((float)rand() / (RAND.MAX+1) * 10);
GENE c1, c2;
crossover(popul[p1], popul[p2], c1, c2);

```

Posteriormente evaluamos los hijos generados, y los insertamos de nuevo en la población.

```

evaluation(c1);
popul.push_back(c1);
evaluation(c2);
popul.push_back(c2);

```

Finalmente, eliminamos los individuos que no ya no resultan aptos: para ello ordenamos toda la población conforme a su valor de adaptación, y borramos los peores.

```

std::sort(popul.begin(), popul.end());

while(popul.size() > 10)
    popul.pop_back();
}

```

Una vez que finaliza el bucle principal, nos limitamos a poner los valores de la población final, y acabamos.

```

while(popul.size())
{
    std::cout << "f(" << popul.back().x << ', ' << popul.back().y
                << ")_=_ " << popul.back().fitness << std::endl;
    popul.pop_back();
}

```

Con el ejemplo anterior, el programa tarda 0.028 segundos en mostrar la siguiente solución:

$f(0.998388, 1.99899) = 3.60831e-06$

Que está razonablemente cerca de la solución analítica del problema, que sería $x = 1$ e $y = 2$.

▷ 3.3

Optimización de clusters aislados

Hoy en día, el silicio constituye el material semiconductor más importante y utilizado para la industria microelectrónica. Si la tendencia a la miniaturización continúa, los dispositivos alcanzarán muy pronto un tamaño de cluster atómico. En dicho régimen, las propiedades y la estructura de los materiales son drásticamente diferentes de las que presenta su volumen y, por tanto, es más que interesante investigar estas minúsculas estructuras de silicio.

Pese a ello, la optimización de estructuras que contengan más de unos pocos átomos supone una tarea formidable, porque el número de posibles isómeros y estados meta-estables se incrementa de forma gigantesca con el número de átomos. Es por ello que en este campo tan especial, la técnica GA ha sido empleada de forma satisfactoria para encontrar las estructuras de más baja energía para sistemas con varios átomos y, especialmente, para clusters aislados de Si (Ho et al., 1998; Rata et al., 2000).

▷ 3.3.1

Introducción al modelo

Nuestro modelo utiliza los algoritmos de MD expuestos en el capítulo 2 junto con la técnica GA introducida en éste para optimizar clusters aislados en silicio. La dinámica molecular se encarga de calcular la energía total del sistema, utilizando el conocido potencial de Stillinger–Weber (Stillinger and Weber, 1985), y también es capaz de relajar las estructuras, encontrando configuraciones locales de mínima energía. La MD es excelente para encontrar estos mínimos locales cuando relaja estructuras, pero en absoluto es capaz de explorar configuraciones alternativas, quedando “enganchada” en dichos mínimos locales que, generalmente, están muy lejos de ser óptimos. Por otra parte, la técnica GA es particularmente buena explorando diferentes configuraciones y cambiando, a veces de forma drástica, la estructura de los clusters, pero muchos de estos intentos suelen estar muy lejos del equilibrio. Así pues, la combinación de ambas técnicas, permitiendo que la GA cambie las estructuras y la MD las relaje, forma una combinación muy adecuada para este problema.

Más detalles
en sec. 2.6.1,
pág. 20.

Consecuentemente con lo anteriormente dicho, la MD debe suministrar la función objetivo (que será la energía potencial total del sistema), y una manera de relajar las configuraciones. Por otra parte la GA tendrá que implementar los operadores genéticos (cruce y mutación) y el criterio para eliminar los individuos poco adecuados de la población. Para ello hemos utilizado las librerías GALib (Wall, 1996) e implementado dichos operadores con ellas.

▷ 3.3.2

El operador mutación

Hemos implementado tres tipos diferentes de mutación, de las cuales se selecciona una al azar. Estos tipos son:

Mutación aleatoria Toma un átomo cualquiera y le cambia su posición al azar.

Mutación de compactación Toma el cluster entero y mueve los átomos intentando que el cluster quede lo más compacto posible.

Mutación espejo Calcula el centro de masas del cluster, y seleccionando una

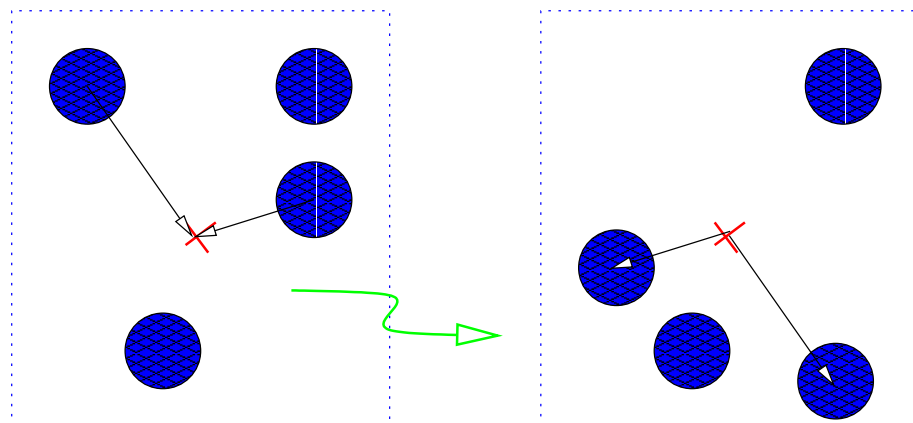


Figura 3.2: Mutación especular: Una vez que se ha calculado donde está el centro de masas del cluster, se borran la mitad de los átomos y se reconstruyen como su reflejo respecto al punto centro de masas.

mitad del cluster al azar, construye la otra mitad como un reflejo puntual de ésta. La figura 3.2 muestra esta técnica.

La idea de estos operadores es, por una parte, introducir cambios drásticos en los clusters, que es la misión de una mutación, pero por otra parte, hay que permitir que haya un cierto significado físico en dichos cambios. Así, por ejemplo, el operador de mutación genera simetrías en los clusters, que pueden facilitar considerablemente la búsqueda de las configuraciones de mínima energía.

▷ 3.3.3

El operador de cruce.

Es menester decir que la idea de tomar dos clusters, (los padres), y cruzarlos de forma que generen “hijos” suena bastante extraña, pero ésta es la misión del operador de cruce. Para implementarlo hemos intentado seguir las ideas del mismo expuestas en la referencia (Deaven and Ho, 1995), que permiten generar un operador de cruce con cierto sentido físico. La idea es escoger aleatoriamente un plano que divida los clusters iniciales en dos partes, y mover un poco este plano para asegurarse que el número de átomos a la izquierda de un cluster más el número de átomos a la derecha del plano del otro cluster, sigan sumando el número deseado de átomos del cluster total. Una vez que esto se consigue se generan dos “hijos” de la siguiente forma: la parte izquierda del primer cluster más la derecha del segundo conforman el primer hijo, y la derecha del primero más la izquierda del segundo el segundo hijo, como se muestra en la figura 3.3. Gracias a esta operación es posible mezclar simetrías diferentes, o combinar lo mejor de dos clusters en uno sólo.

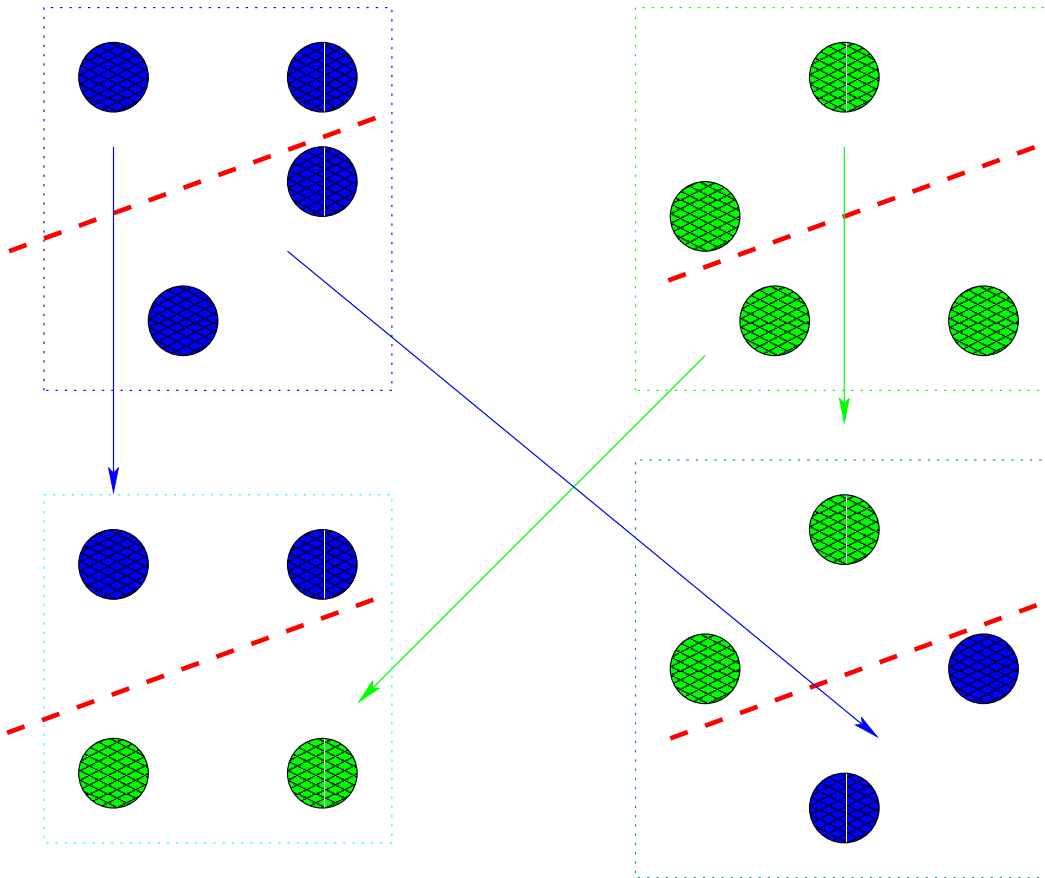


Figura 3.3: Para cruzar dos clusters se genera aleatoriamente un plano que divida los clusters padre en dos partes, y posteriormente se crea la descendencia como la mitad de cada padre. A veces, es necesario desplazar el plano para que las dos mitades sumen el número adecuado de átomos.

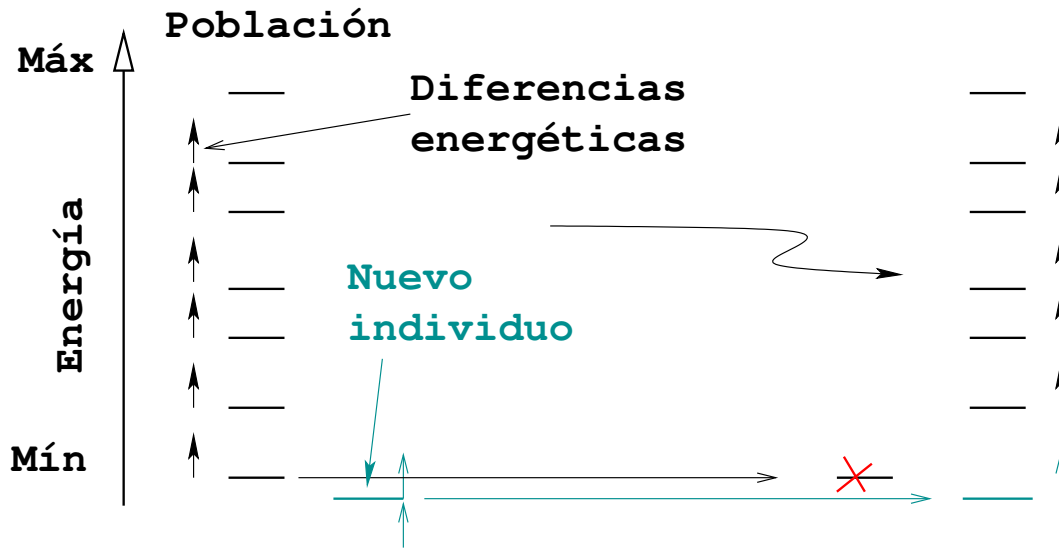


Figura 3.4: El operador de reemplazo intenta mantener una cierta variabilidad en la población, asegurándose una diferencia mínima de ΔE entre todos los individuos de ésta.

▷ 3.3.4

Reemplazando los individuos viejos o menos aptos

Después de que se haya construido un nuevo individuo, se llama al operador de reemplazo, pasándole este nuevo individuo. La labor de dicho operador es la de seleccionar qué elemento de la (vieja) población debe ser reemplazado por el nuevo, aunque el operador puede también decidir descartar el nuevo individuo y quedarse con toda la población anterior.

Nuestro modelo intentan descartar a los peores individuos de la población, definiendo peor como aquel individuo cuya energía potencial sea más grande que la de otros. A su vez el algoritmo procura mantener cierta diversidad de valores en la población, forzando a que existan unas diferencias energéticas mínimas, (codificadas por ΔE) entre unos individuos y otros. Cuando el nuevo individuo resulta ser mejor que todos los anteriores, la rutina descarta al peor, siempre que la diferencia energética entre el nuevo mejor y el viejo sea mayor que ΔE , porque si no el descartado sería el mejor de la vieja población. Cuando el nuevo individuo no resulta ser el mejor de toda la población, la rutina se preocupa de ver si puede insertar este nuevo individuo manteniendo la diversidad energética de la población. Cuando esto es posible, se descarta el peor individuo pero, si no fuera así, sería el nuevo individuo el borrado. Estas ideas están representadas gráficamente en la figura 3.4.

▷ 3.4

Resultados

Hemos minimizado clusters libres utilizando dos potenciales diferentes. Por una parte, la figura 3.5 muestra los resultados de las configuraciones con mínima energía para un potencial de Lennard–Jones. Este potencial está dado por la siguiente ecuación

$$V_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right],$$

donde σ y ϵ son parámetros específicos del potencial Lennard–Jones, diferentes según el tipo de átomos que interaccionen. En nuestro caso hemos tomado

$$\sigma = 0,3405 \text{ nm} \quad \epsilon = 1,65398 \times 10^{-21} J.$$

Lennard–Jones es un potencial central y por consiguiente los ángulos entre sus partículas no son tomados en consideración: sólo la distancia entre ellas. Usando este potencial el cluster de tamaño $n = 4$ resulta ser un tetraedro, la estructura más compacta posible cuando se tienen 4 átomos.

Por otra parte, la figura 3.6 muestra los resultados del algoritmo de optimización para clusters de silicio de tamaños $n = 4, 5, 6, 7, 8, 9, 10$ y 14 cuando se usa el potencial de Stillinger–Weber. Debido a que para este potencial los ángulos entre átomos de $\cos^{-1}(-1/3) = 109,47^\circ$ son los preferidos, las estructuras para tamaños 4 y 5 resultan ser planas, para así lograr ángulos más grandes que los que daría un tetraedro.

Por último, la figura 3.7 muestra un gráfico de la evolución de la energía potencial de un cluster de tamaño 4 usando un potencial de Stillinger–Weber. Las plataformas que aparecen son características en el método de los algoritmos genéticos, y se corresponden a situaciones de simulación en las que las nuevas generaciones no son capaces de mejorar la generación presente hasta que, por fin, una mutación o un cruce afortunado logra mejorar la población.

▷ 3.5

Conclusiones

En este capítulo se ha expuesto brevemente el método de los algoritmos genéticos (GA), y se ha ofrecido un ejemplo, sencillo pero completo, para ver cómo se pueden implementar los mismos. Más adelante se ha aplicado GA a la optimización de estructuras moleculares, junto con la técnica de MD para relajar las estructuras y calcular su energía. El modelo incorpora los necesarios operadores de cruce, mutación y reemplazo, que han sido adaptados para la optimización

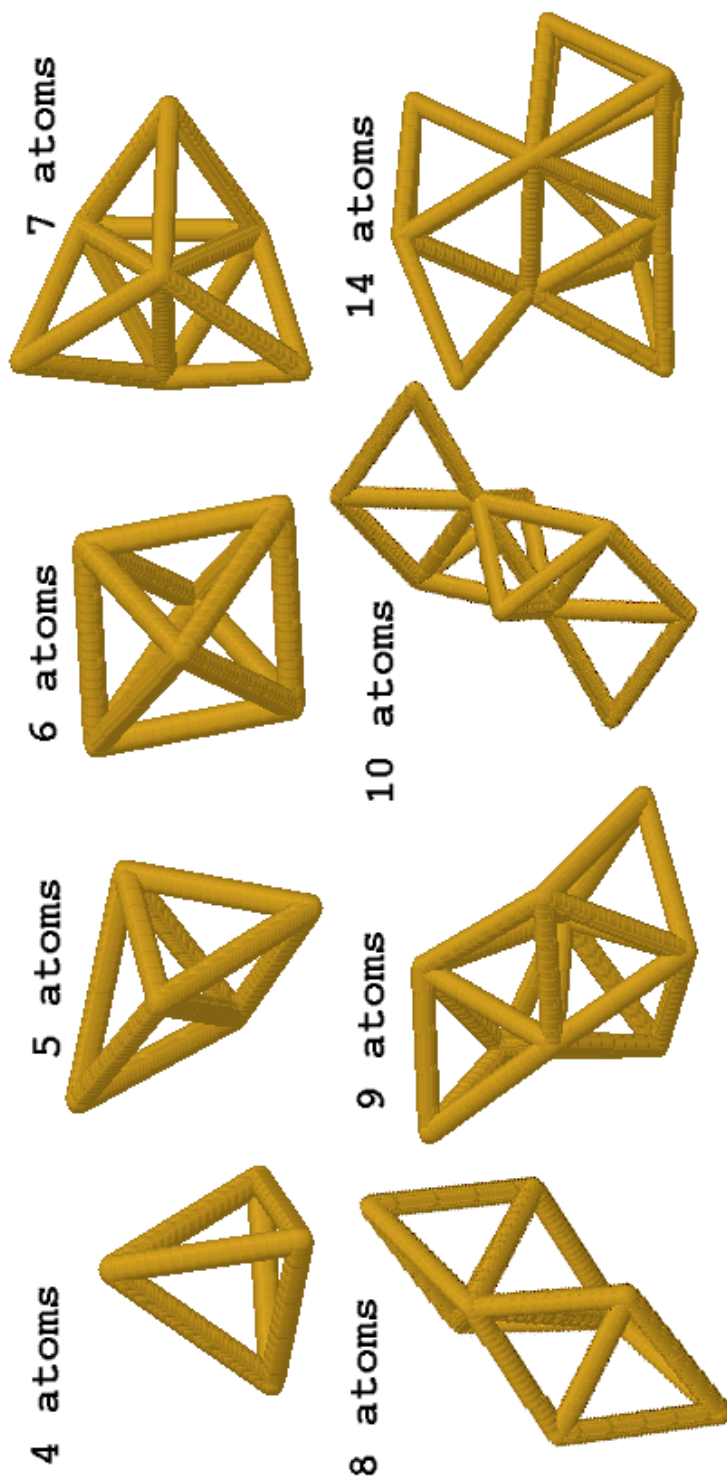


Figura 3.5: Clusters aislados de tamaño $n = 4, 5, 6, 7, 8, 9, 10$ y 14 usando un potencial Lennard–Jones.

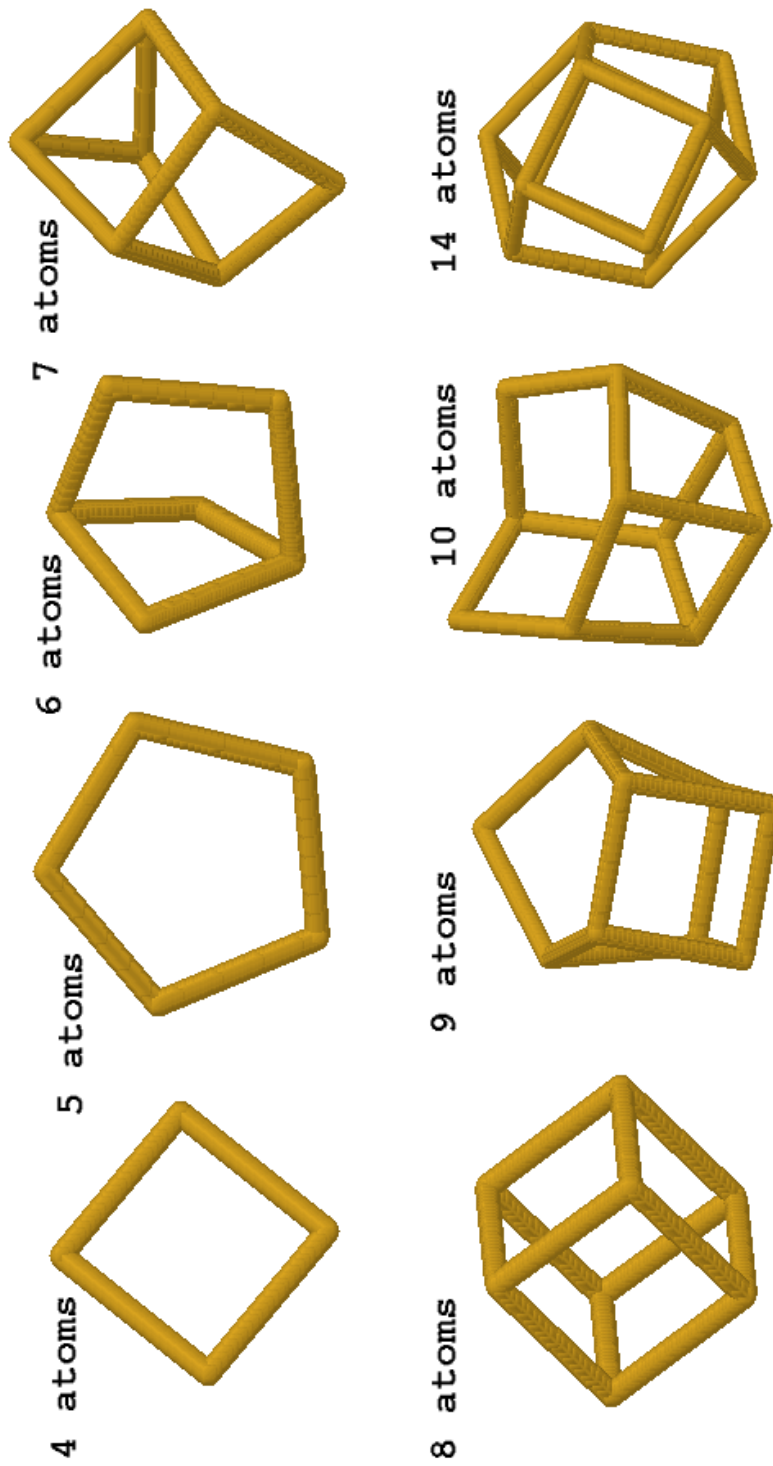


Figura 3.6: Clusters aislados de tamaños $n = 4, 5, 6, 7, 8, 9, 10$ y 14 usando un potencial de Stillinger–Weber.

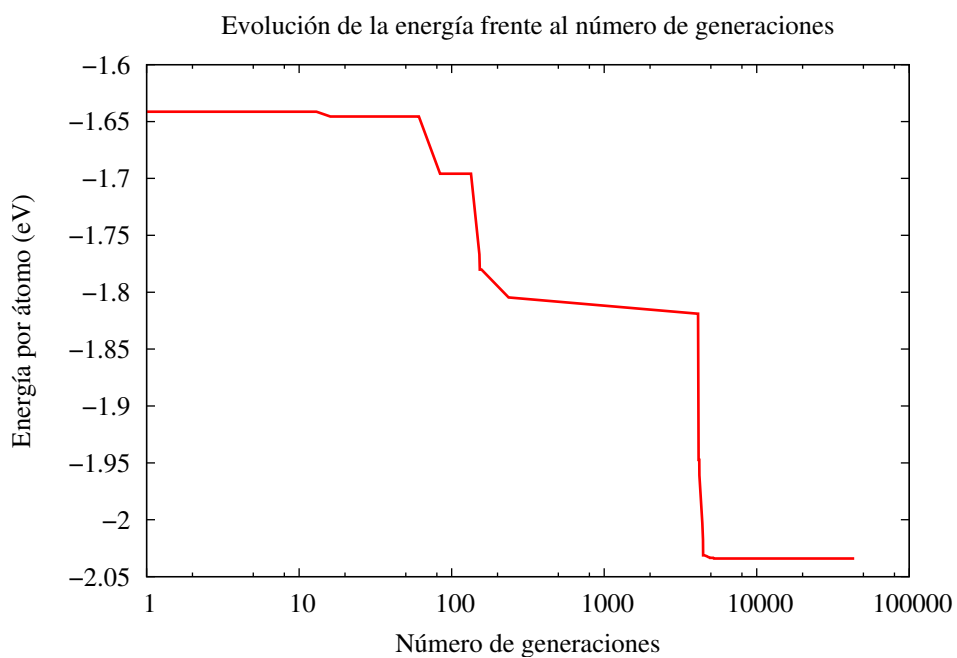
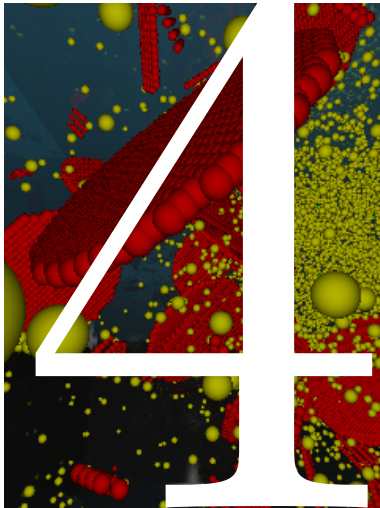


Figura 3.7: Variación de la energía durante las distintas generaciones del proceso de minimización por algoritmos genéticos para un cluster aislado de tamaño 4 usando el potencial Stillinger–Weber.

de clusters. Por último se han mostrado algunos de los clusters obtenidos por este método, tanto para potenciales de Lennard–Jones como Stillinger–Weber, viéndose que el potencial LJ procura crear los clusters de la forma más compacta posible, frente a SW, que intenta generar ángulos de enlace de 109° .

Parte II

El método de Monte Carlo



Los métodos de Monte Carlo

| | |
|---|----|
| 4.1. Introducción . . . | 51 |
| 4.2. Algunos ejemplos básicos | 51 |
| 4.3. Aplicaciones del método MC a la física estadística . | 54 |
| 4.4. Diferentes esquemas MC | 55 |
| 4.5. Conclusiones . . . | 57 |

En dinámica molecular se adopta el esquema de calcular las propiedades de los sistemas físicos a partir de las ecuaciones de movimiento fundamentales, para así crear las trayectorias en el espacio de fases. En cambio, los métodos de Monte Carlo (MC) adoptan una postura diferente: en ellos se comienza con una descripción del sistema, y se elige una configuración adecuada para el problema seleccionado, después se procede a calcular las magnitudes observables del sistema a partir de distribuciones asociadas y la función de partición. La idea es muestrear las contribuciones principales, para así conseguir una estimación de los observables.

▷ 4.1

Introducción

El método de Monte Carlo (MC) se caracteriza por la representación de la solución de un problema en función de una población hipotética, utilizando después una secuencia de números aleatorios para construir una muestra de dicha población, de la que se puede estimar una solución del problema inicial.

Usualmente, se siguen los siguientes tres pasos en los modelos de Monte Carlo:

- En primer lugar, el problema físico bajo consideración se traduce en un modelo análogo probabilístico o estadístico.
- En segundo lugar, este modelo probabilístico es resuelto a partir de un muestreo para el que se toman números aleatorios.
- En un tercer paso, se obtienen los datos a partir de estimaciones estadísticas.

El carácter estocástico de esta forma de operar requiere grandes cantidades de números aleatorios, que son generados por ordenador utilizando generadores aleatorios¹. Finalmente, la validez de este método está garantizada por el teorema del límite central de la teoría estadística.

▷ 4.2

Algunos ejemplos básicos

Una introducción breve al método MC puede darse intentando resolver algunos problemas muy sencillos en una y dos dimensiones, después de lo cual se explicarán algunos aspectos matemáticos respecto a la convergencia del método MC.

▷ 4.2.1

Una dimensión

Como el caso de simulación de Monte Carlo más sencilla, se puede considerar una integración numérica. Por ejemplo, para el caso de una dimensión consideremos la integral

$$\int_0^1 f(x)dx.$$

¹Como, por ejemplo, el método congruente, que calcula un número “aleatorio” r_{n+1} como $(a \times r_n + c) \text{ mód } n$, donde se comienza con un número inicial r_0 que actúa a modo de semilla.

Correspondiendo al eje x generaremos una secuencia de números aleatorios en el intervalo $r_i \in [0, 1]$, y sobre éstos calcularemos la operación

$$\frac{\sum_{i=1}^{i=n} f(r_i)}{n}.$$

Cuando $n \rightarrow \infty$, esta operación debería aproximarse al valor deseado aunque, en la práctica, puesto que utilizaremos sólo un número finito de números aleatorios, el resultado deberá estar cercano al verdadero, si bien no tiene por qué resultar exacto.

El código, (unas pocas líneas), que realiza la anterior operación, codificado en C++ sería el siguiente:

```
#include <iostream>
#include "RandomNumber.h"
using namespace std;
int main()
{
    RandomNumber alea;
    unsigned N;
    cout << "Number of iterations ";
    cin >> N;
    double sum[3] = {0, 0, 0};
    for (unsigned i = 0; i < N; ++i)
    {
        double x = alea.rand();
        sum[0] += x;
        sum[1] += x*x;
        sum[2] += x*x*x;
    }
    cout << "Integrals for f1(x) is " << sum[0]/N << " f2(x) is "
        << sum[1]/N << " and f2(x) is " << sum[2]/N << endl;
    return 0;
}
```

Donde los resultados dados son:

```
Number of iterations    1000000
Integrals for f1(x) is 0.499759 f2(x) is 0.333045 and f2(x) is
0.249691
```

Estos números están muy cercanos al resultado real de $1/2$, $1/3$ y $1/4$.

▷El método de MC es bastante sensible a la “calidad” de los números aleatorios generados y, debido a que dichos números no son realmente aleatorios, sino que son generados a partir de una rutina, es de extrema relevancia la calidad de dicha rutina. En nuestro caso, el algoritmo que usaremos para las aplicaciones del método MC es el de Marsaglia expuesto en la referencia (Marsaglia and Zaman, 1987).

▷ 4.2.2

Dos dimensiones

Como un ejemplo para dos dimensiones vamos a calcular el área de un círculo con radio unidad. Para ello se generarán puntos para $x \in [-1, 1]$ e $y \in [-1, 1]$.

Posteriormente se calcula $r^2 = x^2 + y^2$. Si $r^2 \leq 1$ incrementamos un contador y, por último, estimamos el área unidad dividiendo el valor final del contador por el número total de pares de números aleatorios (x, y) generados. El resultado final estará constituido por el resultado de esta división, multiplicado por el área del cuadrado considerado, en nuestro caso 4.

Un algoritmo que realice estas operaciones sería:

```
#include "RandomNumber.h"
#include <iostream>
using namespace std;
int main()
{
    RandomNumber alea;
    unsigned N, count = 0;
    cout << "Iterations = "; cin >> N;
    for (unsigned i=0; i<N; ++i)
    {
        double x=alea.rand();
        double y=alea.rand();
        if (x*x + y*y <=1) ++count;
    }
    cout << "The computed area (pi) is "
    << 4.0*double(count)/double(N) << endl;
    return 0;
}
```

y el resultado del área calculada:

Iterations = 1000000

The computed area (pi) is 3.14368

Que está bastante cerca del valor real de π .

▷ 4.2.3

Convergencia

La convergencia del método MC explicado más arriba está garantizada por la ley de los grandes números: sea x_1, \dots, x_n una serie de variables aleatorias seleccionadas de acuerdo a una función de densidad de probabilidad $\mu(x)$ con

$$\int_{-\infty}^{\infty} \mu(x) dx = 1.$$

Supongamos que $I = \int_{-\infty}^{\infty} f(x)\mu(x) dx$ existe, entonces $\forall \epsilon > 0$

$$\lim_{n \rightarrow \infty} P \left(I - \epsilon \leq \frac{1}{n} \sum_i^n f(x_i) \leq I + \epsilon \right) = 1.$$

No obstante, no se desea generar una gran número de muestras de longitud finita, sino una única muestra larga. La ley fuerte de los grandes números demuestra que:

$$P \left(\lim_{n \rightarrow \infty} \frac{1}{n} \sum_i^n f(x_i) = I \right) = 1.$$

El teorema, por tanto, garantiza que para una muestra de números lo suficientemente grande, se puede lograr un valor de la integral arbitrariamente cercano al resultado final.

▷ 4.3

Aplicaciones del método MC a la física estadística

El método MC puede ser usado para medir observables en física estadística. Consideremos la función de partición canónica

$$Z = \int_C \exp\left(-\frac{U(x)}{k_B T}\right) dx$$

en donde el sistema obedece al hamiltoniano

$$H = \sum_i \frac{p_i^2}{2m_i} + U(x).$$

Reescribiendo la función Z con la medida de Gibbs:

$$Z = \int_C d\mu(x),$$

siendo

$$d\mu(x) = \exp\left(-\frac{U(x)}{k_B T}\right).$$

Los valores medios para un observable O serán:

$$\langle O \rangle = \frac{1}{Z} \int_C O(x) d\mu(x). \quad (4.1)$$

Todos los estados X correspondientes a una energía muy grande darán contribuciones pequeñas a la integral, y únicamente algunos estados van a aportar grandes contribuciones. Se esperará, por tanto, que la distribución tenga un pico muy agudo cerca del valor medio de $H(x)$. De esta manera, si calculamos el observable de la ecuación 4.1 seleccionando al azar los estados, como el espacio de fases tiene muchas dimensiones, necesitaríamos un número tremendo de estados, de los que lamentablemente, muchos apenas aportarán nada a la contribución final. Es por ello que no se toman los puntos del espacio de fases completamente al azar y, al contrario, se les selecciona con una probabilidad $P(x)$. Esta probabilidad se toma como

$$P(x) = \frac{1}{Z} \mu(x)$$

para reducir la varianza del observable. Con esta elección $\langle O \rangle$ puede ser calculado como

$$\langle O \rangle \approx \frac{1}{n} \sum_i^n A(x_i).$$

Siguiendo con esta idea se necesita un algoritmo que genere estados con una distribución acorde a la probabilidad $P(x) = Z^{-1}\mu(x)$, y es aquí cuando puede haber variaciones. Si la distribución se conoce a priori, se puede realizar lo que ha sido llamado Monte Carlo cinético, en caso contrario se puede utilizar Metropolis Monte Carlo.

▷ 4.4

Diferentes esquemas MC

En los siguientes apartados vamos a exponer muy brevemente los métodos de Metropolis Monte Carlo (MC), Monte Carlo cinético (kMC) y la Aproximación de colisiones binarias (BCA).

▷ 4.4.1

Metropolis MC

Cuando no se conoce la distribución de estados a priori, el método de Metropolis Monte Carlo rodea el problema generando una cadena de Markov que, comenzando desde un estado inicial x_i , vaya creando más estados que, finalmente, estarán distribuidos de acuerdo con una probabilidad $P(x)$. Para hacer esto, hay que especificar las probabilidades de transición $w(x_n, x_{n+1})$ de un estado x_n al siguiente estado x_{n+1} . Esta elección no es única, aunque las más comunes son:

Metropolis rate $w(x_{n+1}, x_n) = w_0(x_{n+1}, x_n) \min \left[1, \exp \left(-\frac{H(x_{n+1}) - H(x_n)}{k_B T} \right) \right]$.

Gaubler rate $w(x_{n+1}, x_n) = w_0(x_{n+1}, x_n) \left\{ 1 - \tanh \left[\exp \left(-\frac{H(x_{n+1}) - H(x_n)}{k_B T} \right) \right] \right\}$.

De manera general, el método de Metropolis MC podría ser como a continuación se detalla:

1. Se especifica un punto inicial x_n en el espacio de fases.
2. Se genera un nuevo estado x_{n+1} .
3. Se calcula la probabilidad entre transiciones $P(x_n, x_{n+1})$.
4. Se genera un número aleatorio de manera uniforme en el intervalo $r \in [0, 1]$.

5. Si la probabilidad de transición entre estados P es menor que el número aleatorio generado r , entonces el viejo estado se considera como uno nuevo, es decir $x_{n+2} = x_n$. Si no, se acepta el nuevo estado $x_{n+2} = x_{n+1}$.

El problema que presenta este método consiste en que, cuando las probabilidades de transición son pequeñas, cuesta mucho encontrar un nuevo estado que sea aceptado por la probabilidad de transición, lo que puede suponer un gasto de recursos importante.

▷ 4.4.2

Monte Carlo cinético (kMC)

El método Monte Carlo cinético (kMC) se utiliza para simular procesos de activación, es decir, aquellos que ocurren con una probabilidad exponencial de:

$$P(x_i, x_f) = \exp\left(-\frac{E_a}{k_B T}\right). \quad (4.2)$$

En estos procesos, los estados son conocidos, y por tanto en las transiciones entre estados uno conoce las energías inicial y final necesarias para aplicar la ecuación 4.2. Ésta es una diferencia importante con el Metropolis MC: en kMC no se rechaza ningún estado, lo que acelera bastante el rendimiento del algoritmo en ciertas aplicaciones.

Un ejemplo típico y característico de aplicación de kMC es el proceso de migración de defectos en sólidos. Este proceso puede simularse tanto en sistemas no direccionales (non lattice kMC, sistemas sin red) como en sistemas con una red cristalina embebida (Lattice kMC, LMC). El algoritmo LMC se explica en el capítulo 5, y se da un ejemplo de la formación de caras durante el recocido de un cristal de aluminio. Por último, el método de MC cinético es estudiado con gran profundidad en la parte III de esta disertación, constituyendo de hecho, el tema principal de esta tesis. Esos capítulos incluyen una descripción completa de los mecanismos que se producen durante el procesado del silicio, y cómo éstos han sido modelizados e implementados en un potente simulador atómico kMC de procesos.

▷ 4.4.3

La aproximación de colisiones binarias (BCA)

Como comenta (Robinson, 1992):

La aproximación de colisiones binarias (BCA) ha sido usada desde hace tiempo en la simulación por ordenador de las interacciones entre átomos muy energéticos con blancos sólidos. Aunque se trata de una aproximación realizada para altas energías, el método BCA

retiene también mucho significado cualitativo para bajas energías y, con la adecuada formulación, da también información útil. Además, las simulaciones por ordenador basadas en el formulismo BCA pueden lograr una estadística buena en muchas situaciones en las que aquellas basadas en modelos de pura dinámica clásica (MD) requieren el más avanzado hardware de ordenador y, aún así, resultan impracticables.

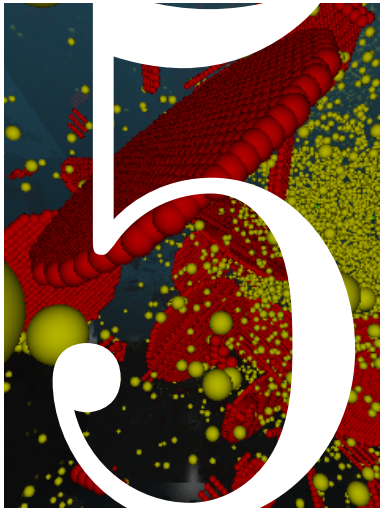
Hay que distinguir dos clases de BCA. En una de éstas, tipificadas por MARLOWE (Robinson and Torrens, 1974), la localización de los potenciales átomos de blanco están determinados por una estructura de cristal bien definida. Los métodos aleatorios juegan un papel, como mucho, auxiliar en este tipo de códigos, dando, por ejemplo, posiciones y direcciones de partículas iniciales, desorden local y cosas similares. Para la otra clase de modelos BCA, los blancos no tienen estructura (son amorfos). En estos códigos de Monte Carlo (MC), se utilizan métodos aleatorios para situar los potenciales átomos de blanco, o para determinar los parámetros de impacto, distancias de vuelo y demás. De estos últimos, el modelo más conocido es la familia de códigos TRIM (Biersack and Haggmark, 1980).

El método de colisiones binarias está someramente presentado en el capítulo 6.

▷ 4.5

Conclusiones

En este capítulo se ha hecho una introducción al método de Monte Carlo. Esta técnica está especialmente indicada para procesos físicos que involucren la migración de partículas en sólidos. Algunas variantes del método MC son el Metropolis MC, el Lattice MC (tratado en el capítulo 5), el MC cinético (parte 4) y la aproximación de colisiones binarias (capítulo 6). Remitimos al lector interesado a las referencias (Raabe, 1998; Ohno et al., 1999) si desea más información general sobre este método.



El método de Monte Carlo con red

| | |
|---|----|
| 5.1. Introducción . . . | 61 |
| 5.2. Difusión atómica para el modelo LMC | 62 |
| 5.3. Detalles de la implementación . . . | 64 |
| 5.4. Un ejemplo: Recocido de cristales con bordes afilados. | 67 |
| 5.5. Conclusiones . . . | 68 |

La técnica de Monte Carlo con red (Lattice MC) resulta ser excelente para estudiar los procesos de migración de átomos en capas policristalinas delgadas. Su punto clave consiste en permitir el salto de un átomo de una posición de la red a una posición vecina, con una determinada energía de activación E_a . Este mecanismo es suficiente para tener en cuenta diferentes efectos presentes en las simulaciones de migración en capas delgadas, como el crecimiento de los granos, la difusión a lo largo de las fronteras de grano, la formación de caras durante el recocido, etc. Estos efectos aparecen de manera natural cuando se seleccionan los valores correctos de E_a .

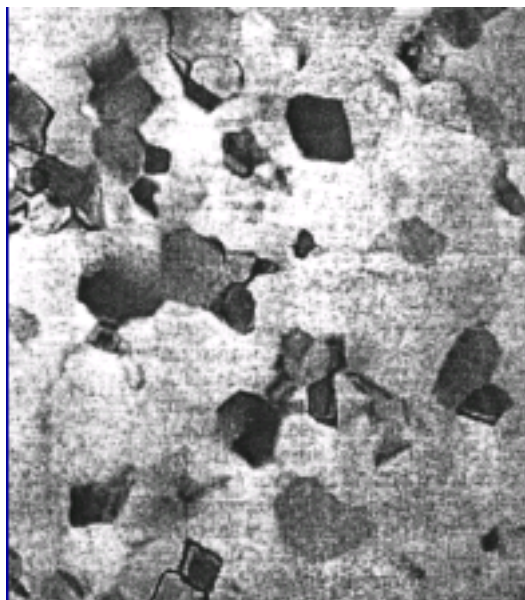


Figura 5.1: Microfotografía de una capa de aluminio de 120 nm tomada usando la técnica "Bright field TEM". Referencia (Kang and Ho, 1997).

▷ 5.1

Introducción

El depósito de capas delgadas policristalinas resulta esencial para la fabricación de dispositivos microelectrónicos. Aproximadamente la mitad de los pasos de procesamiento están dedicados al depósito de conductores metálicos, barreras de difusión y capas de aislante, durante la fase de metalización. El rendimiento y velocidad de los dispositivos de muy alta escala de integración (ULSI) está marcado por sus tamaños decrecientes, que han ido disminuyendo exponencialmente de acuerdo con la ley de Moore. A modo de ejemplo, la figura 5.1 muestra un policristal de aluminio. De esta manera, la micro-estructura y las propiedades de conducción de las capas delgadas son un tema crucial para la fiabilidad de los circuitos de interconexión.

En los últimos años se ha hecho una cantidad considerable de trabajo, tanto experimental como teórico (Lita and Jr., 1998; Lita and Jr., 2000; Barna and Adamik, 1998; Thompson, 2000; Gilmer et al., 2000; Huang et al., 1998; Rubio et al., 2003), para caracterizar y predecir las propiedades de las capas delgadas depositadas bajo diferentes condiciones experimentales, temperatura, velocidad de depósito, tipo de sustrato y distribuciones angulares y energías cinéticas de los átomos incidentes. También han sido propuestas varias soluciones para superar la dificultad de manejar sitios pertenecientes a diferentes orientaciones cristalinarias

dentro de la misma celda de simulación.

▷ 5.2

Difusión atomística para el modelo LMC

Una simulación de LMC puede comenzar con una distribución inicial de átomos en la red, o quizás con una celda de simulación vacía que se va llenando de átomos que se van depositando a una velocidad fija. En cualquier caso, existe también un evento de difusión de átomos que sucede con una frecuencia adecuadamente calculada. Este evento mueve un átomo desde una posición de red a otra adyacente, con una frecuencia que se calcula de acuerdo a la siguiente ecuación

$$\nu_i = \nu_{0i} \exp(-(E_{\text{mig}} + \Delta E)/k_B T), \quad (5.1)$$

donde E_{mig} es la energía de migración, que depende del número de coordinación y $\Delta E = E_{\text{final}} - E_{\text{init}}$ es la diferencia de energía del sistema antes y después del salto, cuando dicha diferencia es positiva, o $\Delta E = 0$ en caso contrario. En nuestro caso, una forma simple y rápida de calcular dicha diferencia es suponer que E_{final} y E_{init} dependen tan sólo del índice de coordinación de los sitios involucrados en el salto del átomo, definiéndose este índice como el número de primeros vecinos existentes en una determinada posición. La figura 5.2 muestra la energía potencial de un átomo de aluminio en función de su número de coordinación. Se considera que las distancias y ángulos entre los diferentes sitios atómicos dentro de un grano con la misma orientación cristalina van a ser los de un cristal FCC perfecto (la estructura característica del aluminio), pero esto hace plantearse qué va a suceder cerca de las fronteras de grano. En las simulaciones que vamos a presentar aquí se ha escogido la siguiente aproximación: la energía está determinada únicamente por el número de coordinación incluso en las fronteras de grano. En el caso de átomos que no tengan posiciones vecinas vacías no se permite su movimiento, de manera que la lista de átomos que pueden saltar se limita únicamente a aquellos que se encuentren cerca de superficies, o en fronteras de grano u otro tipo de interfaces.

Es importante modelizar la dinámica de los granos y las fronteras de grano (ver figura 5.3) de una manera realista pero lo más simple posible. En nuestro caso se permite saltar a los átomos cerca de bordes de grano a los sitios vecinos, fenómeno que da lugar al movimiento de las fronteras de grano, pero no se tiene en cuenta ninguna relajación entre éstos.

Como se muestra en la figura 5.4 se considera que hay varios destinos posibles para los átomos que están saltando cerca de una frontera de grano. Estos sitios pueden estar en diferentes orientaciones de red, y serán permitidos siempre que estén suficientemente cerca (a una distancia de primeros vecinos o menor).

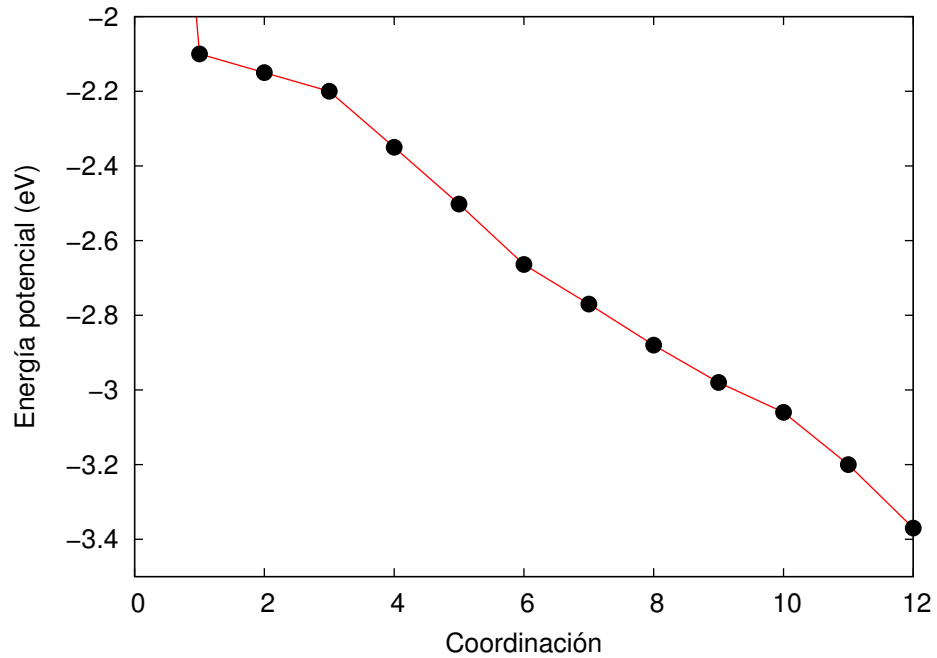


Figura 5.2: Energía de enlace para átomos de aluminio en una red FCC, dependiendo de su número de coordinación. Referencia (Huang et al., 1998).

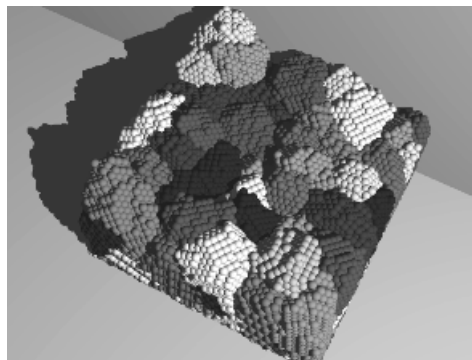


Figura 5.3: Diferentes granos, simulados con un código completamente tridimensional de tipo Lattice Monte Carlo atómico. (Rubio et al., 2003).

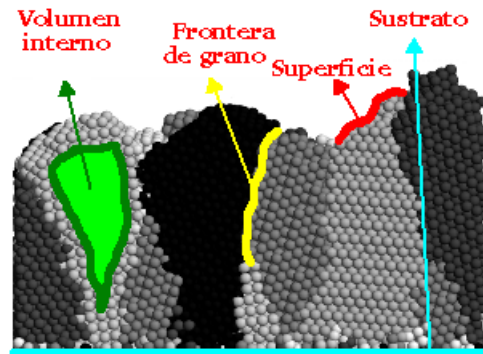


Figura 5.4: Cerca de las fronteras de grano aparecen varios destinos posibles para los átomos que están saltando, de los cuales alguno puede pertenecer a orientaciones de red distintas. Se permite el salto a estos sitios siempre que se encuentren suficientemente cerca, es decir, a una distancia de primeros vecinos o menor.

Con este mecanismo de intercambio de átomos que depende de las energías de activación, el movimiento de las fronteras de grano emerge como un resultado macroscópico de los mecanismos atomísticos.

El algoritmo LMC mueve un átomo cada vez, e incrementa el tiempo de la simulación en una cantidad proporcional al inverso de la suma de todas las frecuencias de migración. A su vez, la probabilidad de salir elegido es proporcional a dicha frecuencia de salto: cuanto más grande sea la frecuencia de salto, más grande será el número asociado con su evento particular, de tal forma que se usa esta frecuencia para elegir el siguiente átomo que saltará, haciendo la elección de manera aleatoria, pero teniendo en cuenta que la probabilidad de que un átomo salga elegido debe ser proporcional a su frecuencia de migración. Por último, tras mover el átomo seleccionado, toda la información que tenían los átomos de las inmediaciones debe ser actualizada, junto con sus frecuencias de salto.

▷ 5.3

Detalles de la implementación

La celdilla de simulación es manejada por un objeto, que hemos llamado `LMCSet`, y que es un conjunto de otros objetos más pequeños que a su vez se encargan de gestionar los átomos. Por último cada átomo pertenece a una red con una determinada orientación, y puede moverse de acuerdo a dicha orientación.

▷ 5.3.1

Diferentes conjuntos de átomos

Nuestra implementación de LMC clasifica los átomos en objetos diferentes dependiendo del papel que jueguen a lo largo de la simulación. Estos roles están representados en la figura 5.5. Cada átomo está así contenido en un objeto diferente, y la reunión de todos estos objetos conforma el macro-objeto `LMCSet`. Estos objetos son:

OrientationSet. Gestiona el conjunto de átomos en el cual todos presentan la misma orientación de red, y que a su vez están rodeados por todas partes de átomos, también con esa misma orientación. De esta forma, los átomos que pertenecen al objeto `OrientationSet` conforman el volumen interno de los granos.

SurfaceSet. Maneja los átomos que, teniendo todos la misma orientación de red, se encuentran en las proximidades de sitios huecos. Así pues, este objeto básicamente gestiona las superficies.

InterfaceSet. Gestiona los átomos que teniendo todos la misma orientación de red, estén en las cercanías de sitios con orientación de red diferente, es decir, los átomos pertenecientes a las fronteras de grano.

SubstrateSet. Contiene aquellos átomos que están cerca del sustrato, que es la capa de material sobre la que descansan éstos.

Así pues, se han organizado los átomos en los conjuntos anteriormente citados, donde cada conjunto es responsable únicamente de los átomos que a él pertenecen.

▷ 5.3.2

Gestionando el tiempo

El control del paso del tiempo en la simulación corre a cargo de un gestor de eventos, el cual contiene la lista de todos los eventos posibles y sus respectivos pesos para cada conjunto de átomos citados más arriba. De manera general, podemos decir que este gestor almacena las diferentes frecuencias de salto en una barra (similar a la representada en la figura 8.4, a la que añade nuevos valores cada vez que se inserta un nuevo evento de salto. A la hora de seleccionar uno de ellos genera un número aleatorio entre 0 y la longitud total de la barra, eligiendo el evento, (y por tanto el átomo que saltará), de acuerdo con el número que salga. El incremento de tiempo que duró durante ese evento (o hasta que finalizó) se calcula proporcionalmente al inverso de la longitud total de la barra, que a su vez es proporcional a la suma de todas las frecuencias de salto.

Para nuestras simulaciones de aluminio únicamente hemos implementado un tipo de evento: el salto de migración de un átomo. El esquema que sigue la simulación hasta la consecución total de este evento es:

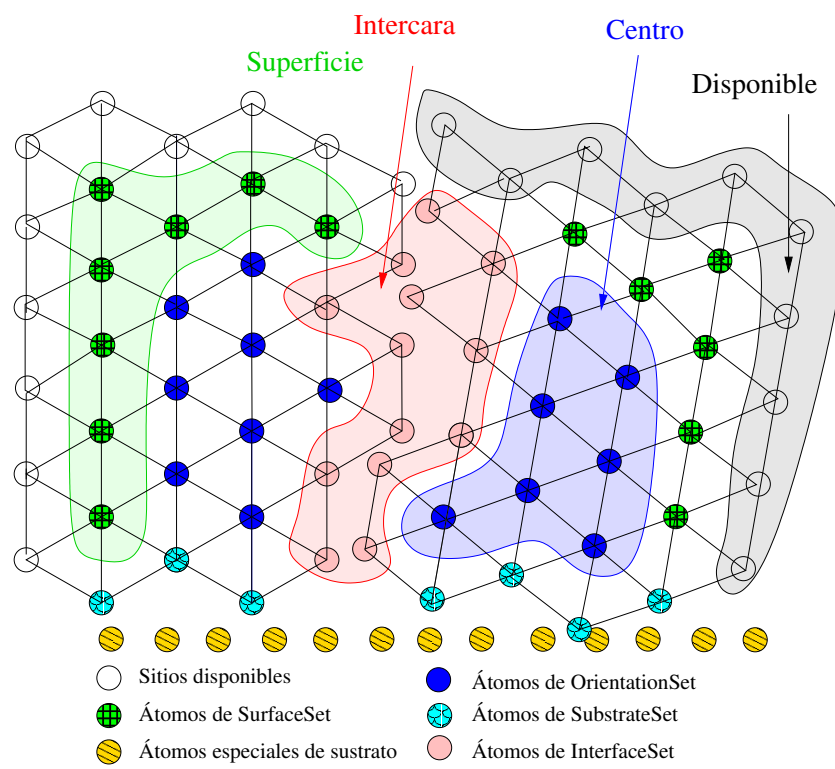


Figura 5.5: Formación de la red y clasificación de los átomos según sus posiciones.

- El objeto `LMCSet` elige, de manera aleatoria, pero proporcional a la diferente probabilidad de cada conjunto, uno de los conjuntos de átomos existentes en la simulación, y llama a su función “haz evento”.
- El objeto seleccionado elige a su vez un átomo (también teniendo en cuenta sus pesos) e invoca su función “haz evento”.
- Cada átomo de la simulación está rodeado de una serie de átomos de la misma o de diferente red, y de unos pseudo-átomos especiales situados en las posiciones vacías o libres de la red (huecos). Cuando se llama a la función “haz evento” del átomo, éste se mueve a una de estas posiciones adyacentes y actualiza las frecuencias de salto de todos sus vecinos viejos y nuevos, así como de él mismo. También se borra a sí mismo y a sus vecinos del grupito al que pertenecían.
- Una vez actualizadas las posiciones y frecuencias de salto, el átomo llama a una función de clasificación que pertenece al objeto global `LMCSet`. Esto es necesario porque al saltar un átomo puede cambiar de grupo de pertenencia. Por ejemplo, un átomo que estuviera en el interior del volumen podría quedar expuesto y pasar a la superficie al saltar su vecino.
- Una vez que el átomo ha sido clasificado, se inserta en el grupo al que pertenezca, y el código queda listo para un nuevo evento.

Esta forma de organizar las cosas puede parecer bastante compleja a primera vista, pero ha sido hecha así para otorgar la mayor flexibilidad y escalabilidad al código. Como ejemplo, los átomos pertenecientes al `OrientationSet` no se pueden mover, y por tanto podrían ser borrados de la simulación para ahorrar tiempo y espacio de memoria, o bien ser gestionados por un simulador diferente, etc. A su vez los átomos que estuvieran cerca del sustrato podrían tener frecuencias distintas dependiendo de las propiedades del sustrato, etc. No hemos implementado aún ninguna de estas propiedades adicionales, pero el camino está hecho y preparado para futuras investigaciones.

▷ 5.4

Un ejemplo: Recocido de cristales con bordes afilados.

La figura 5.6a muestra un pequeño bloque de aluminio que presenta algunos bordes afilados que, una vez recocidos, se suavizan bastante. Los átomos en los bordes presentan un número de coordinación bastante reducido y, consecuentemente, una energía de enlace también pequeña. Por ello es sencillo para este tipo

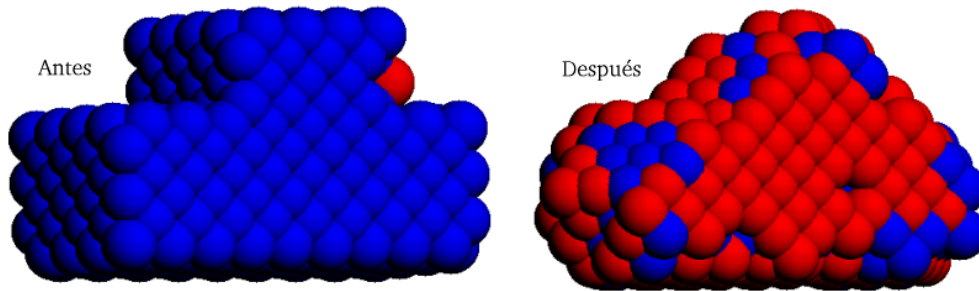


Figura 5.6: Proyección tridimensional de un cristal antes y después del recocido. Los átomos rojos han sido movidos en los pasos anteriores.

de átomos difundirse y moverse a otros sitios, hasta que encuentran algún sitio con un número de coordinación mayor y quedan “pegados”. Algunas caras (111) planas, junto con terminaciones (100) que se han ido formando para evitar los vértices agudos, se pueden ver en la figura 5.7. Este mismo sólido, después del recocido, está expuesto en la figura 5.6b.

Por último, las referencias (Huang et al., 1998) y (Gilmer et al., 2000) proponen un simulador LMC atómico similar al que hemos implementado y explicado en este capítulo. La figura 5.8 muestra sus simulaciones de aluminio depositando 1×10^3 , 2×10^3 y 5×10^4 átomos, y recociendo después a 200K. Hemos realizado una simulación similar con nuestro simulador, cuyos resultados pueden verse en la figura 5.9.

▷ 5.5

Conclusiones

Se ha dado una breve introducción al método de Lattice Monte Carlo aplicado a la migración de átomos en sólidos. El evento principal de este método es la migración o salto de un átomo. Se ha visto también un modelo sencillo para calcular la probabilidad de salto de uno de estos átomos, que depende, siempre en nuestro modelo, de su número de coordinación. Por último se ha comprobado cómo este modelo da lugar a la formación de caras cristalinas y al suavizado de aristas pronunciadas en bloques de aluminio de algunos nanómetros de tamaño.

Para más detalles se puede consultar (Rubio et al., 2003).

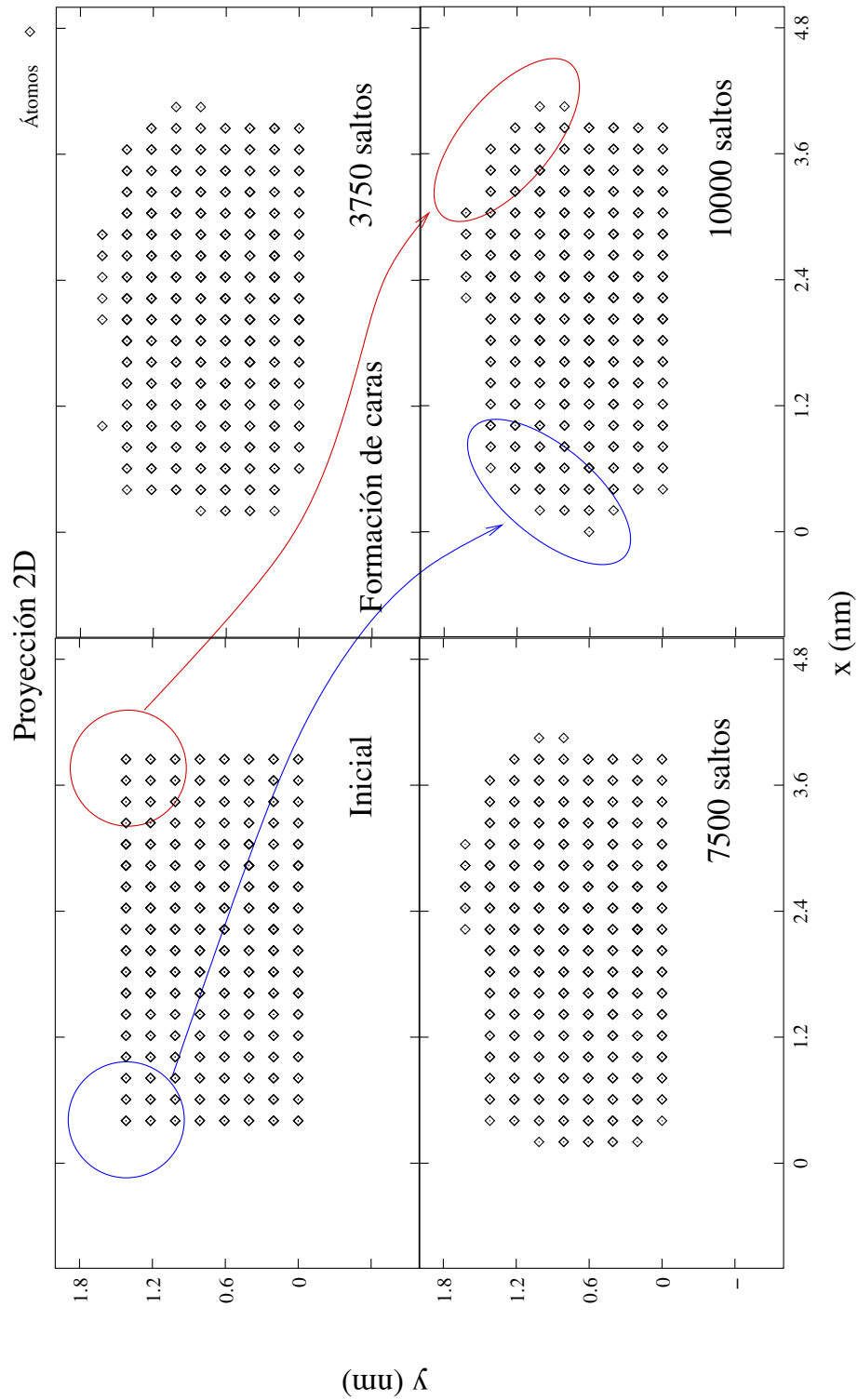


Figura 5.7: Los átomos que están situados en bordes agudos tienen una probabilidad más alta de moverse, de tal manera que estos fenómenos colectivos de migración van llevando a la formación de caras cristalográficas.

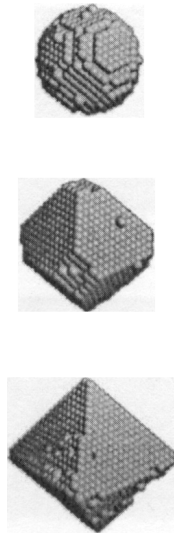


Figura 5.8: Simulación del depósito de 1×10^3 , 2×10^3 y 5×10^4 átomos de aluminio y posterior recocido a -73°C . Referencia (Huang et al., 1998).

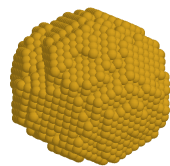
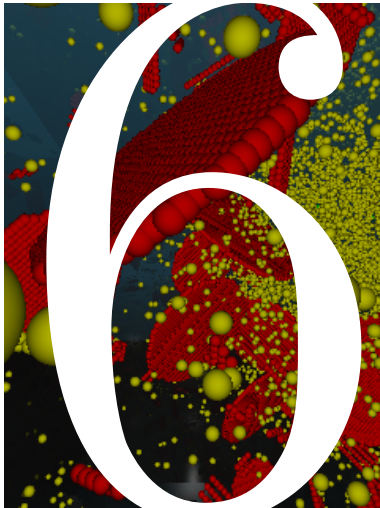


Figura 5.9: Cubo de aluminio recocido a 700°C y enfriado después hasta -73°C . Simulación realizada con el módulo LMC del simulador UVAS.



La aproximación de colisiones binarias

| | |
|---|----|
| 6.1. Introducción . . . | 73 |
| 6.2. La aproximación de colisiones binarias | 74 |
| 6.3. Potenciales interatómicos | 75 |
| 6.4. Pérdidas inelásticas de energía . . | 75 |
| 6.5. Conclusión | 76 |

Uno de los procesos tecnológicos empleados en microelectrónica para introducir perfiles dopantes en los semiconductores es la implantación iónica. El conocimiento de la profundidad del perfil de la especie implantada y del daño (desorden de la red) generado es crucial para poder determinar después las propiedades eléctricas de los transistores que compondrán el circuito integrado (IC) procesado, existiendo dos grandes metodologías para obtener este resultado: la dinámica molecular (MD) y el método de colisiones binarias (BCA). MD es extraordinariamente preciso, pero para ser llevado a cabo requiere de unos recursos computacionales realmente grandes. BCA, por otra parte, no es tan exacto como MD, pero sí es capaz de producir unos resultados suficientemente buenos con una cantidad limitada de recursos.

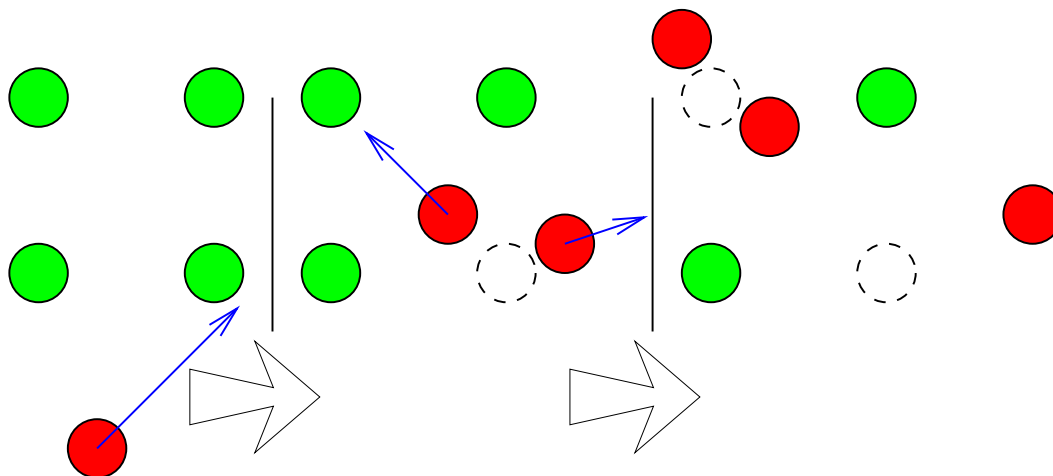


Figura 6.1: Seguimiento de los proyectiles tras las colisiones.

▷ 6.1

Introducción

La simulación del proceso de implantación iónica utilizando el método de la aproximación de colisiones binarias (BCA) se considera generalmente un método de Monte Carlo, en el cual se siguen las trayectorias individuales de iones que inicialmente han sido lanzados al azar para calcular los perfiles finales de impurezas. La suposición hecha por la aproximación de colisiones binarias consiste en suponer que, en cada colisión, únicamente chocan el átomo incidente con un átomo blanco, con lo cual sólo se consideran dos partículas cada vez. La partícula incidente perderá parte de su energía, que será cedida (elásticamente) al blanco, y a los electrones de las inmediaciones (inelásticamente). Por último, cuando la energía cinética del proyectil sea menor que un cierto valor umbral E_T , el código asume que el ion se ha parado.

A su vez, cuando la energía que se transfiere a los blancos sea mayor que la energía de enlace a la red, se crea un par de Frenkel (intersticial-vacante), y se añade este proyectil secundario, el intersticial, a la lista de proyectiles a seguir, dejando así un sitio vacío o vacante (ver figura 6.1).

Respecto a los blancos, éstos pueden ser un material cristalino, o bien uno sin estructura (amorfo). Para los materiales cristalinos se ha desarrollado el programa MARLOWE (Robinson and Torrens, 1974), frente al código TRIM (Biersack and Haggmark, 1980) para los amorfos. En nuestras simulaciones vamos a usar una versión del MARLOWE modificada en la Universidad de Valladolid (Arias et al., 1995).

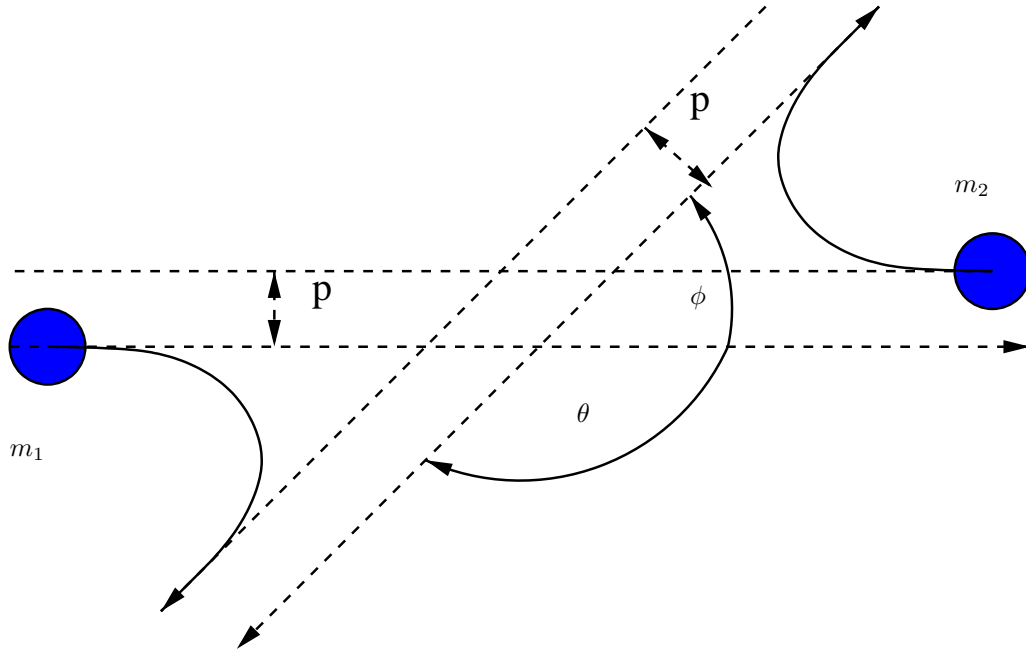


Figura 6.2: Colisión binaria (referencia desde el centro de masas).

▷ 6.2

La aproximación de colisiones binarias

La hipótesis principal realizada por la aproximación de colisiones binarias consiste en separar las interacciones de los proyectiles incidentes en una serie de colisiones a dos cuerpos. Esta separación supone que los cambios relativos en la energía de los átomos que interactúan pueda quedar confinada a las cercanías de los átomos blanco, de tal forma que la deflexión de los proyectiles se complete en una escala bastante menor que las distancias entre átomos del cristal. La figura 6.2 muestra dos átomos vistos desde su centro de masas: el proyectil incidente presenta una masa m_1 y una velocidad v_0 , y el blanco, inicialmente en reposo, tiene una masa m_2 . En la figura p representa el llamado parámetro de impacto.

Con este sistema de coordenadas, el ángulo θ se calcula como:

$$\theta = \pi - 2p \int_{r_{\min}}^{\infty} \frac{dr}{r^2 g(r)}, \quad (6.1)$$

donde

$$g(r) = \sqrt{1 - p^2/r^2 - (1 + A)V(r)/E(m_2/m_1)},$$

$V(r)$ es el potencial entre los átomos, y E la energía cinética del centro de masas.

La energía que se transfiere al blanco tras la colisión resulta ser

$$T = E_0 \frac{4m_2/m_1}{(1+A)^2} \sin^2(\theta/2).$$

La ecuación anterior 6.1 ha de ser resuelta numéricamente. Existen diversos métodos para ello, de los cuales el más popular es el esquema de cuadratura de Gauss-Mehler (Chebyshev) (Kopal, 1961).

▷ 6.3

Potenciales interatómicos

La ecuación 6.1 requiere que haya un potencial $V(r)$ definido entre las partículas que colisionan. En una primera aproximación se puede elegir el potencial de Coulomb

$$V(r) = \frac{Z_1 Z_2 e^2}{4\pi\epsilon_0 r},$$

pero posteriormente hay que tener en cuenta los efectos de apantallamiento $\Phi(r)$ causados por los electrones que rodean los núcleos atómicos. El siguiente potencial representa un esfuerzo para construir un potencial genérico “universal” que tenga en cuenta el apantallamiento, y ha sido creado a partir de múltiples potenciales entre diferentes pares de átomos:

$$\Phi(r) = 0,1818e^{-3,2x} + 0,5099e^{-0,9423x} + 0,2802e^{-0,4029} + 0,02817e^{-0,2016x}.$$

donde x es la división entre la distancia interatómica y la longitud universal de apantallamiento $x = r/a_U$, y

$$a_U = \frac{0,8854a_0}{Z_1^{0,23} + Z_2^{0,23}}.$$

▷ 6.4

Pérdidas inelásticas de energía

Además de perder energía por “scattering” de los átomos del blanco, las partículas incidentes pierden energía a través de un proceso inelástico de excitación de electrones. La sección eficaz para estas pérdidas resulta ser (Lindhard and Schaff, 1961)

$$S_n(E) = \int_0^\infty T d\sigma = \int_0^\infty T(E, p) 2\pi p dp.$$

El programa MARLOWE considera dos modelos para estas pérdidas de energía, un primer y simple modelo, el llamado “no local”, supone que las pérdidas están únicamente relacionadas con la distancia recorrida, y no con las colisiones, y por tanto según dicho modelo

$$S_{\text{local}}(E) = L_m K_L E^{1/2},$$

donde L_m es la distancia recorrida, E es la energía cinética y K_L está representado por

$$K_L = \frac{1,212 z_1^{7/6} Z_2}{(Z_1^{2/3} + Z_2^{2/3})^{3/2} M_1^{1/2}}.$$

El otro modelo es el propuesto por (Oen and Robinson, 1976), donde se establece un vínculo entre las pérdidas de energía y las colisiones. En este modelo la pérdida inelástica es:

$$S_{\text{non local}}(E) = \frac{0,045 * \exp(-0,3r_0/a)}{\pi a^2} K_L E^{1/2}$$

en donde r_0 es la distancia mínima alcanzada entre las dos partículas y

$$a = \frac{0,8853 a_0}{(Z_1^{1/2} + Z_2^{1/2})^{2/3}},$$

siendo a_0 el radio de Bohr.

▷ 6.5

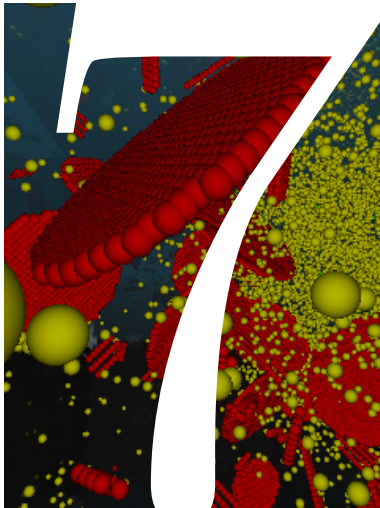
Conclusión

Se ha visto una muy breve introducción al método BCA, donde se han explicado ligeramente las técnicas usadas por el código MARLOWE, que originariamente fueron desarrolladas por (Robinson and Torrens, 1974). Hemos utilizado dicho código para simular todas las implantaciones necesitadas posteriormente por nuestro simulador de difusión DADOS (Jaraiz et al., 2001).

Para obtener información extra sobre este tema se pueden consultar las referencias (Robinson, 1992; Arias, 1995; Hernandez-Mangas, 2000; Jaraiz et al., 1993).

Parte III

El método de Monte Carlo cinético (sin red)



Procesos en Silicio: Mecanismos

| | |
|---|----|
| 7.1. Introducción | 81 |
| 7.2. Breve revisión sobre la difusión y los defectos en Silicio | 82 |
| 7.3. Defectos propios del silicio | 83 |
| 7.4. Los defectos puntuales <i>I</i> y <i>V</i> | 83 |
| 7.5. Defectos extensos de <i>Is</i> y <i>Vs</i> | 86 |
| 7.6. Amorfización y recristalización | 91 |
| 7.7. Intercaras y volumen | 95 |
| 7.8. Átomos de impurezas | 97 |
| 7.9. Procesamiento de dispositivos nanoscópicos | 98 |

El propósito de este capítulo es ofrecer una breve revisión de los hechos que, desde los experimentos hasta la teoría, se conocen acerca de la difusión de defectos puntuales, formación de defectos extendidos, dañado, amorfización y recristalización, difusión de impurezas e interacciones con las intercaras en el procesado del silicio.

Introducción

El procesado de materiales para las nuevas (nano) tecnologías está pidiendo nuevos y comprensibles modelos y esquemas de simulación a escala atómica. Estos nuevos esquemas requieren ser capaces de manejar la extensa variedad de mecanismos de interacción que gobiernan dichos procesos complejos. De hecho, debido a los ímprobos esfuerzos que se están realizando para continuar profundizando en las tecnologías de circuitos integrados (IC) a nivel nanométrico, el procesado del silicio está llegando a niveles de cada vez mayor complejidad técnica. Además, esta situación no puede sino empeorar a medida que se avanza más en dicha investigación, y se la pone al límite de sus posibilidades. El denominado “front-end processing”, en particular, intenta extender las técnicas ya conocidas de dopaje del silicio (implantación iónica más recocido RTP) a los dominios de las técnicas nanoscópicas (por debajo de la micra) pero, aparecen muchos nuevos efectos y mecanismos cuando estas técnicas, que estaban establecidas para dominios mucho mayores, son aplicadas a este nuevo régimen de procesamiento. Además, como muchos de estos efectos suceden al mismo tiempo, la interpretación de los experimentos se vuelve ambigua, y el uso de la simulación predictiva de procesos, se transforma en una necesidad imperativa. El método de Monte Carlo Cinético (kMC) para la simulación y modelado de procesos, se muestra entonces particularmente apto para satisfacer estas necesidades. Consecuentemente, está emergiendo como una de las más valiosas herramientas de simulación para la investigación del procesado avanzado de materiales.

La tradicional aproximación del continuo, basada en resolver ecuaciones en derivadas parciales (PDE), no puede manejar de manera simultánea el cada vez mayor número de diferentes fenómenos que tienen lugar durante la fabricación de dispositivos nanoscópicos. Alternativamente, los cálculos *ab-initio* pueden aportar una descripción libre de parámetros de pequeñas estructuras atómicas, mientras que la dinámica molecular clásica se puede utilizar para determinar el comportamiento de los defectos extendidos, que por otra parte son inaccesibles (por su gran tamaño que se traduciría en tiempo de computación) para los cálculos *ab-initio*. En cualquier caso, la escala temporal asociada a este tipo de investigaciones se encuentra en el orden de los nanosegundos, lo que la excluye de su uso directo en simuladores de procesado de materiales.

Es nuestro propósito mostrar que la técnica de Monte Carlo cinético resulta especialmente útil para rellenar el hueco entre las anteriores técnicas de simulación y la simulación realista del procesado de materiales. El kMC ha demostrado ser la más exitosa herramienta de unión entre los conocimientos de índole teórica obtenidos por *ab-initio* y cálculos de dinámica molecular, y las escalas de tiempo y espacio implicadas en las condiciones de procesamiento de dispositivos actuales.

En este capítulo se hará una revisión concisa de los hechos más relevantes

sobre la difusión y formación de defectos en Silicio, y se asentarán las bases para nuestro particular esquema de simulado kMC. Posteriormente en el capítulo 8 se describirán los modelos y formas de operar del simulador DADOS (Jaraiz et al., 2001) tal y como han sido implementados, para continuar en el capítulo 9 con un sumario de todas las partículas permitidas por DADOS y sus modelos, y ofrecer después, en el capítulo 10 una revisión de ejemplos, tests y múltiples experimentos simulados que ilustran las capacidades de nuestro modelo. Finalizaremos con algunos de los más recientes resultados obtenidos con el programa, expuestos en el capítulo 11.

▷ 7.2

Breve revisión sobre la difusión y los defectos en Silicio

Para recordar o definir de manera más concisa y clara lo que va a ser modelizado, comenzaremos con una breve compilación de algunos de los hechos conocidos (tanto teóricos como experimentales) sobre los defectos y la difusión en Silicio. Esta recopilación no pretende, ni mucho menos, resultar exhaustiva o definitiva, sino simplemente servir como referencia durante la posterior descripción de los modelos atómicos que implementaremos.

Los defectos básicos que aparecen en el Silicio son los defectos puntuales intrínsecos: los intersticiales (I) y las vacantes (V). Su aglomeración lleva a la formación de defectos extendidos, como los $\{311\}$, voids o bucles de dislocación. Además, siempre existen una cierta indeseada cantidad de impurezas, particularmente oxígeno y carbono, que constituyen los defectos extrínsecos nativos. Finalmente, otras impurezas como boro, fósforo, arsénico u otras, son introducidas de manera intencional para dopar o, en cualquier caso, modificar las propiedades intrínsecas del silicio, cuya presencia añade nuevas formas de defectos y mecanismos de interacción. Resumiremos ahora los conocimientos actuales que existen sobre estos defectos intentando usar una metodología intuitiva a la par que descriptiva. Para más detalles, el lector puede acudir a las magníficas revisiones que sobre este tema han realizado (Fahey et al., 1989), (Hu, 1994), (Bracht, 2000), y (Dabrowski, 2000).

▷ 7.3

Defectos propios del silicio

Partiendo de una consideración termodinámica básica, a cualquier temperatura $T > 0K$, incluso un cristal de Silicio perfecto contiene una cierta concentración de intersticiales (I) y vacantes (V) propias. Estos defectos puntuales, como se verá, juegan un papel fundamental en las propiedades (como la difusión) de los dopantes comúnmente usados. Estos defectos marcarán, por tanto, el comienzo de nuestra revisión.

▷ 7.4

Los defectos puntuales I y V

▷ 7.4.1

Configuración atómica

Una vacante es un sitio de red sin átomo. Debido a las distorsiones de la red puede adoptar diferentes configuraciones. Concretamente la configuración es sensible a cambios del estado de carga, que a su vez dependen del nivel de Fermi y, por tanto, del entorno local. La figura 7.1 muestra los enlaces de distintas configuraciones de V s. V^+ y V^- pueden detectarse usando experimentos de resonancia paramagnética de electrones (EPR).

Modelo en la
sec. 8.2,
pág. 109.

Un intersticial propio, es un átomo extra insertado en la red del silicio. Como la V , puede adoptar distintas configuraciones y estados de carga (figuras 7.1(b) y 7.1(c)). Cuando la extensión de la distorsión de la red se expande sobre varios sitios de red (defecto puntual extendido (Seeger and Cheek, 1968)), las propiedades del defecto pueden ser especialmente complejas (Dabrowski, 2000).

Cada configuración tiene unas diferentes energías de formación y migración, que resultan ser las entradas fundamentales de los simuladores kMC. Sencillamente la energía de formación de, por ejemplo la V , establece su concentración de equilibrio, mientras que la energía de migración determina su difusividad. En algunos casos es posible estimar los valores de dichas energías a partir de cálculos ab-initio, pero como medida alternativa se pueden obtener estos parámetros a partir de experimentos en los que se midan tanto concentraciones como difusividades. Hay que tener en cuenta que la versión “simplista” anterior se complica con la presencia de los prefactores, que incluirán términos de entropía.

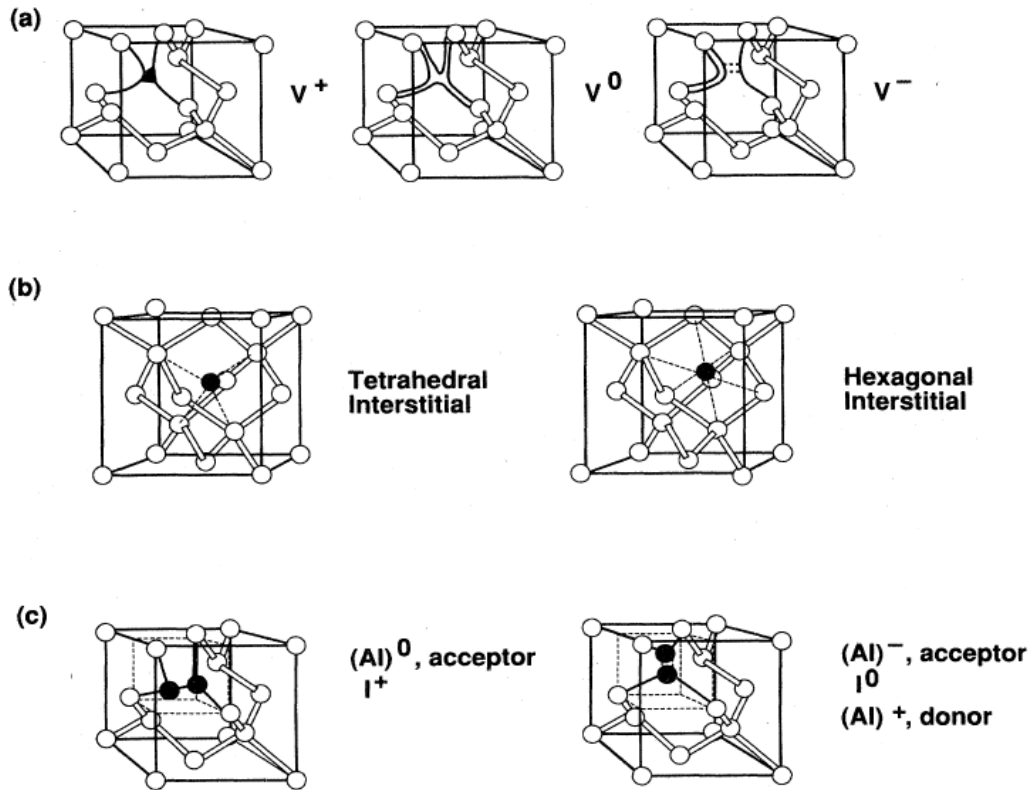


Figura 7.1: Algunas configuraciones para (a) una vacante, (b) un intersticial propio y, (c) una impureza intersticial. Tomado de (Watkins et al., 1979).

| A | $H_A^f + H_A^m$ | $S_A^f + S_A^m$ | H_A^f | S_A^f | H_A^m | S_A^m |
|---|-----------------|-----------------|------------|-----------|------------|-----------|
| I | 4.95 eV | 13.2 k_B | 3.2-3.5 eV | 4-6 k_B | 1.4-1.8 eV | 7-9 k_B |
| V | 4.24 eV | 63 k_B | 2.0-4.0 eV | 1-6 k_B | 0.2-2.2 eV | 0-6 k_B |

Tabla 7.1: Propiedades termodinámicas de los intersticiales propios (I) y las vacantes (V), obtenidas de estudios experimentales y teóricos. Referencia (Bracht, 2000).

▷ 7.4.2

Difusividades y concentraciones en equilibrio

La tabla 7.1 resume los valores aceptados para los parámetros fundamentales de los intersticiales y vacantes, conseguidos a través de estudios tanto teóricos como experimentales. Aunque la suma de la energía de formación más los términos de entropía o entalpía de migración están bastante bien caracterizados y establecidos, no ha sido posible aún conciliar los resultados de diferentes experimentos en los cuales se separan estas dos contribuciones de formación y migración, especialmente para las vacantes. Por ejemplo $H_{V,f} + H_{V,m} = 4,14$ eV (experimento), 4.03 eV (experimento (Bracht et al., 1998)), 4.07 eV (teoría (Tang et al., 1997)). De acuerdo con esto, $H_{V,f}$ está cerca de los 3.3 eV para cálculos teóricos (Casali et al., 2001) y 3.6 eV para experimentos de tiempo de vida de positrones (Dannefaer et al., 1986), y $H_{V,m}$ es menor que los 0.5 eV para ciertos cálculos (Blöchl et al., 1993) y para resultados de recocidos a baja temperatura de dañado iónico (Watkins et al., 1979). No obstante, medidas a altas temperaturas (Tan and Gösele, 1985; Zimmermann and Ryssel, 1992; Bracht et al., 1995; Mogi et al., 1996) producen valores de $H_{V,m}$ cerca de 2 eV. Como ya se menciona en la referencia (Bracht, 2000), el conocimiento de estas contribuciones separadas no es estrictamente necesario bajo condiciones de equilibrio, pero puede afectar de manera definitiva el desarrollo de una simulación bajo condiciones de no equilibrio, las cuales desgraciadamente suelen prevalecer en condiciones de fabricación de dispositivos nanométricos. Es por tanto necesario clarificar estos descubrimientos contradictorios, y dicha tarea debería tener una alta prioridad si se quiere avanzar en el modelizado de procesos.

▷ 7.4.3

Efectos debidos al nivel de Fermi

Como se muestra en la figura 7.1, los defectos puntuales pueden existir bajo diferentes estados de carga. La concentración de defectos en un estado de carga negativo, relativa al estado neutral, depende de la posición del nivel energético del defecto puntual, así como de la posición del nivel de Fermi. Como se comenta en (Fahey et al., 1989):

Modelo en la
sec. 8.8,
pág. 157.

$$\frac{C_{X^-}}{C_{X^0}} = \frac{g_{X^-}}{g_{X^0}} \exp\left(-\frac{E_{X^-} - E_f}{k_B T}\right) \quad (7.1)$$

Los términos g_X son el número de grados de libertad internos para los defectos, y E_{X^-} sería su nivel de energía en el gap. Los niveles de altísimo dopado que se encuentran en los dispositivos de última generación están demasiado a menudo por encima de la concentración intrínseca de portadores n_i a las temperaturas de difusión, de tal forma que el nivel de Fermi es controlado por las concentraciones locales de dopantes. Además, puesto que los procesos de cambio de carga son

mucho más rápidos que los de difusión, el cociente expresado anteriormente se produce de manera local. Como se mostrará más adelante, en la sección 8.8, este requerimiento es fácilmente implementable en un simulador kMC.

▷ 7.5

Defectos extensos de I s y V s

Modelo en la sec. 8.3, pág. 114. La implantación iónica se usa hoy en día de una manera rutinaria en la tecnología del Silicio para distintos procesos, como el dopaje o la amorfización. Dicha implantación iónica daña severamente el cristal. Desde un punto de vista del procesado microelectrónico, dicho dañado puede clasificarse en dos categorías, dependiendo de si se alcanza o no el nivel de amorfización y, de momento, consideraremos únicamente las situaciones en las cuales no se amorfiza la red. En este caso, posteriores tratamientos del cristal a alta temperatura transforman el dañado del cristal en defectos extendidos, separados y bien localizados, gracias a la aglomeración de defectos puntuales, o bien al recocido de regiones de alto dañado. Cuando la temperatura es suficientemente alta, recocidos prolongados pueden llegar a disolver los defectos extendidos a través de la continua emisión de defectos puntuales. A su vez, el flujo causado por el movimiento de dichos defectos puntuales pueden inducir la difusión o desactivación de dopantes. Los exigentes requerimientos impuestos en los perfiles muy abruptos que la industria microelectrónica requiere hoy en día, demandan, por tanto, el mínimo movimiento de dopantes debido a causas térmicas. Así pues es más que deseable predecir la temperatura y tiempos mínimos necesarios para hacer desaparecer el dañado creado por la implantación. Esto supone un detallado conocimiento de los defectos concretos que van a ir apareciendo a lo largo de la simulación, así como de sus energías de emisión de defectos puntuales. Por si pareciera poco, en el caso de los defectos extendidos, su geometría particular tiene a veces un papel decisivo en las frecuencias de emisión y captura de defectos puntuales. Revisaremos por tanto a continuación algunos de estos aspectos para los defectos extendidos más comunes asociados a los I y V s que se producen en el Silicio.

▷ 7.5.1

Clusters de intersticiales

Modelo en la sec. 8.3.1, pág. 114. Los pequeños clusters (aglomeraciones) de I no parecen tener un comportamiento lineal para su energía ni forma respecto a su tamaño como se deduce de los cálculos (Arai and Takeda, 1997; Chichkine et al., 2002) y de modelos de análisis inverso de datos de difusión (Covern et al., 1999c). La oscilación de sus energías de formación o enlace se muestran en la figura 7.2, y son bastante

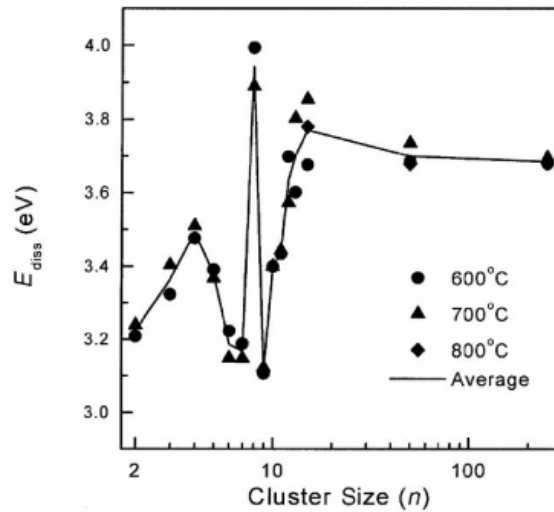


Figura 7.2: Energía de disociación para pequeños clusters de intersticiales propios. Tomado de (Cowern et al., 1999c).

importantes porque dichas oscilaciones pueden actuar como un cuello de botella durante el crecimiento de los clusters en el que los pequeños e inestables clusters se van disolviendo mientras que los grandes, generalmente más estables, crecen a costa de estos pequeños (proceso conocido como Ostwald ripening). Los clusters pequeños son difíciles de apreciar por microscopía pero, afortunadamente, nuestro modelo de simulación atómica no necesita una detallada información de la forma de estos pequeños clusters, cosa que sí es esencial para los grandes. Cuando las dosis no son bajas y el recocido es suficientemente largo, muchos de estos pequeños clusters pueden crecer y evolucionar (Cowern et al., 1999c; Eaglesham et al., 1994) hacia unos pocos clusters grandes, con un carácter cristalográfico perfectamente definido: son los defectos $\{311\}$ que se muestran en la figura 7.3.

▷ 7.5.2

Los defectos de intersticiales $\{311\}$

Los defectos $\{311\}$ (Fig. 7.3) han sido sometidos a una investigación intensa en los últimos años debido a su íntima asociación con el fenómeno de la difusión aumentada del boro (transient enhanced diffusion TED) (Eaglesham et al., 1994). Estos defectos son como barras rectangulares de intersticiales depositados en un plano $\{311\}$ a lo largo de una dirección $\langle 110 \rangle$. Un modelo atómico para la estructura de estos defectos $\{311\}$ fue derivado experimentalmente, a partir de gráficos tomados con microscopía electrónica de alta resolución, por (Takeda, 1991). Estudios de índole más teórico (Kohyama and Takeda, 1992; Kim et al., 1997; Alippi and Colombo, 2000) parecen confirmar la estabilidad de la estruc-

Modelo en la
sec. 8.3.2,
pág. 118.

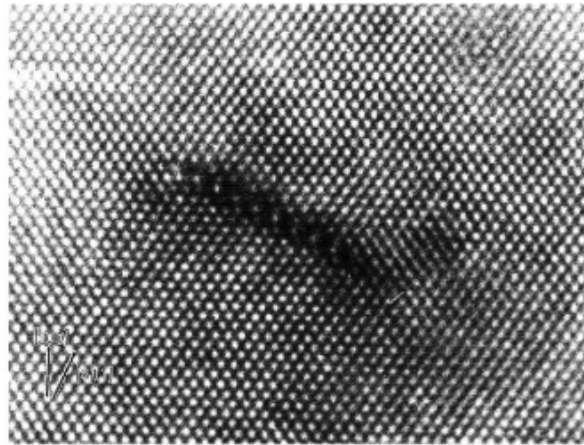


Figura 7.3: Corte de una vista por microscopía TEM de un defecto extendido de intersticiales tipo $\{311\}$. El defecto es como una cinta alargada. Ref. (Eaglesham et al., 1994; Stolck et al., 1997).

tura propuesta, y llevan a valores energéticos que concuerdan aceptablemente bien con los mostrados en la figura 7.2 para clusters de grandes tamaños. Recopilaciones de datos como éstas son valiosísimas para el modelizado de procesos atomístico, porque gracias a ellos los defectos $\{311\}$ pueden ser incorporados a las simulaciones sin la necesidad de hipótesis foráneas de simplificación.

▷ 7.5.3

Discos de dislocación

Modelo en la sec. 8.3.3, pág. 121. Cuando la dosis y la temperatura son suficientemente altas aparece una nueva forma de defecto extendido: el disco de dislocación (dislocation loop). Estos discos de dislocación (Fig. 7.4), son defectos planos asentados sobre un plano $\{111\}$ (de Mauduit et al., 1994) y pueden ser “imperfectos” (faulted, FDL) o “perfectos” (PDL). Los FDL consisten en un apilamiento circular rodeado de una línea de dislocación. Por otra parte, los PDL tienen simplemente un plano extra de átomos en vez del apilamiento y están alargados en la dirección $\langle 110 \rangle$. La figura 7.5 representa las energías de formación de ambos FDL y PDL como una función del área del defecto. Como ya se ha comentado con anterioridad, el conocimiento de la geometría y energías permite la directa inclusión de estos defectos en un simulador atomístico de procesos.

Es curioso observar esta continua evolución de las morfologías de los defectos a medida que un recocido a alta temperatura avanza. Los defectos puntuales móviles empiezan a agruparse prontamente para dar lugar a pequeños clusters

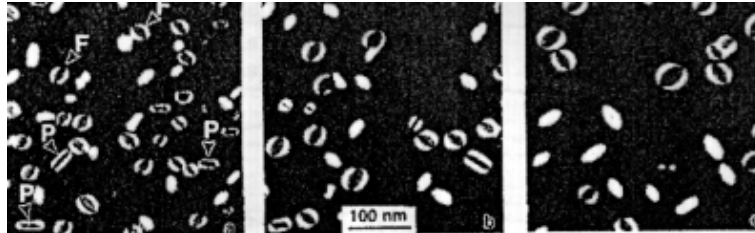


Figura 7.4: Vista por microscopía TEM de los discos de dislocación. Tomado de la referencia (Cristiano et al., 2000).

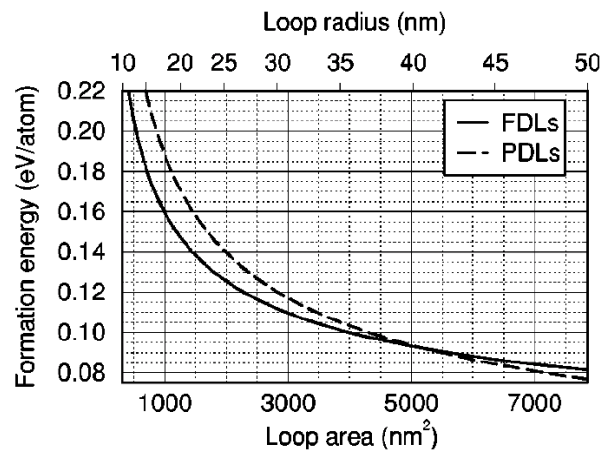


Figura 7.5: Energías de formación para los discos de dislocación imperfectos (faulted FDL) y perfectos (PDL). Tomado de (Colombeau et al., 2001).

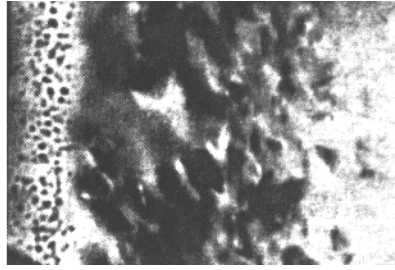


Figura 7.6: Vista de los voids (huecos) por microscopía TEM. Ref. (Holland and White, 1991).

inmóviles de I . A partir de entonces, la maduración y evolución de estos defectos comienza: mientras que los más pequeños (con cocientes de emisión/captura muy grandes) empiezan a disolverse y desaparecer, los más estables crecen y terminan siendo $\{311\}$ s. Cuando la superficie libre (que captura intersticiales) está lejos y la dosis es suficientemente grande, estos $\{311\}$ pueden evolucionar hacia los discos de dislocación “imperfectos” FDL, que son más estables que los $\{311\}$ y, finalmente, hacia los PDL que, como se muestra en la figura 7.5 tienen una energía de formación por átomo menor que los FDL a partir de un cierto tamaño límite.

▷ 7.5.4

Defectos extendidos de V en el Silicio

Pequeños clusters de vacantes han resultado identificados experimentalmente usando resonancia paramagnética de electrones (V , (Watkins, 1992) V_2 , (Watkins and Corbett, 1965) V_3 , V_4 y V_5 (Lee and Corbett, 1974)) y por fotoluminiscencia (V_6) (Hourahine et al., 2000). Estudios teóricos (Chadi and Chang, 1988; Estreicher et al., 1997) indican que el cluster de vacantes con forma de anillo hexagonal debería ser particularmente estable.

▷ 7.5.5

Voids (huecos)

Los clusters de vacantes que son lo suficientemente grandes para ser vistos por microscopía electrónica aparecen como huecos esféricos. (Fig. 7.6, tomada de (Holland and White, 1991)). No obstante, para tamaños muy grande (cerca de 150 nm) su forma se convierte en octaédrica (Falster and Voronkov, 2000; Ueki et al., 1997) debido a la formación de caras internas en la dirección $\langle 111 \rangle$. Las energías de enlace hasta un tamaño de 35 vacantes se han calculado usando dinámica molecular de tipo “tight binding” (Fig. 7.7, referencia (Bongiorno and Colombo, 1998)).

Modelo en la
sec. 8.3.4,
pág. 125.

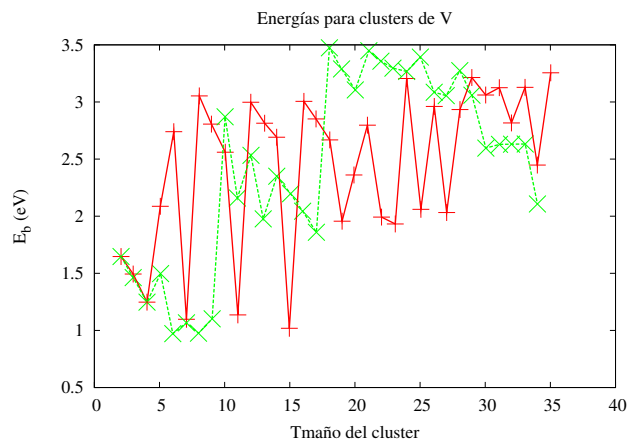


Figura 7.7: Energía de enlace de clusters de vacantes en función de su tamaño, calculada con "Tight-Binding". Ref. (Bongiorno and Colombo, 1998).

▷ 7.6

Amorfización y recristalización

La naturaleza y conducta global que caracteriza al daño generado por implantación en Silicio es un tema bastante complejo que está siendo objeto de una intensa investigación. La irradiación electrónica y la implantación de iones muy ligeros pueden generar defectos puntuales aislados (I , V) pero, por otra parte, la implantación de iones muy pesados puede incluso llegar a derretir parcialmente el silicio en el interior de una cascada de implantación, como demuestran algunas simulaciones MD. Estas regiones derretidas solidifican después muy rápidamente, dejando regiones con un alto nivel de daño, a las que comúnmente se llaman *pockets amorfos* (de la Rubia and Gilmer, 1995). Por regla general, para iones con una masa media, que son los empleados en microelectrónica, se observa una interesante combinación de ambos tipos de efectos. Por tanto, en una implantación a temperatura ambiente, las sucesivas cascadas de implantación pueden ir acumulándose y situándose en regiones que ya estaban previamente dañadas. Para dosis suficientemente altas la muestra puede, eventualmente, amorfizarse.

Entre otras aplicaciones, se ha encontrado que estos fenómenos de amorfización resultan interesantes en el campo de la microelectrónica para prevenir el acanalamiento durante las implantaciones. Además, las regiones amorfas que se encuentran en contacto con otras cristalinas recristalizan muy fácilmente cuando son sometidas a recocidos de alta temperatura.

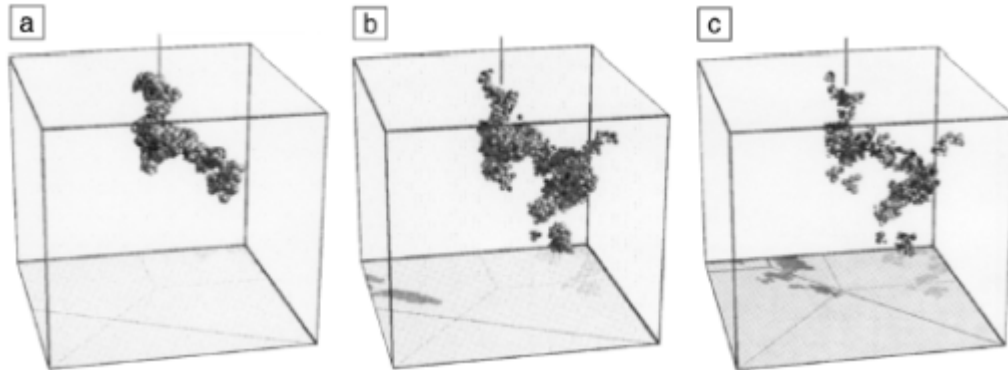


Figura 7.8: Simulación por dinámica molecular de una serie de cascadas de iones de Si a 5 keV después de un recocido a alta temperatura. a) 0.1 ps, b) 2.5 ps y c) 9 ps. Hay $1,6 \cdot 10^5$ átomos en la simulación, pero solamente los átomos con una energía potencial más grande que 0.2 eV están dibujados. Tomado de la referencia (Law et al., 2000).

▷ 7.6.1

Regiones desordenadas (amorphous pockets)

Modelo en la
sec. 8.4.1,
pág. 127.

La dinámica molecular (MD) únicamente es capaz de simular muy pequeñas escalas de tiempo y, por tanto, no es capaz de predecir el estado final de un pocket amorfo para las temperaturas y tiempos de interés práctico en microelectrónica. No obstante, MD ha demostrado ser de gran utilidad para defectos puntuales y algunos tipos específicos de defectos extendidos debido al número limitado, y la discreta cantidad de configuraciones y energías involucradas. Además, se ha desarrollado un método (Voter, 1997) para extender las técnicas MD en varios órdenes de magnitud en los casos en los que ciertos mecanismos particulares dominen la simulación. Lamentablemente, para los pockets amorfos existe una continua, casi ilimitada, distribución de configuraciones y energías que imposibilitan el uso del mencionado método. Como un ejemplo aproximado, las simulaciones MD pueden ser lanzadas a temperaturas muy altas para ver como los pockets amorfos se colapsan y transforman en clusters aislados de defectos puntuales, (Fig. 7.8, sacada de la referencia (Law et al., 2000)), pero estos resultados no pueden simplemente extrapolarse a temperatura ambiente, porque los mecanismos elementales implicados serán distintos dependiendo de las energías de activación, y por tanto el resultado final podría ser diferente del predicho. Algunas simulaciones de dinámica molecular (Marques et al., 2001) realizadas para estudiar la estabilidad de la red cristalina del silicio a medida que se aumenta el número de intersticiales en su interior, indican que ésta se colapsa, transformándose en una región amorfa, cuando el exceso de intersticiales alcanza aproximadamente el 25 % de la concentración atómica del silicio.

Como aún no disponemos de modelos predictivos para el recocido dinámico y

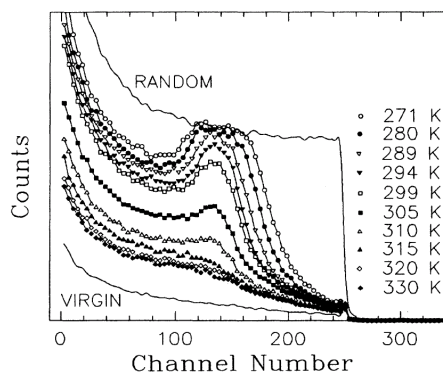


Figura 7.9: Espectro RBS de acanalamiento para implantaciones de Si en Si con 1 MeV de energía, dosis de 10^{15} cm^{-2} y las diferentes temperaturas de sustrato indicadas en la figura. Hay una dependencia crítica, que está cerca de la temperatura ambiente. Tomado de (Schultz et al., 1991).

la amorfización, se han construido modelos empíricos a partir de los datos experimentales, datos de los que, por otra parte, hay gran abundancia. No obstante, como se muestra en la figura 7.9, hay una dependencia crítica con la temperatura, para implantaciones con valores típicos de dosis, energías y rapidez de implantación, precisamente cerca de la temperatura ambiente. Los experimentos realizados para dilucidar esta sensibilidad (tipo beam heating/sample mounting) ofrecen en cambio resultados contradictorios. La energía de activación para la recombinación del dañado a temperatura ambiente parecen ser 0.9 eV (Schultz et al., 1991), que se opone al valor más elevado de 1.2 eV medido a temperaturas más altas (400-550 K) (Elliman et al., 1988). Estos resultados podrían incidir en la idea de que la amorfización es el resultado de la evolución dinámica de las regiones altamente dañadas, y que por tanto podría tener un rango continuo de energías de activación. A temperaturas más altas, las bajas energías de activación de los procesos hacen que éstos sucedan muy rápidamente, y ello podría ser reflejado en esos valores más altos.

Nuevamente, las simulaciones MDs (de la Rubia and Gilmer, 1995) indican que la recombinación de los pockets amorfos es un proceso de baja energía, que tiene lugar principalmente debido a la reordenación interna de sus pares IV más que a una emisión de defectos puntuales. Únicamente cuando al pocket amorfo le queda ya un exceso de I o V , es cuando comienza la emisión de defectos puntuales. Cuando éste es el caso, los intentos de modelizar la pre-amorfización (o interpretar experimentos de amorfización) a través de una representación de clusters, podrían fallar, muy especialmente a bajas temperaturas. Por otra parte, a temperaturas altas los clusters son detectados experimentalmente como huecos de V s o discos de dislocaciones de intersticiales, como ya se mostró en la figura 7.6. Este patrón de dañado (clusters de V cerca de la superficie, y de I s en su

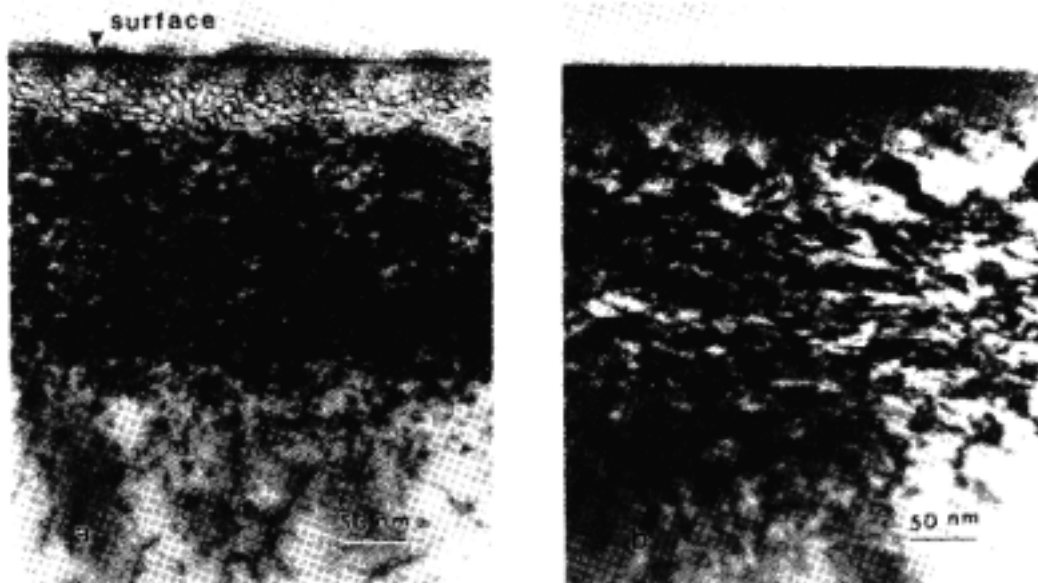


Figura 7.10: Patrones de dañado de distinto tipo, que han surgido de la implantación de dos iones, con casi la misma masa (a la izquierda As y a la derecha Ge), y que sugieren que los efectos de tipo químico o eléctrico pueden jugar de un papel central a la hora de dilucidar la morfología de los procesos de dañado. Tomado de la referencia (Holland et al., 1996).

profundidad) se puede entender fácilmente teniendo en cuenta el momento del ion implantado aunque, no obstante, efectos de tipo químico podrían también tener un papel crítico, como resulta obvio de la comparación entre dos iones, con casi la misma masa pero siendo uno un dopante y el otro un material isoeléctrico, que se muestra en la figura 7.10. Resultados similares han sido obtenidos para P y Si (Ref. (Holland et al., 1996)).

Modelo en la
sec. 8.4.1,
pág. 127.

Así pues el modelado de la amorfización usualmente supone dos tareas:

1. Modelar el proceso de acumulación de dañado, teniendo en cuenta los mecanismos de recocido dinámico y
2. establecer un umbral de dañado a partir del cual comience la amorfización.

▷ 7.6.2

Recristalización

Modelo en la
sec. 8.5,
pág. 134.

La recristalización de las regiones amorfas que se encuentran en contacto con regiones cristalinas sigue una ley de Arrhenius para el rango de temperaturas comprendido entre 400 y 1360 °C (Roorda and Sinke, 1989). Durante la recristalización, bajas condiciones de procesamiento típicas, la interfaz amorfo-cristal va

avanzando rápidamente, barriendo prácticamente los defectos puntuales y dejando una red casi perfecta, con átomos de impurezas pero sin apenas I s o V s.

▷ 7.7

Intercaras y volumen

▷ 7.7.1

Condiciones de equilibrio

Una intercara de Si (bien una superficie libre o una interfaz con otro material, como óxido o nitruro de Silicio), puede ser considerada como una clase especial de defecto extendido. Así, en clara analogía con estos otros defectos, puede emitir y capturar defectos puntuales, que serán I s y V s para una superficie libre. Las frecuencias de emisión y migración para estos I s y V s se puede calcular a partir de los parámetros expuestos en la tabla 7.1. A su vez, las frecuencias de captura dependerán también de las secciones de captura, las cuales generalmente se asume que son iguales a la distancia de salto (λ), que coincide con la distancia a segundos vecinos en la red del Silicio. Cuando la temperatura aumenta, las frecuencias de I s y V s también aumentan, por lo que la concentración de estas especies sufrirá un incremento hasta que, nuevamente, se equilibren las frecuencias de captura y emisión superficial. Existe también una cierta evidencia experimental sobre que las superficies libres en el Silicio no son un sumidero perfecto para los intersticiales, pudiendo rechazar algunos de éstos. Este comportamiento se caracteriza a través de una longitud de recombinación ($L \approx 0,1\mu m$) (Lim et al., 1995; Covern et al., 1999a). Este fenómeno, visto desde un punto de vista atomístico, implica que hay una probabilidad λ/L para que un I sea atrapado cuando alcanza la superficie, siempre que $L > \lambda$ (si no sería atrapado siempre).

Modelo en la
sec. 8.7,
pág. 149.

Los intersticiales y las vacantes pueden, a su vez, recombinarse dentro del volumen, así como ser generados en él, pero para dispositivos nanoscópicos los efectos de la superficie habitualmente prevalecen sobre el volumen y, por esta razón, en una primera aproximación se desprecian estas generaciones en volumen.

▷ 7.7.2

Condiciones de no equilibrio: Oxidación y nitridación

Es un hecho conocido que las reacciones que se producen en la superficie del silicio pueden alterar las frecuencias netas de generación y recombinación de defectos puntuales. Los casos más comunes incluyen la oxidación, nitridación y silicidación. Puesto que aún no hay modelos teóricos cuantitativos claramente establecidos para estos procesos, se recurre al uso directo de datos experimen-

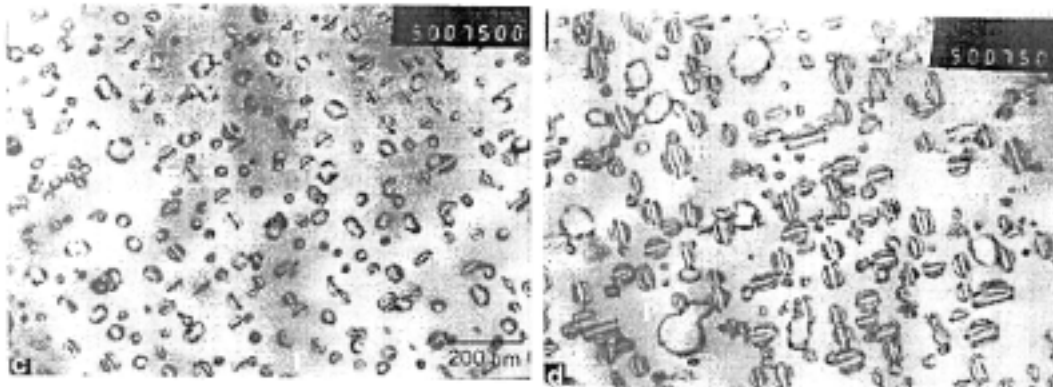


Figura 7.11: Bucles de dislocación después de a) 90 min. de recocido a 900°C en un ambiente inerte y b) en un ambiente oxidante. La comparación tanto del número como del tamaño de los bucles permite hacer una estimación de la frecuencia de emisión de I_s desde la superficie oxidante a la temperatura dada. Tomado de la referencia (Skarlatos et al., 1999).

tales para la creación de modelos empíricos. La observación de que, bajo ciertas condiciones, los defectos de apilamiento crecen durante la oxidación, pero se reducen en la nitridación, llevan a concluir que, en estas particulares condiciones, la oxidación inyecta mayoritariamente I_s frente a la nitridación, que lo hace con las V_s .

Como un ejemplo de las metodologías que se siguen para extraer los parámetros necesarios para la gestión de los modelos (Skarlatos et al., 1999), la comparación de las microfotografías TEM expuestas en la figura 7.11 permiten estimar cuantitativamente el número de I_s capturado por los discos de dislocación durante una oxidación seca, cosa que se hace al tener en cuenta el número de átomos de silicio por unidad de área que contienen dichos discos (Cristiano et al., 2000). Los mencionados números de I_s en los discos pueden usarse, junto con el necesario modelizado (Skarlatos et al., 1999; Hu, 1981) para estimar la supersaturación promedio de I_s producida por la superficie durante la oxidación seca.

Respecto a la formación de siliciuros, ésta puede generar una supersaturación de V_s mantenida en el orden de 2 a 3, junto con un descenso de la concentración de I de cerca de 0.3 (Herner et al., 1998). Por último, descubrimientos recientes muestran que capas superficiales de boruros (Agarwall et al., 1999) generan una supersaturación de I del orden de 4.

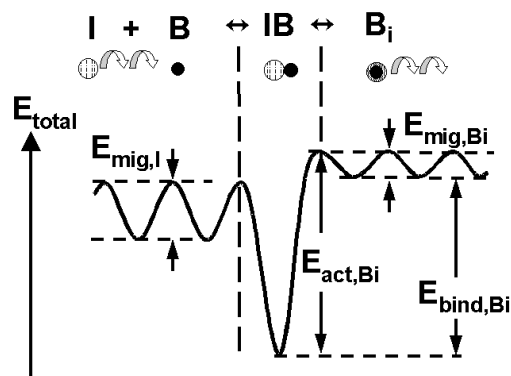


Figura 7.12: Esquema de la formación, migración y ruptura de un defecto puntual emparejado (mecanismo kick-out).

▷ 7.8

Átomos de impurezas

Hasta ahora nos hemos preocupado solo de la presencia (o ausencia) de átomos de Silicio pero, además de los dopantes que de manera intencionada se introducen en dicho material (B, P, As) es habitual la presencia de otras impurezas indeseadas, que están siempre presentes hasta un cierto nivel incluso en procesos de generación muy depurados, como es el caso del *C* y el *O*. Describiremos por tanto ahora, de manera bastante concisa, las reacciones de estos otros átomos con los defectos puntuales y sus fenómenos de agrupamiento y precipitación.

Modelo en la
sec. 8.6,
pág. 139.

▷ 7.8.1

Procesos de atrapamiento y reemplazo (kick-out)

La figura Fig. 7.12 muestra una visión bastante general (Fahey et al., 1989; Cowern and Rafferty, 2000) de la interacción de un defecto puntual (*I* o *V*) con otro tipo de átomo. En este ejemplo, un intersticial que se estaba moviendo encuentra en su camino un átomo de boro en posición sustitucional, y se enlaza con él formando un par *IB* que podría o no ser móvil. Generalmente, estos defectos puntuales emparejados, pueden ejecutar distintos mecanismos.

Por ejemplo, la difusión del boro, que ha sido objeto de una intensa investigación en estos últimos años, parece responder al esquema indicado más arriba (Cowern et al., 1990). Para el carbono y el oxígeno, que son generalmente las impurezas “de fondo” más habituales, estudios tanto teóricos como experimentales muestran que también capturan *I*s (el carbono) y *V*s (el oxígeno), modificando por tanto las difusividades de dichas especies (Casali et al., 2001; Stolk et al., 1997), al constituir una “trampa” para ellas. Este tipo de fenómenos dificulta la

Modelo en la
sec. 8.6.1,
pág. 139.

adecuada predicción y control de la difusión de los dopantes.

▷ 7.8.2

Formación y precipitación de impurezas

Modelo en la sec. 8.6.2, pág. 143.

Los mecanismos dados en la figura 7.12 resultan ser insuficientes, bajo ciertas circunstancias, para explicar las observaciones experimentales. Por ejemplo, se sabe que los dopantes son eléctricamente inactivos a partir de un cierto nivel de concentración (Fahey et al., 1989) y también se ha demostrado que altas concentraciones de I_s pueden llevar a desactivar eléctricamente una fracción de los B sustitucionales, incluso cuando su concentración está por debajo de su solubilidad (Stolk et al., 1995). Para explicar estos hechos ha sido necesario acudir a la existencia de un fenómeno de “clustering” (Hu, 1973) del boro con intersticiales para formar $I_n B_m$ (Pelaz et al., 1997). Así mismo, la desactivación eléctrica puede deberse también a una precipitación (Fahey et al., 1989) de los dopantes. Personalmente, nosotros creemos que, así como se sugiere y comenta para los clusters de AsV en (Fahey et al., 1989), los pequeños clusters del dopante “X” con I_s pueden resultar los embriones de precipitados SiX y, de esta manera, los fenómenos de formación de clusters y precipitación ocurrirían a la vez, pero sería uno u otro el fenómeno dominante según las condiciones experimentales.

▷ 7.9

Procesamiento de dispositivos nanoscópicos

Hasta el momento, se ha expuesto uno por uno una variedad de posibles fenómenos que tienen lugar durante el procesamiento del silicio. Algunos de ellos son, por sí mismos, lo suficientemente complejos como para resultar todo un reto para su explicación y posterior simulación predictiva basada en la física inherente a ellos. Pero nos vemos obligados a señalar que, la tecnología de fabricación de dispositivos lleva en muchas ocasiones, como la que se intenta ilustrar en la figura 7.13, a situaciones en la que más de uno, por no decir todos, de estos procesos, suceden a la vez. El dibujo intenta representar un drenador y una fuente (S/D) de tamaño $0,1 \mu m$ en un NMOSFET, durante un recocido. La implantación de arsénico genera una región amorfa, que rápidamente cristaliza dejando en su camino únicamente las impurezas externas. En la interfaz amorfo-cristalina se han formado una serie de clusters y defectos de tipo $\{311\}$, que en su disolución producen TED y clusters de impurezas con el boro presente en el canal del transistor. A su vez, la alta concentración de As implantado induce dependencias de la difusión con el nivel de Fermi y, por si pareciera poco, también es importante tener en cuenta que, por las pequeñas dimensiones del dispositivo involucrado, se

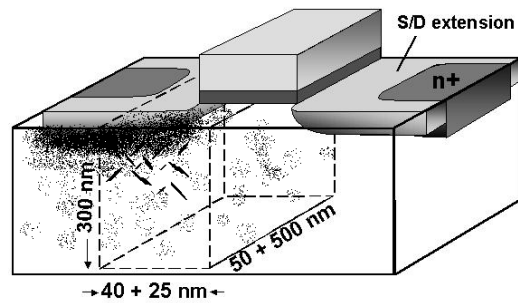
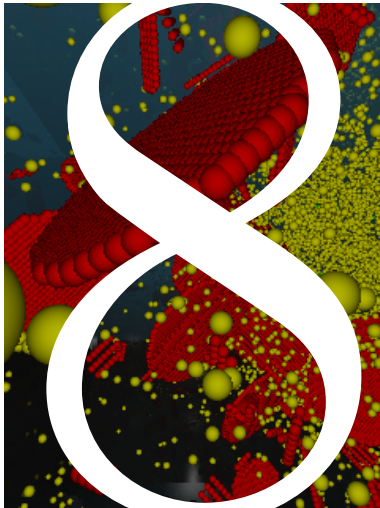


Figura 7.13: Región de simulación de drenador y fuente para un transistor NMOSFET durante un recocido.

requiere una simulación en 2 o 3 dimensiones para poder extraer efectos realistas. Finalmente, la naturaleza discreta de los dopantes y los defectos puntuales junto con sus posibles estados de carga, puede introducir correlaciones en su distribución espacial, con lo que, todo esto, deberá ser adecuadamente tenido en cuenta y simulado, como veremos en el resto de esta serie de capítulos.



Implementación atomística de los mo- delos físicos.

| | |
|---|-----|
| 8.1. Introducción al funcionamiento de DADOS. | 103 |
| 8.2. Defectos puntuales I y V | 109 |
| 8.3. Defectos extendidos de Is y Vs | 114 |
| 8.4. Acumulación de daño | 126 |
| 8.5. Recristalización | 134 |
| 8.6. Impurezas | 139 |
| 8.7. Intercaras entre materiales | 149 |
| 8.8. Modelo de carga | 157 |
| 8.9. Comandos de DADOS | 175 |

El propósito de esta sección es mostrar la forma en que los mecanismos físicos expuestos en la sección previa han sido incorporados y modelizados en el simulador de procesos en silicio por Monte Carlo cinético DADOS. Básicamente se hace una detallada descripción de los diferentes tipos de defectos, así como de las interacciones y reacciones existentes entre ellos. También se modelizan la recristalización y acumulación de daño, las dependencias con el nivel de Fermi y las reacciones en las intercaras.

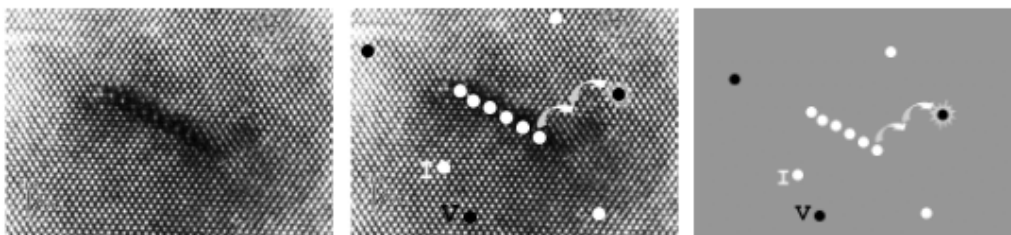


Figura 8.1: En la técnica kMC no se incluyen los átomos de la red, sino que tan sólo los átomos pertenecientes a los defectos son tomados en cuenta.

▷ 8.1

Introducción al funcionamiento de DADOS.

DADOS (*Diffusion of Atomistic Defects, Object Oriented Simulator*), es un simulador atomístico tridimensional de difusión y defectos en silicio, basado en la técnica de Monte Carlo cinético (kMC). La figura 8.1 ilustra el concepto que hay detrás de la aproximación kMC: la imagen del principio (tomada de (Stolk et al., 1997)) muestra una microfotografía de alta resolución tomada mediante TEM de un defecto $\{311\}$ extenso que se encuentra embebido entre el cristal de silicio. En MD se simularían todos los átomos de la red y de los defectos, pero en kMC únicamente se simularán los que pertenezcan a defectos puntuales o extensos (representados como círculos sobre-impresos en la fotografía de la figura 8.1 (imagen del centro)). Así pues, en la muestra representada todos los átomos de la red se encontrarían vibrando (con un periodo cercano a 10^{-13} s) y, únicamente de vez en cuando (por ejemplo cada 10^{-9} s), alguno de los defectos aislados (defectos puntuales), saltarían a una posición vecina, donde podrían quizás ser capturados por algún defecto extenso próximo. A intervalos de tiempo aún más largo (por ejemplo, cada 10^{-3} s) emitiría el defecto extenso algún defecto puntual.

Para simular esto DADOS no tiene en cuenta ningún átomo de la red (fondo de la figura 8.1). Así DADOS comenzaría a simular tiempos cercanos a 10^{-9} s. Además, cuando los defectos puntuales, que se mueven muy deprisa, desaparecan, quedará sólo el defecto puntual extendido, con lo que los pasos de tiempo podrían ser incrementados hasta 10^{-3} s.

Las partículas sencillas pueden moverse solas (defectos puntuales), o bien pertenecer a defectos extendidos (como un $\{311\}$). Cuando una partícula se mueve sola DADOS la denomina *defecto puntual*, poniendo dicha categoría en su etiqueta de tipo de defecto. Los defectos puntuales propios del silicio, es decir, los intersticiales y las vacantes, serán explicados a continuación, en la sección 8.2, dejando la sección 8.6.1 para una explicación del modelo usado para los dopantes que se

muevan solos, o bien a través de mecanismos intersticiales o vacantes.

Por último, los intersticiales, vacantes e impurezas pueden ser atrapados en diferentes tipos de defectos extendidos. Los defectos extendidos de partículas propias del silicio (aglomeraciones de I_s o V_s) van a ser tratados en la sección 8.3, y los clusters que contengan también impurezas o dopantes estarán en 8.6.2.

▷ 8.1.1

Unidades

Las unidades usadas en el programa DADOS de forma habitual, y las que también usaremos nosotros preferentemente, son:

Espacio Nanómetros (nm).

Tiempo Segundos (s).

Energía Electrón-voltio, (eV).

Difusividad ($\text{cm}^{-2} \text{s}^{-1}$)

Concentración átomos por cm^3 .

Las unidades anteriores son las estándar utilizadas en simuladores de este tipo, tanto atomísticos como de continuo, de tal manera que no parece que su elección requiera de ninguna explicación extra. No obstante, DADOS utiliza una medida poco convencional para la frecuencia, que por otra parte es una magnitud vital para el buen funcionamiento de un simulador de Monte Carlo cinético, puesto que los diferentes *eventos* de la simulación estarán programados según su frecuencia de actuación. Un vistazo a la tabla 8.7, por ejemplo, nos hará ver que los prefactores de emisión de un cluster y, por tanto, su frecuencia, se encuentra en unidades de difusividad. Se escogió esta unidad por coherencia con los prefactores de migración, que están en unidades de difusividad, para facilitar su comparación con cálculos teóricos y experimentales, y poder poner así todos los prefactores en la misma unidad. Incluso así, está claro que se necesita pasar de estas unidades de difusividad a frecuencia, cosa que DADOS hace para calcular los tiempos. Para ello se usa la ley de Fick

Más detalles
en sec. A.1.2,
pág. 325.

$$\frac{\partial n}{\partial t} = D \frac{\partial^2 n}{\partial x^2}$$

donde

$$D = \frac{1}{6} \frac{\lambda^2}{\tau},$$

siendo λ la distancia de salto media y τ el tiempo medio consumido en uno de estos *saltos*. Con esto, es fácil observar que la conversión se hace con la ecuación:

$$\nu = \frac{6D}{\lambda^2}.$$

▷ 8.1.2

Partículas y defectos puntuales y extensos en DADOS

DADOS asigna cinco variables numéricas a cada una de las partículas. De éstas, tres son utilizadas para almacenar las coordenadas x , y , y z , y las otras para representar el *tipo de defecto* y el *tipo de partícula*.

El tipo de defecto es una etiqueta especial para conocer a qué tipo de estructura especial o defecto extendido pertenece la partícula. De esta forma, una partícula puede moverse sola (ser un *defecto puntual*) o estar en un *hueco*, en un $\{311\}$, en un *cluster* de impurezas, etc.

Por último el tipo de partícula se etiqueta relacionada con lo que podríamos llamar la especie química, aunque no de forma estricta. Así pues, podremos tener intersticiales, vacantes, boro, boro intersticial, carbono, etc. No está sólo relacionado con el número atómico, sino también con cierto tipo especial de configuración, como boro sustitucional, boro intersticial negativo o vacante doblemente positiva.

▷ 8.1.3

Tratamiento del espacio en DADOS

DADOS, a la hora de manejar el espacio, supone que la celda de simulación es un paralelepípedo. Los ejes x , y y z del paralelepípedo se pasan al principio de la simulación y, una vez que el programa tiene las dimensiones del espacio a simular, procede a dividir éste en pequeñas cajas, de las que creará, como mucho, 2^{maxbits} (ver figura 8.2). Para ello intenta dividir los ejes (siempre en números potencia de dos, por razones de eficiencia) proporcionalmente a su tamaño, para que así los ejes divididos en $2^m_{x,y,z}$ partes den un total de $m_x + m_y + m_z = \text{maxbits}$ cajas.

DADOS asume también que el eje x está orientado hacia el fondo de la oblea, dando la profundidad de ésta, de tal forma que la superficie de la oblea se encuentra en $x = 0$ y su área resulta ser yz .

Para simular formas irregulares, DADOS asigna distintos tipos de material a cada una de las pequeñas cajas si bien, si estas cajas no fueran suficientemente pequeñas, esto podría dar lugar a una pérdida de precisión al transformar una forma continua e irregular en una sucesión de rectángulos. En el programa DADOS incluido en el Simulador Atomístico de la Universidad de Valladolid (UVAS-DADOS) se supone que todas las cajas cuyo centro esté a menos de 1.5 nm de la superficie son SiO_2 , tomándose en cualquier caso como mínimo una fila de cajas de este material. El resto pasa a ser Si cristalino.

Por último, UVAS-DADOS va recopilando información para la proyección del eje x según avanza la simulación, (perfiles de concentración en profundidad), dividiendo para ello el eje x en N_{slice} lonchas (ver tabla 8.1). Si se desea incrementar la resolución de esta información siempre es posible incrementar este número, si bien la memoria empleada depende también mucho tanto del parámetro N_{slice}

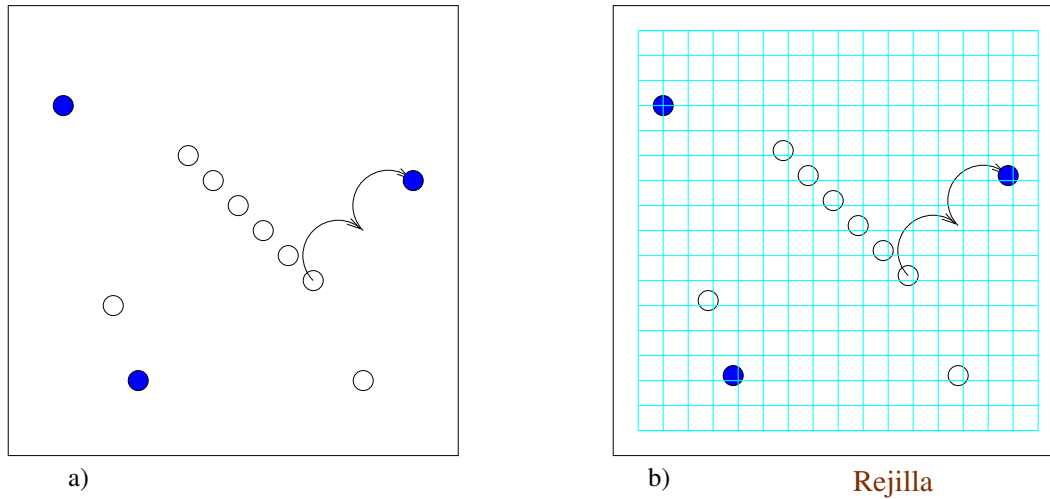


Figura 8.2: DADOS divide el espacio en pequeñas cajas, dividiendo cada eje por un número que es potencia de 2.

| Símbolo | Nombre ddp | Valor |
|--------------------|------------|----------|
| maxbits | BitsBoxes | 18 cajas |
| N_{slice} | HistoBits | 6 tiras |

Tabla 8.1: Opciones globales de DADOS para la regulación del espacio.

como de maxbits.

▷ 8.1.4

Distintos materiales en DADOS

Los siguientes tipos de materiales se encuentran definidos en DADOS:

Ambiente. No se trata estrictamente de un material, pero es la forma de referirse a la ausencia de éste.

Silicio. Silicio cristalino o amorfo.

Óxido de silicio. Material tipo SiO_2 .

Nitruros de silicio.

DADOS implementa modelos de difusión, formación de defectos, acumulación de daño y carga para el material silicio, y un modelo de difusión simple para los otros tipos de material.

También, como ya se ha comentado, debido a la forma en que se trabaja internamente, (con cajas rectangulares), y a que cada una de estas cajas sólo

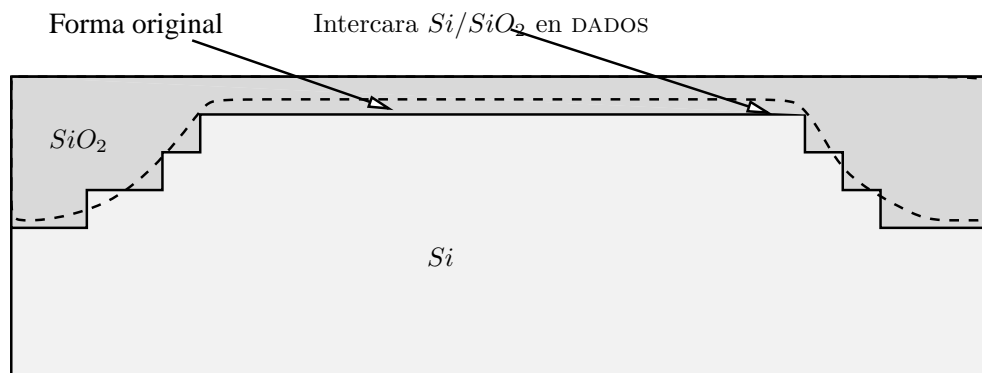


Figura 8.3: Al tomarse el tipo de material en pequeñas cajas rectangulares individuales, las intercaras entre los materiales pueden presentar una estructura escalonada.

puede contener un único tipo de material, las intercaras (de las cuales hablaremos en la sección 8.7) pueden presentar un aspecto cuadrículado o escalonado que, en la mayoría de los casos, no debería representar ningún problema especial. (Ver figura 8.3).

▷ 8.1.5

Gestión del tiempo en DADOS

La forma de medir el paso del tiempo es una tarea bastante compleja en el simulador DADOS, debido a la posible presencia de multitud de partículas saltando con distintas frecuencias, intercaras, superficies y defectos extendidos emitiendo partículas de vez en cuando, así como otros fenómenos, como eventos de recristalización, también con sus tiempos característicos, todo ello funcionando a la vez. Para gestionar esta ingente cantidad de procesos paralelos DADOS organiza éstos en dos categorías especiales, a las que se ha denominado *rates* y *timers*.

Los *timers* se usan para los eventos de los cuales se conoce con antelación cuando van a suceder, en principio independientemente de que sucedan otros fenómenos, y por tanto se recuerdan de forma parecida a como se programaría la alarma de un reloj. Por ejemplo, se utilizan timers para las emisiones desde la superficie y los eventos de recristalización.

En cuanto a las *rates* estas están más relacionadas con todos los saltos y emisiones de partículas por parte de defectos. Se podría razonar que la emisión de un intersticial por parte de un cluster, por ejemplo, también puede ser programado como un timer, razonamiento que es cierto, pero en principio cada vez que el cluster atrapa un nuevo I habría que cambiar dicho timer, lo cual no sería eficiente, siendo mejor esta segunda aproximación. Las rates están relacionadas directamente con las frecuencias de los eventos de migración, emisión, etc. y son representadas internamente por medio de un número entero, lo que afecta a su

| Nombre | Valor |
|----------------------------------|----------------|
| Δt_{\min} | 1e-13 s |
| $N_{\text{Particles}_{\max}}$ | 1e+10 |
| ν_{\max} | 1e+13 Hz |
| $K_{\nu \rightarrow \text{int}}$ | 3.40282e+15 |
| ν_{\min} | 2.93874e-16 Hz |

Tabla 8.2: Valores internos de DADOS para las frecuencias mínimas y máximas.

precisión, concretamente a los valores máximos y mínimos permitidos para la frecuencia.

En DADOS, el periodo más corto con el que algo puede suceder es denominado Δt_{\min} , siendo un valor fijo que el usuario no puede, en principio, cambiar, pero que está programado para ser más que suficiente para representar cualquier fenómeno físico. En consecuencia, la frecuencia más alta posible será

$$\nu_{\max} = \frac{1}{\Delta t_{\min}}.$$

A su vez se permite que un número máximo de, como mucho, $N_{\text{Particles}_{\max}}$ partículas tengan esta frecuencia máxima (un número mayor podría dar problemas de redondeo). Estas cantidades se encuentran expuestas en la tabla 8.2. También, puesto que DADOS almacena las frecuencias como enteros, concretamente de 128 bits, es posible mantener un factor de conversión entre frecuencias y enteros $K_{\nu \rightarrow \text{int}}$ gracias la siguiente ecuación:

$$2^{128} = \underbrace{\nu_{\max} \times K_{\nu \rightarrow \text{int}}}_{\text{Máxima frecuencia interna}} N_{\text{Particles}_{\max}}.$$

Con todo lo anterior se puede ver que la frecuencia mínima posible será 1 en esta representación interna, luego

$$\nu_{\min} = \frac{1}{K_{\nu \rightarrow \text{int}}},$$

valor que también está expuesto en la tabla 8.2.

Todo lo anterior hace que un componente central para un simulador kMC como DADOS sea un algoritmo que vaya seleccionando secuencialmente un evento al azar de acuerdo con su frecuencia respectiva. La figura 8.4 ilustrar un posible mecanismo de selección para la configuración mostrada en la figura 8.1, que consistía en 3 vacantes (V), 2 intersticiales (I) y 1 defecto $\{311\}$. Podemos suponer, a modo de ejemplo, que las frecuencias de salto para el I y la V sean 1000 s^{-1} y 100 s^{-1} respectivamente, y que la frecuencia de emisión para el $\{311\}$ sea 10 s^{-1} . Con esto, si quisiéramos simular 1 s necesitaríamos simular un total de 3210 eventos diferentes, de los cuales es fácil ver que la probabilidad de tomar el de una V

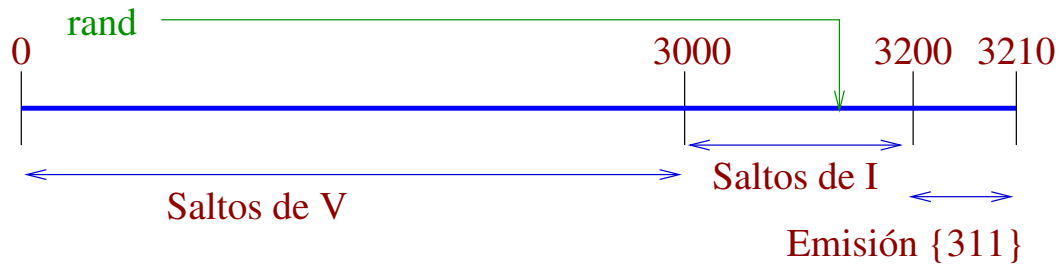


Figura 8.4: Un algoritmo que selecciona un evento de manera aleatoria, pero de acuerdo a la frecuencia con que cada uno de ellos debe suceder, dependiendo de su configuración atómica instantánea.

debería ser $3000/3210$, para un I $200/3210$ y por último para el $\{311\}$ sería de $10/3210$. Con esto podríamos escoger un número aleatorio entre 0 y 3210 y ver qué evento es elegido entre todos. En el caso concreto de la figura 8.4, se habría seleccionado el salto de un I , y aún tocaría escoger a cuál de los dos I s aplicar el salto mediante el uso de otro número aleatorio.

▷ 8.2

Defectos puntuales I y V

Los distintos eventos que un defecto puntual puede realizar están representados en la figura 8.5. En principio, éstos pueden saltar (proceso de migración), y al moverse pueden ser capturados por otras partículas (bien otros defectos puntuales o defectos extendidos).

Teoría en la
sec. 7.4,
pág. 83.

▷ 8.2.1

Migración

La difusión de intersticiales y vacantes está producido por un proceso de migración. DADOS simula la migración de los defectos puntuales cambiando sus coordenadas una distancia λ fija en uno de los tres ejes con la frecuencia apropiada para cada defecto. Esta distancia se corresponde con la distancia de segundos vecinos, y la frecuencia se calcula de acuerdo al prefactor y energía de difusividad:

$$D = D_{m,0} \times \exp\left(-\frac{E_m}{k_B T}\right), \quad (8.1)$$

donde D_m es el prefactor de migración y E_m su energía. Estos valores se pasan como parámetros al programa DADOS, encontrándose los valores utilizados

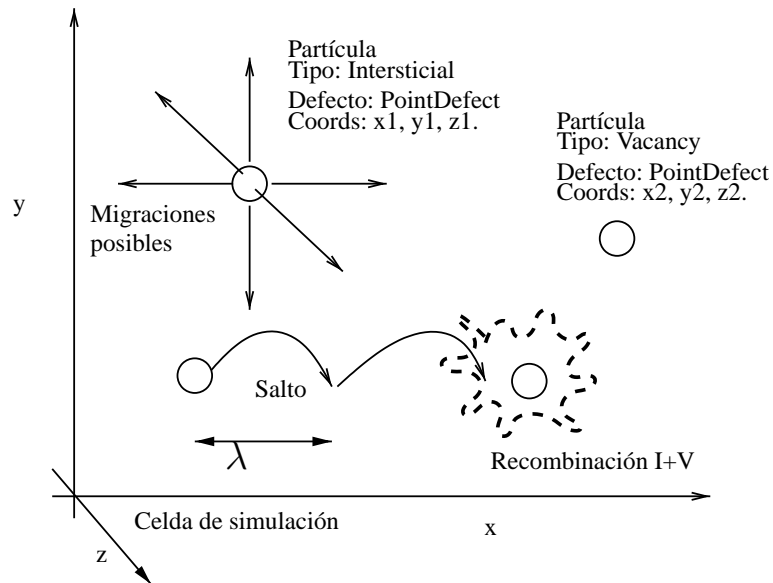


Figura 8.5: Un defecto puntual tiene tres coordenadas, junto con dos números para especificar el tipo de *partícula* y *defecto*. Se mueve una distancia λ en direcciones ortogonales y puede interactuar con otras partículas siempre que se encuentre en su radio de captura.

| Símbolo | Nombre ddp | Valor |
|--------------|------------|--|
| $D_{m,0}(I)$ | Dm_I | $0.005 \text{ cm}^{-2} \text{ s}^{-1}$ |
| $E_m(I)$ | Em_I | 1 eV |
| $D_{m,0}(V)$ | Dm_V | $0.001 \text{ cm}^{-2} \text{ s}^{-1}$ |
| $E_m(V)$ | Em_V | 0.4 eV |

Tabla 8.3: Parámetros para los intersticiales y vacantes, tomados del fichero de parámetros de DADOS.

actualmente en la tabla 8.3.

Como ya se ha comentado, DADOS modifica las coordenadas de las partículas únicamente en alguno de los tres ejes ortogonales, permitiéndose por tanto un total de 6 saltos diferentes ($\pm x, \pm y, \pm z$). Esta aproximación permite al código ser bastante rápido, sin eliminar nada esencial del proceso de difusión.

► 8.2.2

Interacciones

Siempre que una partícula se encuentre en las cercanías de otra (y dentro de su radio de captura) es posible una interacción entre ambas. Para que esta interacción se permita DADOS establece los siguientes criterios:

- En primer lugar la interacción tiene que estar permitida en el código mismo

| | I | V | As | B | C | F | In | O | P | - | $AsNitride$ | $BNitride$ | $PNitride$ | - | $AsOxide$ | $BOxide$ | $POxide$ | - | As^0 | As_i | As_i^+ | As_V | As_V^- | As_V^+ | B^0 | |
|-------------|-----|-----|------|-----|-----|-----|------|-----|-----|---|-------------|------------|------------|---|-----------|----------|----------|---|--------|--------|----------|--------|----------|----------|-------|--|
| I | s | | | | | | | | | | | | | | | | | | | | | | | | | |
| V | s | s | | | | | | | | | | | | | | | | | | | | | | | | |
| As | s | s | s | | | | | | | | | | | | | | | | | | | | | | | |
| B | s | n | n | n | | | | | | | | | | | | | | | | | | | | | | |
| C | s | n | n | n | n | | | | | | | | | | | | | | | | | | | | | |
| F | n | s | n | n | n | s | | | | | | | | | | | | | | | | | | | | |
| In | s | n | n | n | n | n | n | | | | | | | | | | | | | | | | | | | |
| O | s | s | n | n | n | n | n | s | | | | | | | | | | | | | | | | | | |
| P | s | s | n | n | n | n | n | n | s | | | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | |
| $AsNitride$ | n | n | n | n | n | n | n | n | n | | n | | | | | | | | | | | | | | | |
| $BNitride$ | n | n | n | n | n | n | n | n | n | | n | n | | | | | | | | | | | | | | |
| $PNitride$ | n | n | n | n | n | n | n | n | n | | n | n | n | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | |
| $AsOxide$ | n | n | n | n | n | n | n | n | n | | n | n | n | | n | | | | | | | | | | | |
| $BOxide$ | n | n | n | n | n | n | n | n | n | | n | n | n | | n | n | | | | | | | | | | |
| $POxide$ | n | n | n | n | n | n | n | n | n | | n | n | n | | n | n | n | | | | | | | | | |
| - | | | | | | | | | | | | | | | | | | | | | | | | | | |
| As^0 | s | s | n | n | n | n | n | n | n | | n | n | n | | n | n | n | | n | | | | | | | |
| As_i | n | s | s | n | n | n | n | n | n | | n | n | n | | n | n | n | | n | n | | | | | | |
| As_i^+ | n | s | n | n | n | n | n | n | n | | n | n | n | | n | n | n | | n | n | n | | | | | |
| As_V | s | s | n | n | n | n | n | n | n | | n | n | n | | n | n | n | | n | s | n | s | | | | |
| As_V^- | s | n | s | n | n | n | n | n | n | | n | n | n | | n | n | n | | n | n | n | n | | | | |
| As_V^+ | s | n | n | n | n | n | n | n | n | | n | n | n | | n | n | n | | n | n | n | n | n | | | |
| B^0 | s | n | n | n | n | n | n | n | n | | n | n | n | | n | n | n | | n | n | n | n | n | n | | |

Tabla 8.4: Interacciones permitidas y prohibidas. Se listan todas las especies y cargas. (i)

(programada). Por ejemplo, la interacción boro sustitucional (B) más un intersticial (I) está permitida, pero no así B más vacante.

- En segundo lugar hay que tener en cuenta los aspectos energéticos de la interacción.
- Por último, hay que tener en cuenta los estados de carga, que prohíben las interacciones entre partículas con la misma carga $(+, +)$ o $(-, -)$.

Las interacciones entre partículas (primer aspecto), como hemos adelantado ya, se encuentran fijadas en el código, no son un parámetro. Las tablas 8.4, 8.5 y 8.6 muestran cuáles son las interacciones permitidas y las que no.

Una vez que el primer aspecto de la interacción se cumple, es decir, que el código permite interaccionar las partículas, hay que tener en cuenta los aspectos energéticos:

- En principio DADOS siempre va a permitir energéticamente la interacción entre dos defectos puntuales (partículas solas). Ahora bien, en los casos en

| | <i>I</i> | <i>V</i> | <i>As</i> | <i>B</i> | <i>C</i> | <i>F</i> | <i>In</i> | <i>O</i> | <i>P</i> | AsNitride | | | AsOxide | | | <i>As</i> ⁰ | <i>As</i> _{<i>i</i>} | <i>As</i> _{<i>i</i>} ⁺ | <i>As</i> _{<i>V</i>} | <i>As</i> _{<i>V</i>} ⁻ | <i>As</i> _{<i>V</i>} ⁺ | <i>B</i> ⁰ |
|---|----------|----------|-----------|----------|----------|----------|-----------|----------|----------|-----------------|-----------------|-----------------|---------------|---------------|---|------------------------|-------------------------------|--|-------------------------------|--|--|-----------------------|
| | | | | | | | | | | <i>BNitride</i> | <i>PNitride</i> | <i>BNitride</i> | <i>BOxide</i> | <i>POxide</i> | | | | | | | | |
| <i>B</i> ⁰ | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>B</i> _{<i>i</i>} | s | s | n | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>B</i> _{<i>i</i>} ⁻ | n | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>B</i> _{<i>i</i>} ⁺ | n | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>B</i> ₂ <i>I</i> ₂ | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>BI</i> ₂ | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>C</i> _{<i>i</i>} | s | s | n | n | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>I</i> ₂ | s | s | n | s | s | n | n | s | n | n | n | n | n | n | n | n | n | s | s | s | n | |
| <i>I</i> ⁻ | s | s | s | n | n | n | n | s | s | n | n | n | n | n | n | n | n | s | n | s | n | |
| <i>I</i> ⁺ | s | s | n | s | n | n | s | s | n | n | n | n | n | n | n | n | n | s | s | n | n | |
| <i>In</i> _{<i>i</i>} | s | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>In</i> _{<i>i</i>} ⁻ | s | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>In</i> _{<i>V</i>} | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>In</i> _{<i>V</i>} ⁻ | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>O</i> _{<i>V</i>} | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>P</i> ⁰ | s | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>P</i> _{<i>i</i>} | s | s | n | n | n | n | n | n | s | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>P</i> _{<i>i</i>} ⁺ | s | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>P</i> _{<i>V</i>} | s | s | n | n | n | n | n | n | s | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>P</i> _{<i>V</i>} ⁻ | s | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>P</i> _{<i>V</i>} ⁺ | s | s | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | |
| <i>V</i> ⁻ | s | s | s | n | n | s | n | n | s | n | n | n | n | n | n | s | s | n | n | n | n | |
| <i>V</i> ⁻⁻ | s | s | s | n | n | s | n | n | s | n | n | n | n | n | n | s | s | n | n | n | n | |
| <i>V</i> ⁺ | s | s | n | n | n | s | n | n | n | n | n | n | n | n | n | s | n | n | n | n | n | |
| <i>V</i> ⁺⁺ | s | s | n | n | n | s | n | n | n | n | n | n | n | n | n | s | n | n | n | n | n | |

Tabla 8.5: Interacciones permitidas y prohibidas. (ii)

| | B^0 | B_i | B_i^- | B_i^+ | B_2I_2 | BI_2 | C_i | I_2 | I^- | I^+ | In_i | In_i^- | In_V | In_V^- | O_V | P^0 | P_i | P_i^+ | P_V | P_V^- | P_V^+ | V^- | V^{--} | V^+ | V^{++} | |
|----------|-------|-------|---------|---------|----------|--------|-------|-------|-------|-------|--------|----------|--------|----------|-------|-------|-------|---------|-------|---------|---------|-------|----------|-------|----------|--|
| B^0 | n | | | | | | | | | | | | | | | | | | | | | | | | | |
| B_i | n | s | | | | | | | | | | | | | | | | | | | | | | | | |
| B_i^- | n | n | n | | | | | | | | | | | | | | | | | | | | | | | |
| B_i^+ | n | n | n | n | | | | | | | | | | | | | | | | | | | | | | |
| B_2I_2 | n | n | n | n | n | | | | | | | | | | | | | | | | | | | | | |
| BI_2 | n | n | n | n | n | n | | | | | | | | | | | | | | | | | | | | |
| C_i | n | n | n | n | n | n | s | | | | | | | | | | | | | | | | | | | |
| I_2 | n | s | s | s | s | s | s | s | | | | | | | | | | | | | | | | | | |
| I^- | n | n | n | n | n | n | n | s | n | | | | | | | | | | | | | | | | | |
| I^+ | n | n | n | n | n | n | n | s | s | n | | | | | | | | | | | | | | | | |
| In_i | n | n | n | n | n | n | n | n | s | s | n | | | | | | | | | | | | | | | |
| In_i^- | n | n | n | n | n | n | n | n | n | s | n | n | | | | | | | | | | | | | | |
| In_V | n | n | n | n | n | n | n | s | n | n | n | n | n | | | | | | | | | | | | | |
| In_V^- | n | n | n | n | n | n | n | s | n | n | n | n | n | n | | | | | | | | | | | | |
| O_V | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | | | | | | | | | | | |
| P^0 | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | n | | | | | | | | | | |
| P_i | n | n | n | n | n | n | n | s | s | s | n | n | n | n | n | n | n | | | | | | | | | |
| P_i^+ | n | n | n | n | n | n | n | s | s | n | n | n | n | n | n | n | n | n | | | | | | | | |
| P_V | n | n | n | n | n | n | n | s | s | s | n | n | n | n | n | n | s | n | s | | | | | | | |
| P_V^- | n | n | n | n | n | n | n | s | n | s | n | n | n | n | n | n | n | n | n | n | | | | | | |
| P_V^+ | n | n | n | n | n | n | n | s | s | n | n | n | n | n | n | n | n | n | n | n | n | | | | | |
| V^- | n | s | n | s | n | n | n | s | n | s | s | n | n | n | n | n | s | s | s | n | s | n | | | | |
| V^{--} | n | s | n | s | n | n | n | s | n | s | s | n | n | n | n | n | s | s | s | n | s | n | n | | | |
| V^+ | n | s | s | n | n | n | n | s | s | n | s | s | n | n | n | n | s | n | s | s | n | s | s | n | | |
| V^{++} | n | s | s | n | n | n | n | s | s | n | s | s | n | n | n | n | s | n | s | s | n | s | s | n | n | |

Tabla 8.6: Interacciones permitidas y prohibidas. (iii)

los que dicha reacción sea poco favorable, es muy probable que en poco tiempo un nuevo evento disocie las partículas volviendo a emitir las partículas originales (ver sección 8.6.1).

- Cuando un defecto puntual interacciona con un defecto extenso, éste puede rechazar la interacción según cuál sea la energía de formación del defecto con la nueva partícula¹.

Cuando la partícula salta cerca de la superficie o una intercara, es posible también que se interaccione con ella, tal y como se explica en la sección 8.7.

Así pues, las partículas móviles (como I s, V s y dopantes móviles, explicados en la sección 8.6.1) son extraordinariamente importantes puesto que a través de ellas se establecen todas las interacciones. Los defectos extendidos, al no moverse, tienen un papel más pasivo, puesto que su crecimiento y reacciones se producen cuando les llega una de estas partículas.

▷ 8.3

Defectos extendidos de I s y V s

Teoría en la
sec. 7.5,
pág. 86.

Un defecto extendido es aquél que abarca más de una partícula. Los defectos extendidos no migran, pero interaccionan con las partículas móviles que llegan hasta ellos, y también pueden emitir partículas. Las partículas que pertenecen a un determinado defecto extendido presentan en su etiqueta de tipo de defecto el tipo de defecto en el que se encuentran.

Como los defectos extendidos se construyen a partir de aglomerados de distintas partículas es posible situar éstas de tal manera que representen el defecto de forma precisa, lo cual también influye en la variación del volumen de captura del defecto extenso con su tamaño que, de esta manera, surge de manera natural —en una primera aproximación— por la propia representación de éste. En general, el volumen de captura para los defectos extendidos de DADOS es la superposición de las regiones de captura de cada una de las partículas que constituyen el defecto, aproximación que, si bien no es totalmente precisa, sí que sigue bastante de cerca el crecimiento y disminución del defecto extendido y la consiguiente evolución de la sección de captura.

▷ 8.3.1

Pequeños clusters de I y V

Teoría en la
sec. 7.5.4,
pág. 90.

En DADOS hay un tipo de cluster pequeño para I s y otro para V s. Estos

¹Para más detalles se puede consultar el defecto concreto.

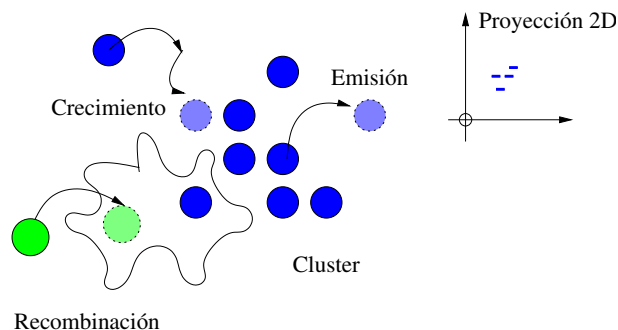


Figura 8.6: Los clusters son conglomerados desordenados de partículas I o V . Pueden crecer atrapando partículas, o bien disolverse por emisión o recombinación.

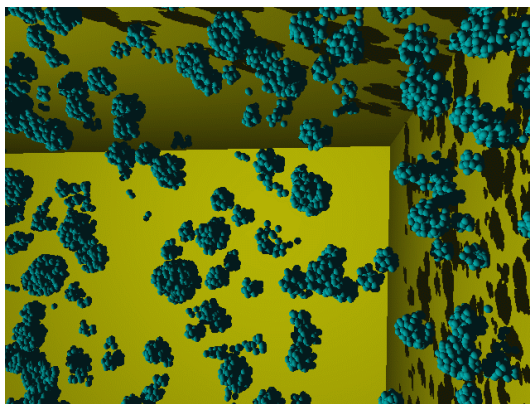


Figura 8.7: Pequeños clusters de intersticiales y vacantes simulados con DADOS.

pequeños clusters tienen modelizada la forma y los procesos de captura, emisión y recombinación representados en la figura 8.6. Todas las partículas que pertenecen a este tipo de cluster (que sólo pueden ser I s para los clusters de I , o V s para los de V) presentan la etiqueta “Cluster”.

Dado que los clusters de I s y V s están modelizados de la misma manera, y únicamente cambia el tipo de partícula y sus parámetros respectivos, nos vamos a centrar en el cluster de I para las siguientes explicaciones.

Teoría en la
sec. 7.5.1,
pág. 86.

Forma

Los pequeños clusters de I s o V s no presentan ninguna forma especial o, lo que es lo mismo, DADOS no los reordena, sino que se van quedando con la forma de las partículas que van llegando, que se limitan a adherirse al cluster formado.

Esto supone que la forma de los pequeños clusters va a ser irregular e impredecible, puesto que depende de cómo le vayan llegando las partículas en su movimiento aleatorio, como se representa en la figura 8.7.

Por último, cuando un cluster crece lo suficiente, DADOS lo transforma en un $\{311\}$ (como se expone en la sección 8.3.2) cuando es de I s y en un void (ver 8.3.4) cuando es de V s. Los parámetros de transición de unos a otros son $\text{Cluster} \rightarrow \{311\}$ (tabla 8.8) y $\text{Cluster} \rightarrow \text{Void}$ (tabla 8.10) para I s y V s respectivamente.

Captura

Cuando algún I o V se sitúa a una distancia menor que el radio de captura de cualquiera de las partículas integrantes de un cluster, éste puede interaccionar con ella. Si la partícula tuviera más de un cluster o partícula con la que interaccionar (imaginemos que entra en una región muy poblada), se selecciona una interacción al azar entre las posibles.

Si un cluster de I interacciona con un I (o V con V) siempre se atrapa la partícula, se añade ésta a la lista de partículas que gestiona el cluster, se le informa a la nueva partícula de quien la gestiona ahora y, consecuentemente, el cluster crece. Un cluster atraparé una partícula siempre, incluso aunque su energía de ligadura sea muy pequeña. En este último caso la probabilidad de re-emisión sería bastante grande, y por tanto el cluster se desharía pronto de la recién llegada partícula.

Emisión

La siguiente ecuación ofrece la frecuencia de emisión que DADOS asigna a cada cluster de la simulación, dependiendo de su tamaño:

$$\nu_{\text{emission}} = \nu_{0,\text{emission}} \times \exp\left(-\frac{E_{\text{diss}}}{k_B T}\right). \quad (8.2)$$

el prefactor $\nu_{0,\text{emission}}$ también depende del tamaño del cluster (de manera lineal en el modelo actual) y $E_{\text{diss}} = E_b(I_n) + E_m(I)$ es la energía necesaria para romper un cluster de tamaño n y emitir una partícula, donde E_b y E_m son, respectivamente, las energías de enlace y migración de un I (o V según el caso, ver figura 8.8).

Los valores para las energías de enlace según el tamaño del cluster, $E_b(I_n)$, se especifican una por una como parámetros aunque también se pueden calcular con unos pocos parámetros extra a partir de la siguiente ecuación:

$$E_b(I_n) = E_{b,L} - (E_{b,L} - E_{b,S}) \frac{(n^a) - (n-1)^a}{2^a - 1}, \quad (8.3)$$

donde $E_{b,L}$ sería la energía de enlace para un cluster de tamaño infinito, $E_{b,S}$ para el cluster más pequeño posible y a es un exponente, que normalmente se sitúa en $a = 2/3$ o $a = 3/4$. La figura 8.9 muestra los valores concretos que hay en el fichero de parámetros del programa para los clusters de I , junto con los valores que saldrían tras la aplicación de la ecuación 8.3. La figura 8.10 hace lo mismo con las vacantes.

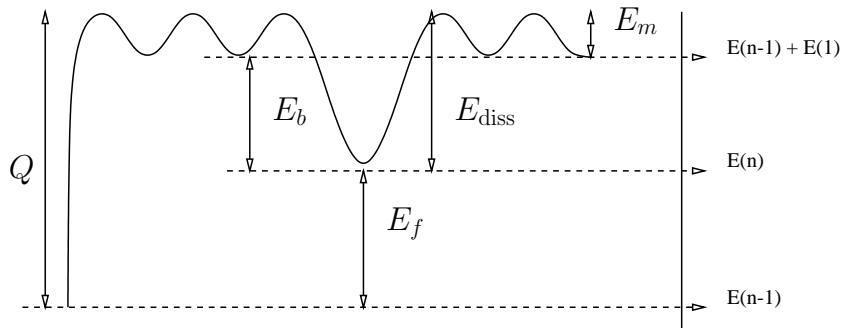


Figura 8.8: Energía necesaria para emitir una partícula desde un cluster de tamaño n .

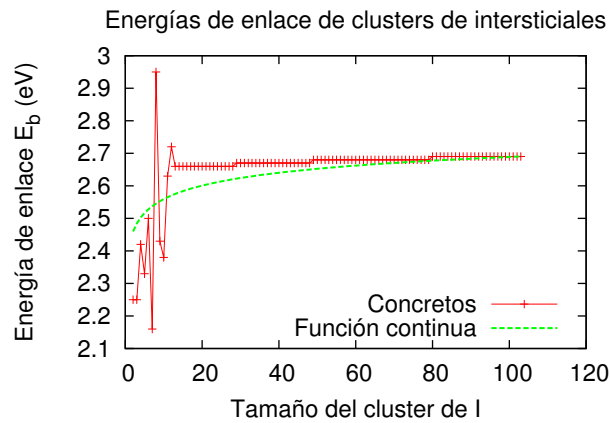


Figura 8.9: Valores concretos y continuos de las energías de enlace para clusters de intersticiales.

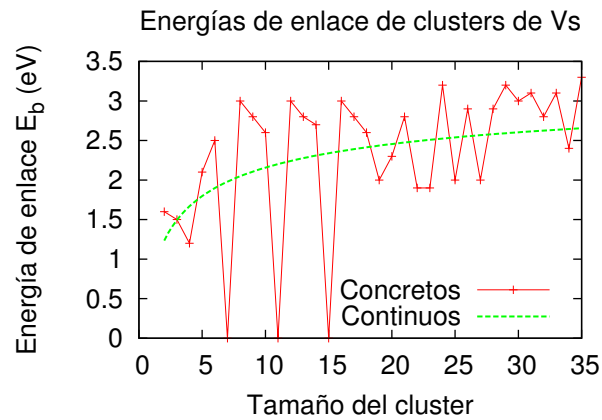


Figura 8.10: Valores concretos y continuos de las energías de enlace para clusters de vacantes.

| Símbolo | Nombre ddp | Valor |
|--------------|--------------------|------------|
| $E_{b,S}(I)$ | Eb_SmallestCluster | 4.42633 eV |
| $E_{b,L}(I)$ | Eb_LargeCluster | 4.42633 eV |
| $a(I)$ | exponent_Cluster | 0.9 |
| $E_{b,S}(V)$ | Eb_SmallestCluster | 4.42633 eV |
| $E_{b,L}(V)$ | Eb_LargeCluster | 4.42633 eV |
| $a(V)$ | exponent_Cluster | 0.6667 |

Tabla 8.7: Valores de los parámetros utilizados en los clusters de I s y V s.

Los parámetros relacionados con los modelos de clusters de I y V usados en DADOS están expuestos en la tabla 8.7.

Recombinación

Cada vez que una vacante interacciona con un cluster de intersticiales (o al revés) se produce un fenómeno de recombinación en el cual, simplemente, se aniquila el par involucrado y por tanto se disminuye el tamaño del cluster en una unidad. Por último, cuando sólo quedan dos partículas en un cluster y una de ellas es recombinada o emitida, el cluster se disuelve liberando la partícula que queda como defecto puntual. En principio no hay ninguna energía asociada a estos procesos de recombinación, siendo por tanto automáticos.

▷ 8.3.2

Defectos {311}

Teoría en la
sec. 7.5.2,
pág. 87.

Los defectos tipo {311} son cadenas de intersticiales que se extienden en un plano {311} hacia una dirección $\langle 110 \rangle$. DADOS modeliza los {311} de la forma que puede apreciarse en la figura 8.11. Los {311} pueden emitir y atrapar I s, así como recombinarse con V . Su forma es específicamente creada para corresponderse con la que puede verse experimentalmente por microscopía.

Forma

DADOS reordena los intersticiales que conforman un {311} como columnas de intersticiales, con los parámetros cristalográficos derivados por (Takeda, 1991). Internamente DADOS representa los {311} como una lista de partículas con dos finales, y va asignando a los nuevas partículas que van llegando posiciones de acuerdo a los parámetros cristalográficos antes citados (ver figura 8.12).

Los {311}s pueden crecer hasta tamaños significativamente grandes, y hay algunos trabajos (Liu, 1996) en los que se apunta a una pequeña variación de geometría del {311} con el tamaño (incremento de la anchura, o número de filas), que afectaría al radio de captura. Estas variaciones también están simuladas por

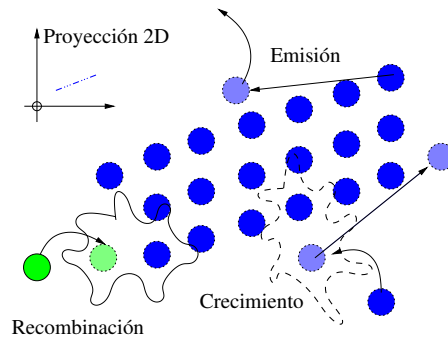


Figura 8.11: Los defectos de tipo $\{311\}$ son cadenas de intersticiales orientadas. Estos defectos pueden atrapar y emitir I así como recombinarse con vacantes.

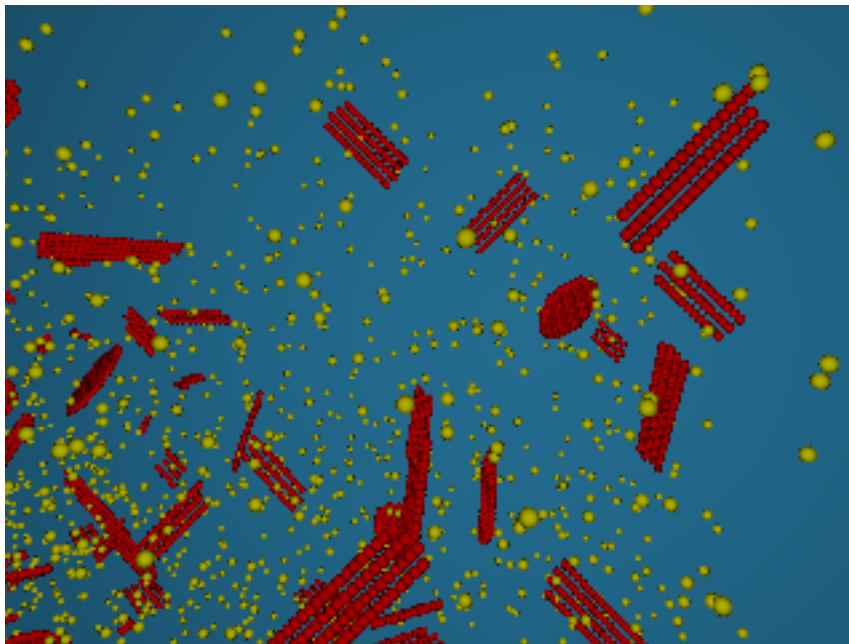


Figura 8.12: $\{311\}$ s extraídos de una simulación de DADOS. También hay presentes dos pequeños bucles de dislocación. Las esferas rojas son intersticiales de Si, las amarillas átomos de As.

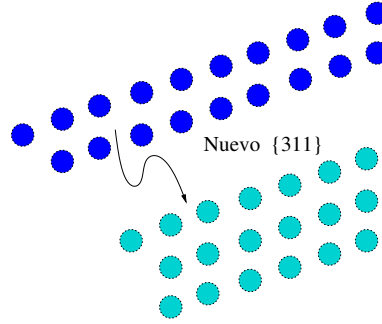


Figura 8.13: La anchura de un $\{311\}$ cambia con el tamaño.

DADOS, cambiando así la forma del $\{311\}$ cuando se alcanzan ciertos valores críticos, tal como se representa en el esquema 8.13. Concretamente la relación experimental entre la longitud de un $\{311\}$ y su anchura es

$$W \approx \sqrt{0,5\text{nm} \times L},$$

donde W y L son proporcionales al número de columnas y filas, respectivamente:

$$W = \frac{\sqrt{22}}{4} \times a \times (N_{\text{col}} - 1), \quad L = \frac{a}{\sqrt{2}} \times (N_{\text{row}} - 1).$$

Respecto al número de filas, N_{rows} , se escoge de forma que siga las expresiones anteriores. Para evitar cambios innecesarios (yendo adelante y atrás sucesivamente si se traspasan las fronteras entre tamaños al emitir y volver a coger una partícula) se establece una pequeña histéresis en este proceso.

Por último, cuando un $\{311\}$ crece lo suficiente, sufre una transformación en bucle de dislocación, de los cuales se habla en la sección 8.3.3. El valor de la transición (N_{loop}) se calcula de acuerdo a la siguiente fórmula

$$N_{\text{loop}} = N_{0,\text{loop}} \times \exp\left(\frac{E_{\{311\} \rightarrow \text{loop}}}{k_B T}\right),$$

donde los parámetros involucrados en la fórmula se encuentran en la tabla 8.8. Cuando este valor vale menos que el tamaño para el cual la energía de formación de los $\{311\}$ s y los bucles de dislocación coinciden, se aumenta hasta dicho valor.

Captura

Los $\{311\}$ s capturan todos los intersticiales que se encuentren dentro de su radio de captura, y los incorporan al defecto y por tanto a su lista de átomos gestionados. De esta manera, cuando un I en su movimiento colisiona con un $\{311\}$ éste toma la partícula y la incorpora a uno de los dos finales del $\{311\}$,

| Símbolo | Nombre ddp | Valor |
|--------------------------------|------------------|------------|
| Cluster $\rightarrow \{311\}$ | MIN311SIZE | 33 átomos |
| $N_{0,loop}$ | size0_311toDLoop | 1.6 átomos |
| $E_{\{311\} \rightarrow loop}$ | E311toDLoop | 0.6 eV |

Tabla 8.8: Valores de parámetros para los defectos $\{311\}$.

aquél que esté más cerca de donde estuviera la partícula original. Cuando esto no es posible (como por ejemplo si un extremo está tocando el borde de la caja de simulación) se añade al otro borde y, si esto tampoco fuera posible (un caso más que improbable) se rechaza la interacción.

Emisión

La frecuencia de emisión de intersticiales desde un $\{311\}$ se calcula exactamente igual que para los clusters de I , cuya descripción se encuentra en 8.3.1. Para preservar la reversibilidad microscópica se toma una partícula al azar de uno de los dos finales, pero se emite desde cualquier punto posible de la superficie del $\{311\}$.

Recombinación

Los $\{311\}$ s recombinan vacantes. Cuando una V , durante su migración, pasa dentro del radio de captura de un $\{311\}$, se aniquila el I del final del $\{311\}$ que más cerca esté de la vacante original. Una vez que, bien por aniquilación o bien por emisión, el $\{311\}$ pasa el límite inferior de tamaño, es convertido en cluster de I s, si bien no se modifica la forma en esta transformación. Esto podría afectar ligeramente al radio de captura, pero dado que estamos hablando de tamaños pequeños no sería un cambio drástico y, por otra parte, no se sabe si hay un cambio de forma o no durante la disolución de los $\{311\}$.

▷ 8.3.3

Bucles de dislocación

Los bucles de dislocación son defectos bidimensionales que reposan en un plano $\{111\}$. Se dividen a su vez en imperfectos o “faulted” (FDL) y perfectos (PDL), si bien estos últimos aún no han sido implementados en DADOS. Los FDL presentan el aspecto de discos de intersticiales, tal y como puede verse en la figura 8.15.

Los diferentes procesos que puede realizar una dislocación se limitan al crecimiento por captura de nuevos I s y a su disolución, por emisión de éstos o recombinación con V s (ver figura 8.14).

Teoría en la
sec. 7.5.3,
pág. 88.

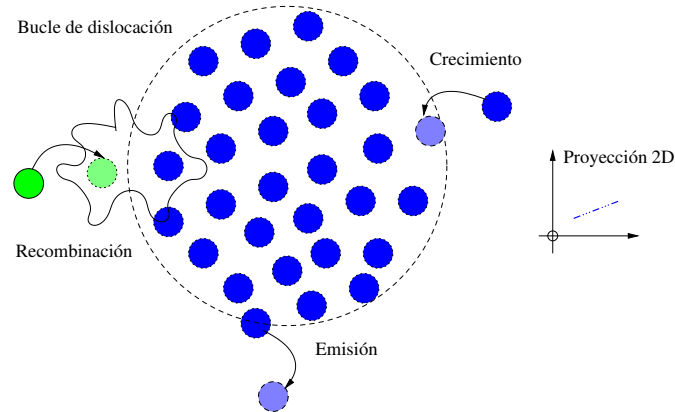


Figura 8.14: Los bucles de dislocación son discos planos de intersticiales. Crecen atrapando nuevos I_s y se disuelven a través de la emisión de dichos I_s o recombiándose con V_s .

Forma

La forma de un bucle de dislocación se calcula para formar un círculo en una orientación $\{111\}$ (o equivalente, es decir, $\{111\}$, $\{-1,1,1\}$, $\{1,-1,1\}$ y $\{1,1,-1\}$). Dicha orientación equivalente es escogida al azar, viéndose algunos ejemplos en la figura 8.15. Una vez que se conocen todas las posiciones posibles para partículas en un disco con la orientación adecuada, se almacenan éstas en un array, el cual es indexado siguiendo una espiral dentro del círculo, es decir, el primer elemento sería el centro y a continuación se van poniendo los átomos según lo alejados que estén de éste. Posteriormente, el crecimiento o decrecimiento de la dislocación será fácilmente manejable al limitarse a añadir o borrar un átomo del final de la lista.

Captura

Las dislocaciones capturan todos los I_s que se encuentran dentro de su radio de captura, y lo incorporan a su estructura añadiéndolos al final de su lista de átomos constituyentes. Una vez capturado se sitúa al átomo en la posición adecuada (según el índice que ocupe en la lista, como se ha explicado anteriormente), y se pone la etiqueta de “DLoop” en el átomo capturado para indicar que ahora pertenece a un bucle de dislocación.

Emisión

La frecuencia de emisión de intersticiales desde una dislocación responde a la siguiente ecuación:

$$\nu_{\text{emission}} = \nu_{0,\text{emission}} \times \exp\left(-\frac{E_{b,\text{loop}}(n) + E_m(I)}{k_B T}\right), \quad (8.4)$$

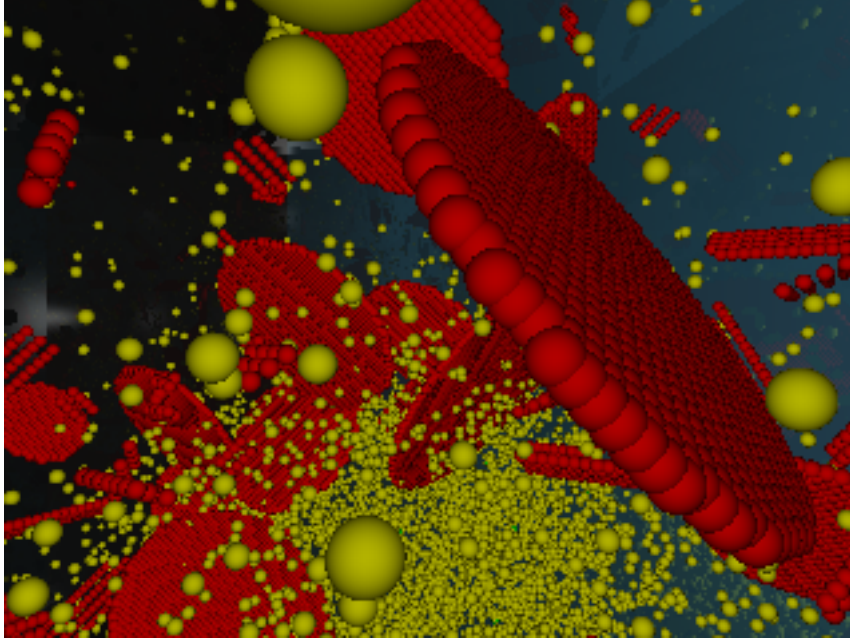


Figura 8.15: Bucles de dislocación en una simulación de DADOS, junto con algunos $\{311\}$ que también pueden verse en la imagen. Las esferas rojas son I y las amarillas As .

donde $\nu_{0,\text{emission}}$ es el prefactor, que depende linealmente con el tamaño del bucle de dislocación, y $E_{b,\text{loop}}$ es la energía de enlace de una dislocación de tamaño n . A su vez el valor de $E_{b,\text{loop}}(n)$ se calcula como

$$E_{b,\text{loop}}(n) = E_f(I) + E_f^{\text{loop}}(n-1) - E_f^{\text{loop}}(n),$$

y para calcular la energía de formación se sigue la referencia (Cristiano et al., 2000):

$$E_f^{\text{loop}}(n) = \pi R_{nm}^2 \gamma + \frac{a^2 \mu}{6(1-\nu)} \times R_{nm} \times \log\left(\frac{8R_{nm}}{b}\right) - nE_f(I),$$

donde $R_{nm} = \sqrt{n/(\pi d_{111})}$. γ , μ , ν y b son parámetros (expuestos en la tabla 8.9), a es el parámetro de red del silicio y d_{111} la densidad atómica en un plano $\{111\}$.

Algunos valores de energía, tomados de la referencia (Colombeau et al., 2001), se encuentran representados en la figura 7.5, y los valores calculados por DADOS para E_b^{loop} a partir de las ecuaciones anteriores y los parámetros de la tabla 8.9, están mostrados en la figura 8.16.

Recombinación

Los bucles de dislocación recombinan V_s destruyendo el último I que fue atrapado, y disminuyendo el tamaño de la dislocación en uno, como es natural,

| Símbolo | Nombre ddp | Valor |
|----------|-------------|--------|
| γ | gamma | 0.4375 |
| μ | mu | 472 |
| ν | nu | 0.3 |
| b | burgVectMod | 0.3135 |

Tabla 8.9: Parámetros para los bucles de dislocación. (Cristiano et al., 2000).

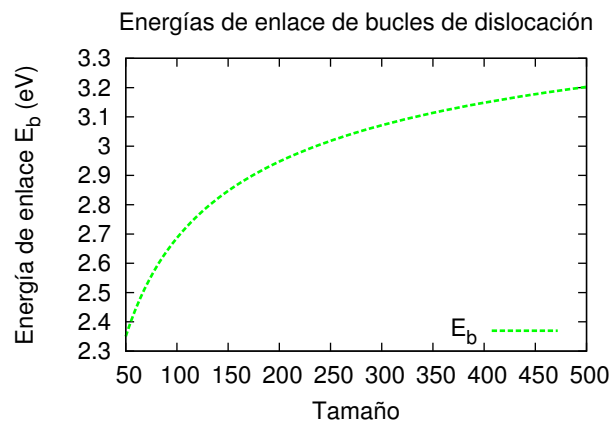


Figura 8.16: Energías de enlace de I a una dislocación, en función de su tamaño.

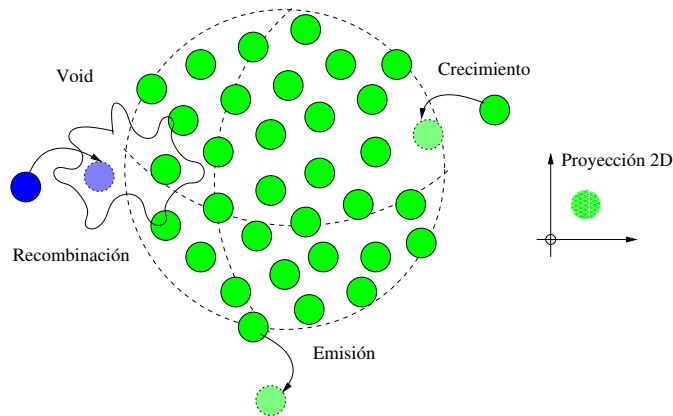


Figura 8.17: Los voids son aglomeraciones con forma esférica de vacantes. Pueden atrapar y emitir V_s , y recombinarse con I_s .

hasta que quede completamente disuelto (cosa que supone tiempos de recocido muy largos).

▷ 8.3.4

Void

Llamamos void a una acumulación de vacantes con forma esférica que crea un hueco vacío dentro del silicio. Las reacciones permitidas para los voids, especificadas en la figura 8.17, consisten en su crecimiento, mediante la captura de nuevas vacantes, y su disminución, mediante la emisión de vacantes y recombinación de las vacantes con I_s .

Teoría en la
sec. 7.5.5,
pág. 90.

Forma

Cuando los pequeños clusters de vacantes superan un cierto tamaño, DADOS los convierte en voids, para lo cual los cambia de forma. Esto se requiere para mantener la adecuada proporción de volumen/superficie a medida que el cluster va creciendo y, para tamaño grandes, se adopta una forma esférica, ocupando el mismo volumen que tendrían n átomos de red. Este proceso se implementa de la siguiente forma: al principio de la simulación se generan m posiciones al azar dentro de un volumen cúbico en el que caben m átomos de silicio. A continuación se van almacenando las posiciones así generadas en un array, indexando éste según la distancia de cada posición al centro del cubo. Esta lista de posiciones es usada más tarde, durante la simulación, como una plantilla que sirva durante el crecimiento de los voids, asegurándose que en todo momento la densidad de un void sea la adecuada.

| Símbolo | Nombre ddp | Valor |
|----------------------------|----------------|-----------|
| Cluster \rightarrow Void | MINVoidsSIZE | 27 átomos |
| | MAXVOIDSDIAMnm | 5 nm |

Tabla 8.10: Valores utilizados en los parámetros de un void.

Captura

Los voids capturan vacantes, que son las responsables de su crecimiento. En principio su captura está permitida siempre, y se regula la estabilidad del void mediante la re-emisión de V s capturadas.

Emisión

La frecuencia de emisión de una V desde un void se calcula con la siguiente ecuación:

$$\nu_{\text{emission}} = \nu_{0,\text{emission}} \times \exp\left(-\frac{E_{b,\text{void}}(n) + E_m(I)}{k_B T}\right), \quad (8.5)$$

en donde el prefactor $\nu_{0,\text{emission}}$ no es linear al volumen del void (linear a su vez con el tamaño n), sino con la superficie de éste. $E_{b,\text{void}}(n)$ es la energía de enlace de una nueva vacante al void, tomadas en principio de la referencia (Bongiorno and Colombo, 1998), y que se pueden ver representadas en la figura 8.10. A su vez, los valores utilizados para los voids se encuentra indicados en la tabla 8.10.

Recombinación

Los voids atrapan y recombinan los intersticiales que se encuentran dentro de su radio de captura. Una vez que todas las vacantes de un void, excepto una, han sido recombinadas o emitidas, el void se disuelve, liberando la vacante como defecto puntual.

▷ 8.4

Acumulación de daño

Durante la implantación de impurezas (generalmente a temperatura ambiente o bastante baja) apenas existe difusión debido a que los intersticiales y vacantes introducidos por la implantación generan regiones de daño donde se acumulan estas partículas de forma desordenada. A medida que la implantación continúa, y dependiendo también de la temperatura, puede ser que los I s y V s se vayan

recombinando y las regiones de dañado comiencen a desaparecer, puede que crezcan o incluso, pueden crecer tan masivamente que den lugar a regiones enteras amorfas.

En el modelo de acumulación de dañado es importante el concepto de recocido dinámico, que se puede explicar como el proceso de recombinación que puede irse produciendo durante la propia implantación de la muestra. En principio el efecto del recocido dinámico será más importante cuanto mayor sea la temperatura y más tiempo pase durante la implantación (menor velocidad de implantación). En el caso de DADOS el tiempo de recocido dinámico sucede entre cascada y cascada, es decir, cada cascada es insertada en un tiempo cero y, posteriormente, se hace un pequeño recocido a la temperatura indicada para la implantación, cuya duración dependerá de la relación entre la dosis y la velocidad de la dosis pedida.

▷ 8.4.1

Pockets amorfos

En DADOS hemos modelizado las regiones de dañado anteriormente mencionadas por medio de pockets amorfos (\bullet). Estos pockets amorfos, de los cuales puede verse una representación esquemática en la figura 8.18, atrapan todo tipo de partículas (tanto I s como V s e impurezas), previniendo así su difusión. Crecen atrapando I s y V s (como hemos comentado atrapan también impurezas, aunque estas no influyen en el tamaño del pocket, que es considerado como el número de pares IV que contiene), y decrecen con sucesivos eventos de recombinación que recombinan uno de los I s con una de las V s constituyentes, hasta que se agotan los I s o V s y desaparecen, liberando los V s o I s desparejados que quedan en forma de clusters, e incorporando las impurezas a los clusters, cuando esto es posible, o liberándolas como defectos solitarios.

Teoría en la
sec. 7.6.1,
pág. 92.

Hay que precisar que un pocket amorfo no es una región amorfa. Un pocket amorfo es una aglomeración de partículas desordenadas en una región de Si cristalina y únicamente cuando esta aglomeración alcanza un nivel determinado pasa la región entera a considerarse como amorfa. Así pues un pocket no es una zona amorfa, sólo es el (posible) precursor de ésta.

Forma

Los pocket amorfos, al igual que los clusters pequeños, presentan una forma irregular, creada por la adhesión de las diferentes partículas según van llegando. En la figura 8.19 se muestra el aspecto de algunos de estos pockets, que han sido creados a partir de una cascada de implantación.

Crecimiento

Como ya hemos adelantado, los pockets amorfos capturan cualquier tipo de defecto móvil que se sitúe en su radio de captura, es decir, tanto defectos puntuales

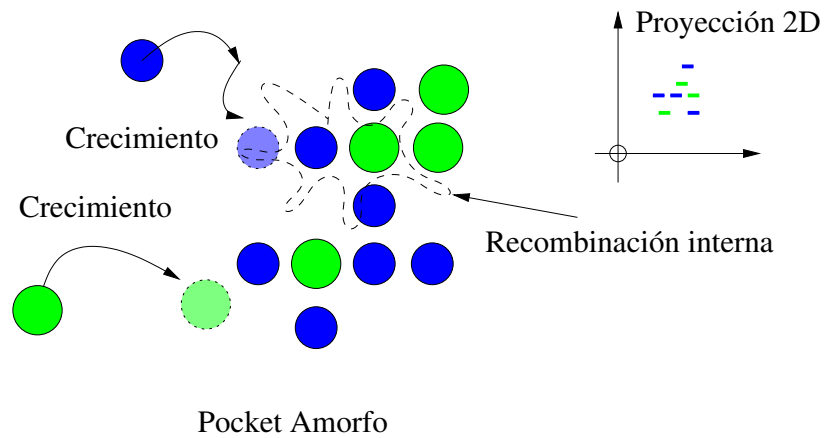


Figura 8.18: Los pockets amorfos son aglomeraciones desordenadas de partículas con una cierta barrera (generalmente baja) para su recombinación. Crecen con la captura de I s y V s y disminuyen recombinando pares IV .

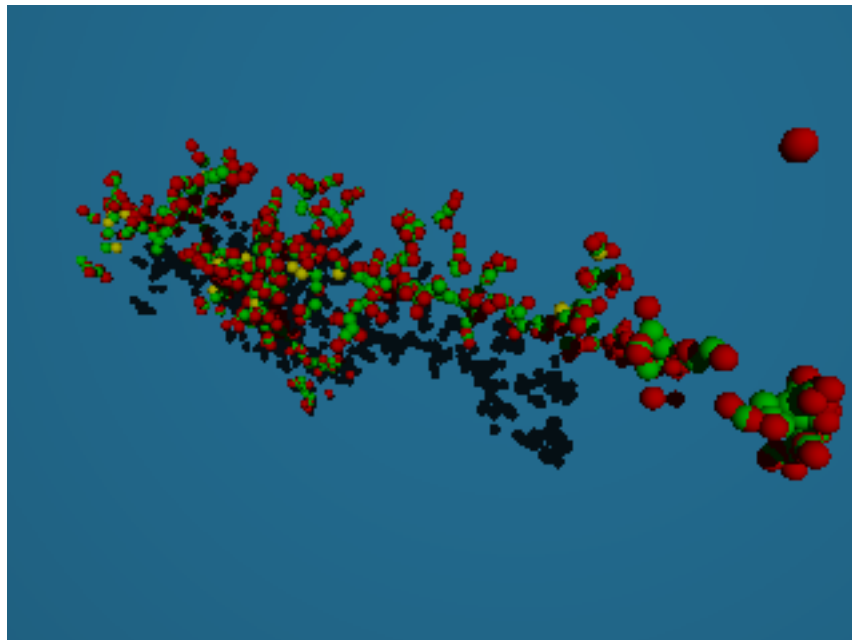


Figura 8.19: Pockets amorfos creados en una simulación de DADOS después de implantar una cascada de arsénico. Esferas rojas: Intersticiales. Verdes: Vacantes. Amarillas: Arsénico.

(I, V) como dopantes móviles ($B_i, As_i, As_V \dots$). Para los defectos puntuales I y V el pocket amorfo transforma estos defectos móviles en partículas inmóviles y las sitúa en sus arrays. Para los dopantes separa la contribución I o V del dopante (por ejemplo, As_i o As_V) y almacena los Is o Vs por un lado y el dopante por otro, si bien no cambia sus coordenadas en el espacio. Como se tratará también en el apartado 8.4.2, cuando la concentración de defectos puntuales supera cierto umbral, el pocket amorfo elimina los Is y Vs que van llegando a él (no almacena sus coordenadas) y se limita a llevar un recuento de cuántos eran, teniendo de esta forma unas partículas visibles, con efectos de visualización y para mantener la sección de captura, y otras ocultas para poder así ahorrar memoria y simular tamaños grandes.

Recombinación interna

Los pocket amorfos no emiten partículas. En su lugar tienen programado un evento de recombinación con el cual pueden eliminar uno de sus pares IV de cada vez. El I y la V a ser eliminados son elegidos de manera aleatoria, y por tanto pueden encontrarse o no espacialmente cerca uno de otro. Si el pocket amorfo tiene partículas ocultas se limita a ir decrementando el contador de éstas, si no borra de la memoria y de la simulación la partícula.

La frecuencia de recombinación depende del tamaño del pocket amorfo, el cual es definido como el número de pares IV que contiene. Dicha frecuencia se calcula como:

$$\nu_{\text{recomb}} = \nu_{0,\text{recomb}} \times n^a \times \exp\left(-\frac{E_{\text{AmPoc}}(n)}{k_B T}\right), \quad (8.6)$$

donde $\nu_{0,\text{recomb}}$ es un prefactor constante, a es un parámetro para el exponente y $E_{\text{AmPoc}}(n)$ es la energía de recombinación de un par IV en función del número de pares n del pocket amorfo. Estas energías se pueden especificar una por una, o bien calcularse, de una forma similar a la de los clusters (ver sección 8.3.1) con la siguiente ecuación:

$$E_{\text{AmPoc}}(n) = E_{\text{AmPoc},L} - \frac{(n+1)^a - n^a}{2^a - 1} (E_{\text{AmPoc},L} - E_{\text{AmPoc},S}), \quad (8.7)$$

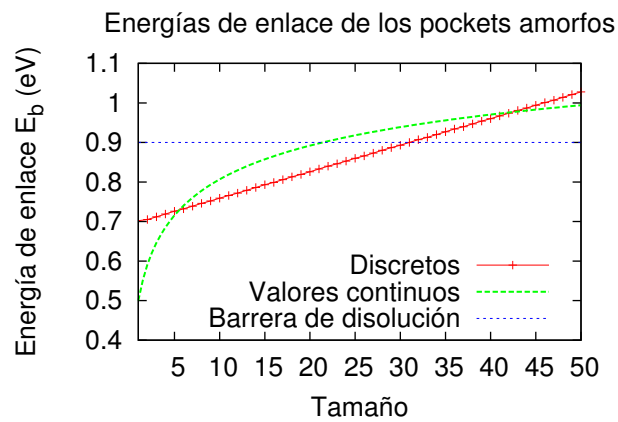
donde hay dos parámetros más, la energía para recombinar un par en un pocket muy grande, que se denomina $E_{\text{AmPoc},L}$, y la necesaria para recombinar un par IV si sólo hubiera uno, $E_{\text{AmPoc},S}$. La figura 8.20 muestra los valores que se tienen para estas energías cuando se usa la ecuación 8.7 junto con los valores expuestos en la tabla 8.11.

Interacción con impurezas

Ya hemos explicado que los pockets amorfos pueden también interaccionar con impurezas, limitándose en ese caso a almacenar éstas sin que intervengan

| Símbolo | Nombre ddp | Valor |
|----------------------------------|-------------------------|---|
| $D_{0,\text{recomb}}$ | DOAmorphousPocket | $1\text{e-}05 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_{\text{AmPoc},S}$ | EAmorphousPocketSmall | 0.5 eV |
| $E_{\text{AmPoc},L}$ | EAmorphousPocketLarge | 1.7 eV |
| a | ExponentAmorphousPocket | 0.85 |
| $E_{\text{AmPoc},\text{recomb}}$ | EAmorphousPocket | 0.9 eV |

Tabla 8.11: Valores definidos para los pockets amorfos.

Figura 8.20: Energía de recombinación de un par IV dentro de un pocket amorfo, en función del tamaño, ver texto.

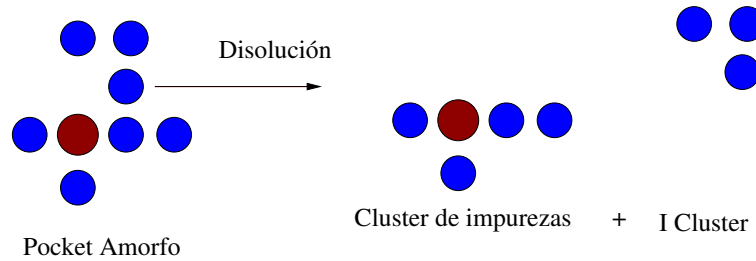


Figura 8.21: Una vez que no es posible recombinar más pares IV , el pocket amorfo se disuelve y forma pequeños clusters.

para nada más. Así pues, una impureza no aumenta el tamaño del cluster y, por tanto, no altera la frecuencia de recombinación. Cuando la impureza se mueve acompañada de un I o V , el pocket captura también dicha partícula, que es incorporada a él como si fuera un I o V normal, y por tanto este sí que influirá en posteriores cálculos del tamaño.

Disolución

Una vez que todos los pares IV de un pocket amorfo se han recombinado, el pocket amorfo espera a un nuevo evento para disolverse completamente y liberar todas las partículas desemparejadas que le queden, junto con las impurezas que capturó. La frecuencia con que se calculan dichos eventos se corresponde con:

$$\nu_{\text{dissolution}} = \nu_{0,\text{recomb}} \times \exp\left(-\frac{E_{\text{AmPoc, recomb}}}{k_B T}\right), \quad (8.8)$$

donde $E_{\text{AmPoc, recomb}}$ es la energía involucrada en el proceso final de disolución, que podría ser diferente de las de recombinación.

En el proceso final de disolución, el pocket amorfo intenta construir clusters de impurezas (ver sección 8.6.2) con los defectos puntuales que tenga disponibles, como se intenta representar en la figura 8.21. Si le siguen sobrando I s o V s los libera como clusters pequeños, de los cuales el tamaño inicial puede resultar crítico. Para una descripción más detallada de los tamaños utilizados en esta disolución se remite al lector a la descripción de cada partícula expuesta en el capítulo 9.

▷ 8.4.2

Modelo de amorfización

El proceso de amorfización, al contrario que los defectos extensos que hemos ido considerando hasta ahora, es una forma masiva de desorden de red y, como tal, no es candidato de un modelo atomístico detallado. Pese a ello, ha sido incorporado dentro de DADOS de la forma que a continuación se va a exponer.

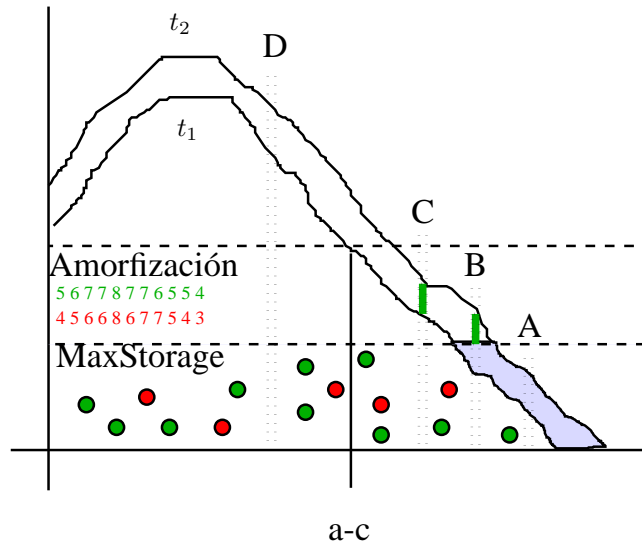


Figura 8.22: Esquema de los perfiles de daño para tiempos $t_1 < t_2$, utilizado para explicar el modelo semi-atomístico de amorfización.

Hay que comentar también que, pese a estar implementado de forma totalmente tridimensional, vamos a referirnos aquí, por simplicidad, a una única dimensión.

La figura 8.22 muestra un mismo perfil de daño en dos tiempos distintos t_1 y t_2 a lo largo de la implantación. El perfil de daño se corresponde con los perfiles de I_s y V_s que se han generado por el ion implantado, que generalmente, son difícilmente distinguibles puesto que sus valores son mucho más grandes que sus diferencias. La figura indica también un nivel máximo de daño antes de la amorfización y un nivel de almacenamiento máximo llamado *MaxStorage*. El nivel para la amorfización se sitúa habitualmente en un valor entre el 20%—30% de la densidad atómica del silicio.

Los valores concretos usados para la amorfización están expuestos en la lista 8.12.

Modo de ahorro de memoria

El nivel de *MaxStorage* se define como la máxima concentración permitida para los $I + V$ en cada caja de la simulación. Este valor es bastante más bajo que el nivel de amorfización (quizás dos órdenes de magnitud más bajo), ya que su objetivo es mantener un número de partículas en la simulación manejable y, por tanto, ahorrar memoria y tiempo de CPU, pero conservando la suficiente información como para reproducir adecuadamente los efectos de recocido dinámico.

De esta manera, en la figura 8.22 el nivel de daño en A es aún bajo, y todas las partículas I y V generadas por las cascadas serán incorporadas a la simulación atomística. En B, cuando el tiempo es t_2 , el daño ha sobrepasado el nivel de

| Símbolo | Nombre ddp | Valor |
|------------------------|------------------------|--------------------------|
| <i>MaxStorage</i> | MaxIVStorage | 1e+20 cm ⁻³ |
| Amorphizationthreshold | AmorphizationThreshold | 1.5e+22 cm ⁻³ |

Tabla 8.12: Parámetros de DADOS para el modelo de amorfización.

almacenamiento máximo, y por tanto se descartan las coordenadas concretas de las partículas nuevas y sólo se almacena su cantidad para poder actualizar los histogramas y niveles de dañado. Así pues, en C y tiempo t_2 todas las partículas por encima del nivel *MaxStorage* se borran. Por último, para la región D que está en niveles de amorfización, se descartan todos² los *I* y *V*, se etiqueta la región como amorfa y se genera un nuevo tipo de defecto (defecto amorfo) en estrecha relación con la región (caja) amorfizada. Hay que darse cuenta también de que, puesto que habrá algo de recocido dinámico entre t_1 y t_2 , las regiones B o C podrían retornar a niveles por debajo de *MaxStorage*, cosa que, en cambio, no es posible para la región amorfa D, la cual puede ya únicamente recrystalizar si se encontrara en una intercara amorfo/cristalina. Por último, decir que durante el proceso de borrar las partículas en exceso hay que ir contándolas para mantener el balance de *Is* y *Vs* adecuado.

Tal como hemos comentado ya, durante la inclusión de las nuevas partículas, sean *Is*, *Vs* o impurezas, éstas se van adosando a pockets amorfos y se produce el crecimiento de dichos pockets. Posteriormente los pares *IV* se podrán recombinar con una energía de medio eV (Marques et al., 2001) cuando estén solos y un valor más grande (Elliman et al., 1988; Jaraiz et al., 1996), pero dependiente del tamaño del amorfo, cuando formen un aglomerado mayor (ver 8.4.1).

Para calcular de manera correcta las frecuencias de recombinación de estos pockets amorfos, aunque se pueden haber despreciado las posiciones concretas de algunas partículas *Is* y *Vs*, se contabiliza el número de éstas (como si fueran partículas ocultas), de tal forma que los pockets amorfos son capaces de recrear posteriormente las partículas ocultas en posiciones aleatorias dentro de la caja a la que pertenecían, a medida que se necesiten.

▷ 8.4.3

Regiones y defectos amorfos

Las regiones amorfas surgen como consecuencia del crecimiento, por encima de una concentración determinada, de los pockets amorfos, si bien su modelo es bastante diferente. Concretamente las regiones amorfas son pequeñas cajas rectangulares de espacio a las cuales se asocia un defecto amorfo. Cuando la concentración de la caja es superior al nivel de amorfización, se crea un defecto

²Para una región sin red (amorfizada) no se puede definir la posición de un defecto puntual, al no haber una referencia (la red) para decir si es o no un defecto.

amorfo en ella y se considera que esa región de material se ha amorfizado.

Las propiedades de las cajas amorfas son:

- Se borra toda la información que se tenía sobre las partículas que estaban en la caja amorfizada, excepto su número y tipo.
- Todos los I s y V s que, en su migración, entren en una región amorfa son también eliminadas, aunque se cuenta su número (básicamente su $I - V$, para poder luego regenerar esta diferencia al recristalizar, ver sección 8.5).
- Las impurezas (especies que no sean I s o V s) que entren en una región amorfa son también borradas, guardando sólo su número y tipo.

A su vez se inserta un temporizador relacionado con los defectos amorfos, cuya misión es generar un evento de recristalización cuya inmediatez está relacionada con la temperatura.

▷ 8.5

Recristalización

Teoría en la
sec. 7.6.2,
pág. 94.

La recristalización de las regiones amorfas es implementada como un tipo especial de evento que es llevado a cabo por los defectos amorfos. Consiste en que a una temperatura dada, la interfaz amorfo-cristalina avanza hacia la parte amorfa con una velocidad v . Por tanto, si w es la anchura de las cajas, el tiempo que se necesita para recristalizar una fila de ellas será $\Delta t = w/v$.

Como ya se ha mencionado en el apartado anterior, cuando una caja se amorfiza, como por ejemplo alguna en la zona B para t_2 (ver figura 8.22), se genera un evento de recristalización para ella en $t + \Delta t$, evento que es reajustado a una nueva temporización después de que tenga lugar, o bien se cambie la temperatura. De esta manera, el cálculo de la velocidad se realiza con la siguiente ecuación:

$$v_{recrys} = V_{0,recrys} \times \exp\left(-\frac{E_{recrys}}{k_B T}\right), \quad (8.9)$$

y los parámetros usados son los especificados en la tabla 8.13. Además aunque las cajas pueden tener longitud diferente en los distintos ejes, en una primera aproximación DADOS asume que son cúbicas con una anchura promedio que de el mismo volumen que la caja real.

La figura 8.23 muestra la evolución del drenador de un transistor a medida que va recristalizando. Para ello se implantó arsénico con una energía de 35 keV, una dosis de $1 \times 10^{15} \text{ cm}^{-2}$, y se recoció durante 100 s a 600 °C.

| Símbolo | Nombre ddp | Valor |
|-----------------|------------|------------------|
| $V_{0,recryst}$ | V0recryst | $1.712e+08$ cm/s |
| $E_{recryst}$ | Erecryst | 2.7 eV |

Tabla 8.13: Valores de los parámetros de recristalización usados por DADOS en la ecuación 8.9.

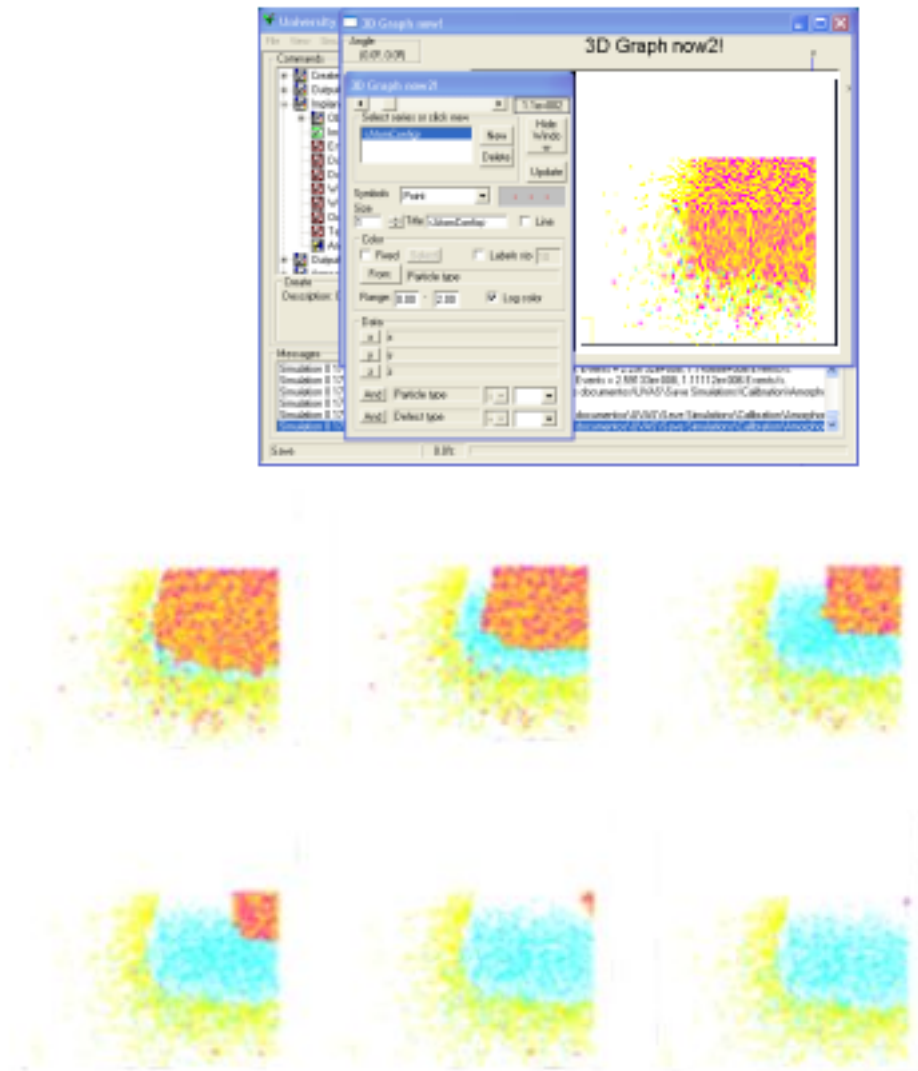


Figura 8.23: Evolución del drenador y canal de un transistor mientras recristaliza. Se ha implantado As a 35 keV y con dosis 1×10^{15} cm^{-2} , haciendo a continuación un recocido a 600°C . Se ve la evolución a $t = 0, 5, 20, 40, 60, 90$ y 100 s mientras recristaliza.

▷ 8.5.1

Suavizando las zonas amorfas

El proceso de recristalización de las regiones (cajas) amorfas recombina todos los pares IV , moviendo el exceso de Is o Vs a las regiones amorfas contiguas y dejando, por tanto, sólo los átomos de impurezas en un cristal perfecto. Durante este proceso DADOS tiene cuidado de recristalizar primero las cajas que más interfaces cristalinas tengan, para así evitar que cuando la topología de la interfaz a-c sea algo compleja, se dejen cajas amorfas sueltas que no sigan el frente de recristalización, que podrían llevar a la aparición artificial de defectos puntuales en medio de una zona recristalizada. Junto con esta estrategia el programa implementa otro mecanismo que ayude a evitar esa situación: DADOS suaviza las cajas amorfas. Para ello, cuando el nivel de dañado promedio de las cajas vecinas está por encima del de amorfización, pero hay aún un porcentaje de cajas no amorfas inferior a $1/3$, el código supone que también dichas cajas son amorfas, de forma que se evita la presencia irreal de minúsculas regiones cristalinas (1 nm^3) dentro de una zona amorfa. Estas mini-regiones podrían aparecer por meros procesos estadísticos, ya que puesto que las cajas son muy pequeñas y hay muchas de ellas, no sería raro que en alguna no hubiera partículas suficientes para detectar el umbral de material amorfo.

▷ 8.5.2

Depositando el exceso de defectos puntuales

En nuestro modelo la recristalización de una zona amorfa se logra recombinaando todos los pares IV que se encuentran en ella, procedentes de un dañado anterior. No obstante, al hacer esto es habitual encontrarse con un exceso de Is o Vs puesto que éstos no tienen por qué encontrarse emparejados. En estos casos, el exceso de defectos puntuales es desplazado hacia las zonas amorfas contiguas, de tal forma que se va generando un frente de recristalización que va arrastrando el exceso de defectos puntuales. Si este frente encuentra una superficie o intercara, dicho exceso desaparece por recombinación con la intercara, pero en el caso en el cual esto no suceda, por ejemplo cuando se ha amorfizado una zona interna custodiada por todos los lados por zonas cristalinas, se depositarán todas las impurezas arrastradas, que generalmente llevan a la formación de defectos extendidos tras un pequeño recocido.

El modelizado de este frente de recristalización es como sigue: cuando una región amorfa va a recristalizar se busca en todas las regiones vecinas para ver si existe alguna que siendo amorfa no vaya a recristalizar aún en este “turno”. Cuando se encuentra se le aporta el exceso de defectos puntuales y se recristaliza sin más problemas (en el siguiente turno la región a la que se vertió el exceso volverá a buscar otra región amorfa y así va avanzando el frente). Cuando todas las regiones vecinas amorfas van a recristalizar en este mismo turno, si hay una interfaz vecina, se recombinan todos los defectos en la interfaz, y si no, se de-

| Símbolo | Nombre ddp | Valor |
|----------------------|------------|-------|
| P_{deposit} | (interno) | 0.3 |

Tabla 8.14: Probabilidad de los dopantes para quedar en la caja donde están y no viajar con el frente de recristalización

positan como defectos puntuales. En realidad, durante este proceso de trasvase, DADOS se limita a modificar los contadores de partículas sin crear éstas, por lo que el mecanismo es bastante eficiente.

▷ 8.5.3

Las impurezas durante la recristalización

El proceso de recristalización afecta a la concentración de impurezas en las regiones recristalizadas. En general, el frente de recristalización arrastra algunas impurezas y limita el número de éstas dependiendo de su solubilidad. (Ver por ejemplo las referencias (Dokumaci et al., 1999; Venezia et al., 2003) para F y B). Estos procesos tienen lugar de dos maneras separadas:

1. Hay una probabilidad P_{deposit} de que los dopantes se queden en la caja en la que están en vez de viajar con el frente de recristalización, probabilidad que se muestra en la tabla 8.14.
2. Una vez que la caja queda cristalizada se calcula la concentración de dopantes que se dejaría en cada caja. Cuando ésta es mayor que la concentración activa máxima permitida $[A]_{\text{max}}$, que se muestra en la tabla 8.15, las partículas sobrantes se depositan como clusters de impurezas, con un número máximo de dopantes e intersticiales (o vacantes, dependiendo del tipo de cluster) de $\text{máx}_{\text{Cluster}}(A)$ y $\text{máx}_{\text{Cluster}}(I, V)$ respectivamente.

Al depositar los clusters de impurezas que estén por encima de la concentración permitida DADOS calcula las probabilidades de depositar clusters de diferentes tamaños. Básicamente para un cluster de tamaño n la probabilidad será $P(n) = P(n-1) \times P_{\text{capture}}^{n-1}$, siendo P_{capture}^{n-1} la probabilidad de que un cluster de tamaño $n-1$ capture una partícula y crezca. Dicha probabilidad es proporcional a $P_{\text{capture}}^{n-1} \propto [\text{Cluster}(n-1)] \times v_{\text{cap}}$, es decir, la concentración de clusters de tamaño $n-1$ por su volumen de captura.

| Nombre | valor ddp | Valor |
|--------------------------------------|-----------|---------------------------|
| $[As]_{\max}^{\text{recrys}}$ | (interno) | $1e+21 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de As | (interno) | 4 |
| Max. Is o Vs en clusters de As | (interno) | 0 |
| $[B]_{\max}^{\text{recrys}}$ | (interno) | $3.5e+20 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de B | (interno) | 3 |
| Max. Is o Vs en clusters de B | (interno) | 0 |
| $[C]_{\max}^{\text{recrys}}$ | (interno) | $5e+22 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de C | (interno) | 0 |
| Max. Is o Vs en clusters de C | (interno) | 0 |
| $[F]_{\max}^{\text{recrys}}$ | (interno) | $5e+22 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de F | (interno) | 0 |
| Max. Is o Vs en clusters de F | (interno) | 0 |
| $[In]_{\max}^{\text{recrys}}$ | (interno) | $5e+22 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de In | (interno) | 0 |
| Max. Is o Vs en clusters de In | (interno) | 0 |
| $[O]_{\max}^{\text{recrys}}$ | (interno) | $5e+22 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de O | (interno) | 0 |
| Max. Is o Vs en clusters de O | (interno) | 0 |
| $[P]_{\max}^{\text{recrys}}$ | (interno) | $1e+21 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de P | (interno) | 3 |
| Max. Is o Vs en clusters de P | (interno) | 0 |

Tabla 8.15: Concentración máxima de dopantes después de la recristalización

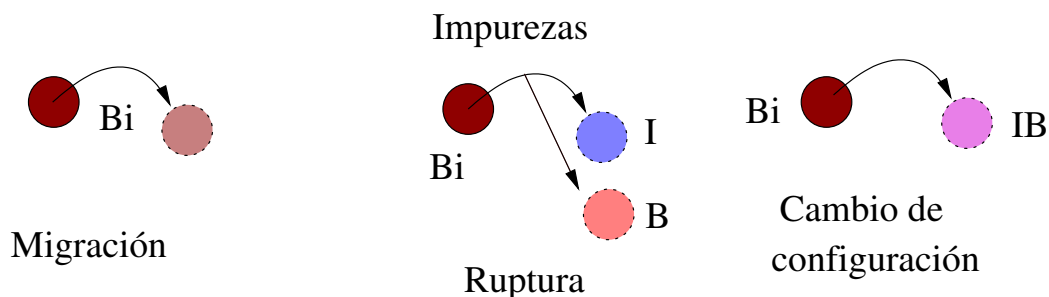


Figura 8.24: Las impurezas difunden saltando una distancia λ en direcciones ortogonales. A su vez pueden romperse en varias partículas o cambiar a otras configuraciones.

▷ 8.6

Impurezas

DADOS trata las impurezas de forma muy parecida a como trata los defectos puntuales. Por una parte las impurezas pueden formar defectos sencillos, similares en cierta medida a los I s y V s, y que son también modelizados como “defectos puntuales”, y por otra parte pueden formar defectos extensos, clusters, que comparten algo del comportamiento de los clusters de I s y V s pero admiten dos partículas diferentes, en vez de sólo una. Llamaremos a éstos clusters de impurezas.

Teoría en la
sec. 7.8,
pág. 97.

▷ 8.6.1

Las impurezas como defectos puntuales

Los átomos de impurezas son modelizados igual que los defectos puntuales I o V . Tienen sus coordenadas tridimensionales, su tipo de defecto (que es defecto puntual, ya que no son clusters) y su tipo de partícula (por ejemplo boro sustitucional, boro intersticial negativo...). Los átomos de impurezas, además del evento de salto necesario para la migración, tienen otros dos más: uno de ruptura (por ejemplo para separar un boro intersticial en boro sustitucional más un intersticial) y otro de cambio de configuración (figura 8.24), que ahora no se está utilizando.

Más detalles
en sec. A.2.3,
pág. 329.

Migración y difusión

La frecuencia de migración para las impurezas se calcula de acuerdo a la ecuación 8.1. De todas formas, la difusividad efectiva resulta ser diferente de la microscópica que se usa en la ecuación. Esto es debido a que la difusividad efectiva macroscópica medida en experimentos resulta del promedio de muchísimos saltos durante mucho tiempo, en los cuales los átomos pueden adoptar distin-

tas configuraciones. Por ejemplo, para el boro, que difunde según el denominado mecanismo de *kick-out*, microscópicamente se ve que los átomos de boro sustitucional, a los que llamaremos simplemente B , reaccionan con los intersticiales produciendo boro intersticial o B_i . El B no migra, pero el B_i si lo hace con una difusividad especificada en la ecuación 8.1. No obstante, de vez en cuando el B_i vuelve a romperse en $B + I$, donde el B vuelve a ser inmóvil hasta que logre capturar otro I y siga migrando. Así la difusividad macroscópica se relaciona, además de con la migración de la especie móvil, en este caso B_i , con la frecuencia con la que dicha especie se rompe. Concretamente:

$$D^{\text{eff}}(B) = D_0^{\text{eff}}(B) \times \exp\left(-\frac{E_{\text{diff}}(B)}{k_B T}\right), \quad (8.10)$$

donde $D_0^{\text{eff}}(B)$ depende de los prefactores de migración y ruptura de B_i , y también del prefactor de migración de I , y

$$E_{\text{diff}}(B) = E_m(B_i) + E_f(I) - E_b(B_i),$$

donde $E_m(B_i)$ es la energía de migración de B_i , $E_f(I)$ la energía de formación de I y $E_b(B_i)$ la energía de enlace de B_i . Los parámetros relacionados con la difusión de las impurezas en silicio se encuentran listados en la tabla 8.16 (las energías de migración), y en la tabla 8.19 (las energías de enlace). Los parámetros de difusión de impurezas en otros materiales que no sean silicio están en la tabla 8.17 para el óxido de silicio y 8.18 para el nitruro de silicio. La figura 8.25 ofrece un esquema energético de la formación y ruptura de un B_i . Además, DADOS no diferencia las partículas intersticiales de las sustitucionales más que por la “etiqueta” de la especie y los valores de las energías de migración, con lo cual, para hacer una especie inmóvil basta con cambiar su energía de migración a un valor muy grande (por ejemplo 5 eV) y/o poner su prefactor a 0.

Por último, puesto que DADOS asume que las especies sustitucionales se encuentran en su estado ionizado (como B^- y As^+), los parámetros de enlace (energía y prefactor) que se muestran para las especies de la tabla están expresados para las especies que se formarían después de la reacción pertinente entre intersticiales o vacantes neutros con la impureza ionizada, es decir, en nuestros ejemplos se listarán para B_i^- y As_V^+ , en vez de B_i y As_V puesto que $B^- + I \rightarrow B_i^-$ y $As^+ + V \rightarrow As_V^+$.

Ruptura

Más detalles
en sec. A.2.3,
pág. 329.

Como en parte se ha explicado anteriormente, el evento de ruptura es responsable de que las impurezas intersticiales (o vacantes) vuelvan a ser sustitucionales y liberen el I (o V) que contenían: $A_i \rightarrow A + I$, para una impureza general A que difunde por intersticiales. El modelo calcula la frecuencia de ruptura como

$$\nu_{\text{breakup}}(A_i) = \nu_{\text{breakup},0} \times \exp\left(-\frac{E_b(A_i) + E_m(I)}{k_B T}\right), \quad (8.11)$$

| Símbolo | Nombre ddp | Valor |
|-----------------|------------|---|
| $D_{m,0}(O)$ | Dm_O | $0.13 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(O)$ | Em_O | 2.53 eV |
| $D_{m,0}(C)$ | Dm_C | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(C)$ | Em_C | 5 eV |
| $D_{m,0}(B)$ | Dm_B | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(B)$ | Em_B | 5 eV |
| $D_{m,0}(F)$ | Dm_F | $0.005 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(F)$ | Em_F | 0.8 eV |
| $D_{m,0}(In)$ | Dm_In | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(In)$ | Em_In | 5 eV |
| $D_{m,0}(As)$ | Dm_As | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(As)$ | Em_As | 5 eV |
| $D_{m,0}(Ph)$ | Dm_Ph | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(Ph)$ | Em_Ph | 5 eV |
| $D_{m,0}(O_V)$ | Dm_OV | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(O_V)$ | Em_OV | 5 eV |
| $D_{m,0}(C_i)$ | Dm_Ci | $0.001 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(C_i)$ | Em_Ci | 0.87 eV |
| $D_{m,0}(B_i)$ | Dm_Bi | $0.001 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(B_i)$ | Em_Bi | 0.25 eV |
| $D_{m,0}(In_V)$ | Dm_InV | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(In_V)$ | Em_InV | 5 eV |
| $D_{m,0}(In_i)$ | Dm_Ini | $1.6\text{e-}06 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(In_i)$ | Em_Ini | 0.9 eV |
| $D_{m,0}(As_V)$ | Dm_AsV | $0.00021 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(As_V)$ | Em_AsV | 1.3 eV |
| $D_{m,0}(As_i)$ | Dm_Asi | $0.004 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(As_i)$ | Em_Asi | 1.8 eV |
| $D_{m,0}(P_V)$ | Dm_PhV | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(P_V)$ | Em_PhV | 5 eV |
| $D_{m,0}(P_i)$ | Dm_Phi | $1\text{e-}06 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(P_i)$ | Em_Phi | 0.8 eV |

Tabla 8.16: Valores de los parámetros de DADOS para la difusión de impurezas en silicio.

| Símbolo | Nombre ddp | Valor |
|---------------|------------|---|
| $D_{m,0}(B)$ | Dm_BOxide | $0.00312 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(B)$ | Em_BOxide | 3.93 eV |
| $D_{m,0}(As)$ | Dm_AsOxide | $0.000294 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(As)$ | Em_AsOxide | 3.15 eV |
| $D_{m,0}(Ph)$ | Dm_PhOxide | $14.2 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(Ph)$ | Em_PhOxide | 4.86 eV |

Tabla 8.17: Valores de los parámetros de DADOS para la difusión de impurezas en óxido de silicio.

| Símbolo | Nombre ddp | Valor |
|---------------|--------------|---|
| $D_{m,0}(B)$ | Dm_BNitride | $0.00312 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(B)$ | Em_BNitride | 3.93 eV |
| $D_{m,0}(As)$ | Dm_AsNitride | $0.000294 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(As)$ | Em_AsNitride | 3.15 eV |
| $D_{m,0}(Ph)$ | Dm_PhNitride | $14.2 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(Ph)$ | Em_PhNitride | 4.86 eV |

Tabla 8.18: Valores de los parámetros de DADOS para la difusión de impurezas en nitruro de silicio.

donde $E_b(A_i)$ es la energía de enlace de A_i , como se representa en la figura 8.25. Los parámetros para las diferentes especies implementadas se encuentran expuestos en la tabla 8.16.

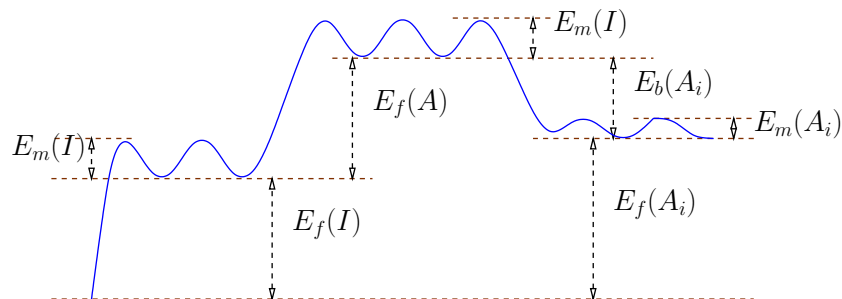


Figura 8.25: Energías relacionadas con la reacción $A_i \rightarrow A + I$.

| Símbolo | Nombre ddp | Valor |
|-------------------|------------|---------------------------------------|
| $D_{b,0}(O_V)$ | Db_OV | $0.01 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_b(O_V)$ | Eb_OV | 2.2 eV |
| $D_{b,0}(C_i)$ | Db_Ci | $2.5 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_b(C_i)$ | Eb_Ci | 1.5 eV |
| $D_{b,0}(B_i^-)$ | Db_BiM | $0.37 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_b(B_i^-)$ | Eb_BiM | 0.1 eV |
| $D_{b,0}(In_V^-)$ | Db_InVM | $0.0015 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_b(In_V^-)$ | Eb_InVM | 1.6 eV |
| $D_{b,0}(In_i^-)$ | Db_IniM | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_b(In_i^-)$ | Eb_IniM | 1.6 eV |
| $D_{b,0}(As_V^+)$ | Db_AsVP | $0.001 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_b(As_V^+)$ | Eb_AsVP | 1.01 eV |
| $D_{b,0}(As_i^+)$ | Db_AsiP | $0.005 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_b(As_i^+)$ | Eb_AsiP | 0.5 eV |
| $D_{b,0}(P_V^+)$ | Db_PhVP | $0.005 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_b(P_V^+)$ | Eb_PhVP | 0.6 eV |
| $D_{b,0}(P_i^+)$ | Db_PhiP | $0.25 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_b(P_i^+)$ | Eb_PhiP | 0.7 eV |

Tabla 8.19: Valores de los parámetros de DADOS para formación de enlaces.

▷ 8.6.2

Clusters de impurezas

Introducción

Teoría en la
sec. 7.8.2,
pág. 98.

Los clusters de impurezas son defectos extendidos que forman agregados estables de impurezas junto con intersticiales o vacantes. DADOS implementa clusters de impurezas de boro, arsénico, carbono, flúor, oxígeno y fósforo.

La figura 8.26 muestra las distintas reacciones posibles que tienen lugar en los clusters de impurezas, las cuales se explican a continuación.

Hay que decir también que para DADOS, por ser un simulador kMC sin red (non-lattice), la notación $A_n I_m$ ($A_n V_m$) representa cualquier configuración en la cual hay n átomos de “A” y m partículas intersticiales (o vacantes). Las partículas en posición intersticial pueden ser impurezas intersticiales o intersticiales de silicio. Así pues la configuración particular en la que se encuentren dichas partículas es despreciada por el modelo, que hace la aproximación de que todas las configuraciones con igual n y m pueden tratarse con un único conjunto de valores energéticos. Si hubiera más de una configuración, dicho valor sería un valor promedio o efectivo de todas ellas. Puesto que se está descartando la configuración particular de los clusters $A_n I_m$, vamos a considerar que los n A son impurezas sustitucionales (eléctricamente inactivas) y que además hay m intersticiales

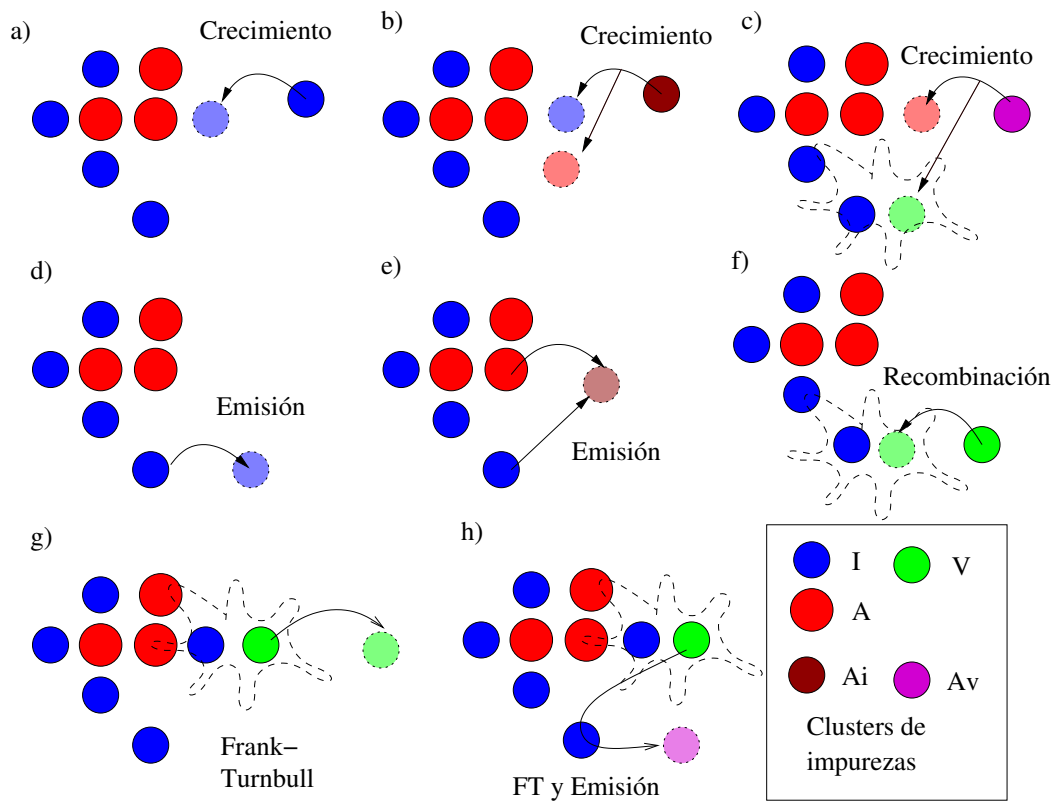
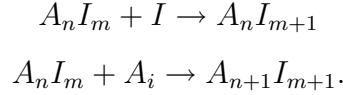


Figura 8.26: Los clusters de impurezas son aglomeraciones desordenadas de dopantes junto con I s (o V s). DADOS almacena separadamente los I s (V s) de las impurezas en estos clusters. Los mecanismos de crecimiento y decrecimiento se relacionan con la captura de nuevas partículas, la emisión de éstas, la recombinación, y el mecanismo de Frank-Turnbull.

propios del silicio, pero se trata únicamente de un convenio para facilitar el modelizado de estos clusters.

Captura

Los clusters de impurezas crecen atrapando nuevos defectos puntuales o impurezas. Concretamente pueden atrapar I s (V s) y A_i (A_v), para clusters tipo $A_n I_m$ ($A_n V_m$):



Cuando se captura un I (V) DADOS renombra su tipo de defecto a “BinComplex³” y almacena la nueva partícula como un intersticial (o vacante). Cuando atrapa un A_i (A_v) DADOS almacena dos partículas, un I (V) y un A , y destruye la partícula original A_i (A_v).

El caso del flúor está tratado de una manera un poco especial. Como ya se ha comentado no hay F sustitucional en el programa, de tal forma que el flúor intersticial es simplemente llamado F , y almacenado en un array separado del que usa para almacenar las V s en el cluster $F_n V_m$.

La captura de partículas por parte de los clusters de impurezas no es automática, como sí lo es para pequeños clusters de I s o V s y $\{311\}$ s. Esto es debido a que las energías de binding de los $\{311\}$, por ejemplo, son siempre favorables a la inclusión de una nueva partícula, pero para los clusters de impurezas no siempre tiene por que ser así. Estos clusters pueden crecer en impurezas o en I s o V s, y no siempre el crecimiento es favorable. Por ello DADOS calcula la probabilidad de que un cluster $A_n I_m$ atrape una partícula A , I o A_i como

$$P_{\text{capture}}(A) = \exp\left(-\frac{E_{\text{capture}}(A)}{k_B T}\right), \quad (8.12)$$

donde

$$E_{\text{capture}}^{A_n I_m}(B) = E_{\text{barrier}}(A_n I_m) + \text{máx}(0, -E_b^{A_n I_m}(B)),$$

siendo $E_{\text{barrier}}(A_n I_m)$ una barrera opcional⁴ a la reacción.

A su vez las energías de enlace $E_b^{A_n I_m}$ se calculan a partir de las energías potenciales totales:

$$\begin{aligned} E_b^{A_n I_m}(I) &= E_{\text{pot}}(A_n I_{m+1}) - E_{\text{pot}}(A_n I_m) \\ E_b^{A_n I_m}(A_i) &= E_{\text{pot}}(A_i) + E_{\text{pot}}(A_{n+1} I_{m+1}) - E_{\text{pot}}(A_n I_m) \end{aligned} ,$$

y estas energías son especificadas como parámetros para DADOS, como se verá en el capítulo 9. Por último, cuando una energía en particular no se especifica (es

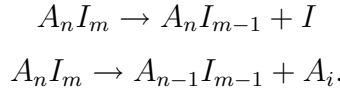
³Notación particular usada por DADOS para los clusters de impurezas.

⁴Aunque ahora mismo no se están utilizando.

decir, no existe una entrada en el fichero de parámetros para una configuración en particular), se asume que dicha configuración no es estable, y se la asigna un valor positivo de 5 eV de energía potencial para evitar que se formen clusters de esas características.

Emisión

Los clusters $A_n I_m$ ($A_n V_m$) emiten partículas tipo I (V) y A_i (A_v):



Las frecuencias de emisión de estas partículas son calculadas como

$$\nu_{\text{emission}}(B) = \nu_{\text{emission},0} \times \exp\left(-\frac{E_{\text{emission}}(B)}{k_B T}\right), \quad (8.13)$$

siendo

$$E_{\text{emission}}(B) = E_m(B) + E_{\text{barrier}}(A_n I_m) + \text{máx}(0, E_b^{A_n I_m}(B)),$$

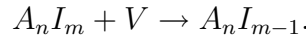
y donde $E_b^{A_n I_m}(B)$ ya ha sido expuesto más arriba, y $E_m(B)$ es la energía de migración de la especie B emitida.

Después de emitir una partícula los clusters de impurezas comprueban si el número de partículas es suficiente para que el cluster continúe. Para ello se requieren al menos dos dopantes, o bien un dopante pero más de un intersticial (o vacante). Si no hay suficientes partículas el cluster de impurezas se disuelve, dependiendo del número de partículas, en:

- Si no hay dopantes: el cluster de impurezas se transforma en otro de I s o V s.
- Si queda una impureza y un I (o V) el clusters desaparece dejando una partícula A_i (o A_v).

Recombinación

Los clusters de impurezas $A_n I_m$ atrapan V para recombinarlas internamente con I :



La probabilidad de captura es calculada de acuerdo a la ecuación 8.12 donde las energías de captura son:

$$\begin{aligned} E_{\text{capture}}^{A_n I_m}(V) &= E_b^{A_n I_m}(V) - E_{\text{pot}}(V) - E_{\text{pot}}(I) \\ E_b^{A_n I_m}(V) &= E_{\text{pot}}(A_n I_{m-1}) - E_{\text{pot}}(A_n I_m) \end{aligned},$$

ya que la recombinación requiere la destrucción de un par IV , con la consiguiente liberación de energía. Además, al disminuir el tamaño del cluster es necesario un chequeo de las partículas que quedan, como ya se ha comentado en la sección 8.6.2.

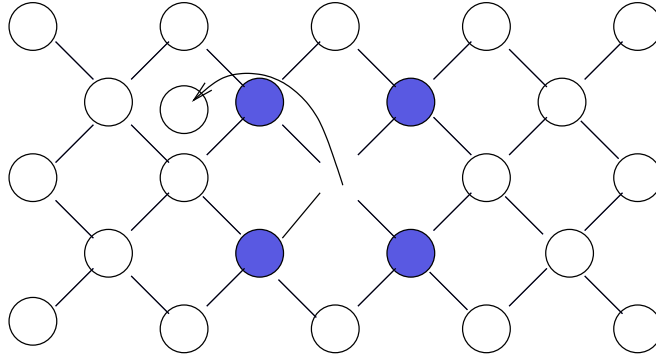
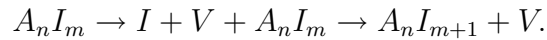


Figura 8.27: Emisión Frank-Turnbull: El cluster emite un intersticial generando también una vacante.

Frank-Turnbull

El mecanismo de Frank-Turnbull consiste en la emisión de una V (I) desde un cluster de impurezas con intersticiales (vacantes) tipo $A_n I_m$ ($A_n V_m$) gracias a la formación de un par de Frenkel (ver figura 8.27):



La implementación del mecanismo de Frank-Turnbull es necesaria para mantener la reversibilidad microscópica: puesto que los clusters de impurezas son capaces de recombinar V s (I s), se requiere también que puedan emitir V s (I s). Esta emisión Frank-Turnbull es un proceso en general poco favorable (energéticamente hablando), puesto que la formación de un par IV supone un gran aporte de energía que pocas veces es compensado, por lo que se trata de un evento que raramente sucede aunque, para algunas configuraciones particulares de clusters, debido a sus diferencias energéticas, puede ser necesario o incluso imprescindible.

La frecuencia de emisión de V s (I s) se calcula también con la ecuación 8.13, siendo

$$E_{\text{emission}}(V) = E_m(V) + \text{máx}(0, E_b^{A_n I_m}(V)),$$

y donde $E_b^{A_n I_m}(V)$ es

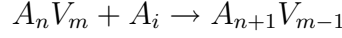
$$E_b^{A_n I_m}(V) = E_{\text{pot}}(I) + E_{\text{pot}}(V) + E_{\text{pot}}(A_n I_{m+1}) - E_{\text{pot}}(A_n I_m).$$

Recombinación complementaria

Algunas impurezas difunden a través de intersticiales y vacantes. Por ejemplo, el arsénico difunde a través de As_i y As_V . En estos casos, los clusters de impurezas (como el $As_n V_m$) pueden recombinar no sólo I s, sino también las impurezas que difunden vía intersticiales. En el ejemplo del arsénico ello supondría

que la reacción $As_nV_m + As_i \rightarrow As_{n+1}V_{m-1}$ también tiene que ser tomada en consideración.

Así pues, la reacción



tiene lugar con una probabilidad:

$$P_{\text{capture}}(A_i) = \begin{cases} \exp\left(-\frac{E_{\text{capture}}(A_i)}{k_B T}\right), & E_{\text{capture}}(A_i) > 0 \\ 1, & E_{\text{capture}}(A_i) \leq 0 \end{cases}. \quad (8.14)$$

Como en otros casos, las energías de recombinación $E_{\text{recomb}}^{A_nV_m}(A_i)$ se calculan a partir de las energías potenciales totales:

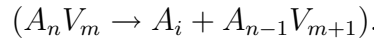
$$E_{\text{recomb}}^{A_nV_m}(A_i) = \begin{cases} E_{\text{pot}}(A_{n+1}V_{m-1}) - E_{\text{pot}}(A_nV_m) - \\ E_f(V) - E_f(I) - E_{\text{pot}}(A_i), & m > 0 \\ E_{\text{pot}}(A_{n+1}V_m) - E_{\text{pot}}(A_nV_m) - \\ E_{\text{pot}}(A_i) + E_m(I) - E_m(A_i), & m = 0, n > 1 \\ E_{A_i \rightarrow A}, & m = 0, n = 1 \end{cases},$$

donde $E_{A_i \rightarrow A}$ es un parámetro interno que no se puede cambiar.

Emisión complementaria

Las impurezas, como el arsénico, el cual difunde a través de mecanismos intersticiales (As_i) y vacantes (As_V), deben ser capaces de reaccionar y ser emitidas por los clusters de impurezas (As_nV_m). La emisión de As_V ha sido ya considerada como el caso natural, puesto que el cluster contiene ambos As y V , pero en cambio la emisión de As_i es un poco más especial, ya que supone la creación de un par IV para poder así emitir el As y el I , como se representa en la figura 8.28.

Para el modelo general, cuando una especie A puede difundir mediante A_V y A_i y sus clusters son A_nI_m (A_nV_m), la emisión complementaria está dada por



La frecuencia de esta emisión también se calcula a través de la ecuación 8.13, siendo

$$E_{\text{emission}}(A_V) = E_m(A_V) + \text{máx}(0, E_b^{A_nI_m}(A_V)),$$

y donde

$$E_b^{A_nI_m}(A_V) = E_{\text{pot}}(I) + E_{\text{pot}}(V) + E_{\text{pot}}(A_{n-1}V_{m+1}) - E_{\text{pot}}(A_nI_m).$$

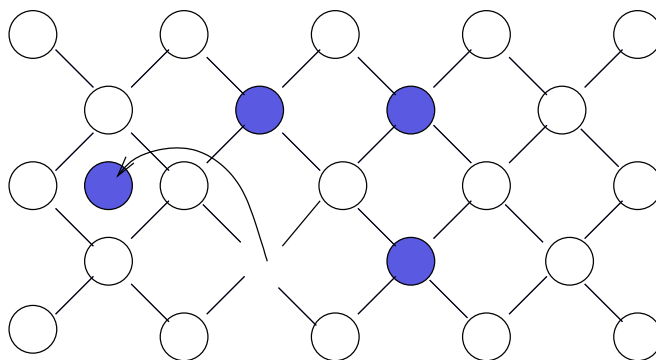


Figura 8.28: Emisión complementaria: El cluster genera un par IV , almacena la vacante y emite el intersticial junto con una impureza.

▷ 8.7

Intercaras entre materiales

Llamamos intercara a la extensión plana comprendida entre dos materiales diferentes. Uno de estos materiales suele ser silicio, y el otro puede no ser nada (*ambient*), o ser SiO_2 o nitruro de silicio. La interfaz más importante de éstas es la $SiO_2 - Si$.

Teoría en la
sec. 7.7,
pág. 95.

Puesto que, como ya se trató en la sección 8.1.3, DADOS divide el espacio en pequeñas cajas, y a cada una de estas cajas asigna un único material, las intercaras serán la sucesión de las caras comunes entre cajas de distinto material (ver figura 8.29).

La importancia de las intercaras es grande: se encargan de establecer la concentración de equilibrio para los defectos puntuales propios del silicio (I , V) y la solubilidad de impurezas y dopantes, de hecho hay un modelo distinto para cada una de estas situaciones. Además DADOS asume que las interacciones son únicamente con especies neutras, para que así no haya dependencias con el nivel de Fermi (ver modelo de carga en la sección 8.8). Las interacciones permitidas o prohibidas en un primer nivel, que no se pueden cambiar, están expuestas en la tabla 8.20 y, posteriormente en un segundo nivel pueden ser reguladas a través de un conjunto de parámetros.

Más detalles
en sec. A.3.3,
pág. 333.

▷ 8.7.1

Modelo de intercaras para los defectos puntuales propios del Silicio

Longitud de recombinación

Cuando la concentración de intersticiales o vacantes está por encima de su concentración en equilibrio, y debido a la captura por parte de la superficie hay

Más detalles
en sec. A.3.1,
pág. 331.

| Nombre | | Nombre | |
|------------|----|-------------|----|
| I | sí | V | sí |
| As | no | B | no |
| C | no | F | sí |
| In | no | O | no |
| P | no | $AsNitride$ | sí |
| $BNitride$ | sí | $PNitride$ | sí |
| $AsOxide$ | sí | $BOxide$ | sí |
| $POxide$ | sí | As^0 | no |
| As_i | sí | As_i^+ | no |
| As_V | sí | As_V^- | no |
| As_V^+ | no | B^0 | no |
| B_i | sí | B_i^- | no |
| B_i^+ | no | B_2I_2 | no |
| BI_2 | no | C_i | sí |
| I_2 | no | I^- | no |
| I^+ | no | In_i | no |
| In_i^- | no | In_V | no |
| In_V^- | no | O_V | no |
| P^0 | no | P_i | sí |
| P_i^+ | no | P_V | sí |
| P_V^- | no | P_V^+ | no |
| V^- | no | V^{--} | no |
| V^+ | no | V^{++} | no |

Tabla 8.20: Interacciones permitidas y prohibidas con las intercaras.

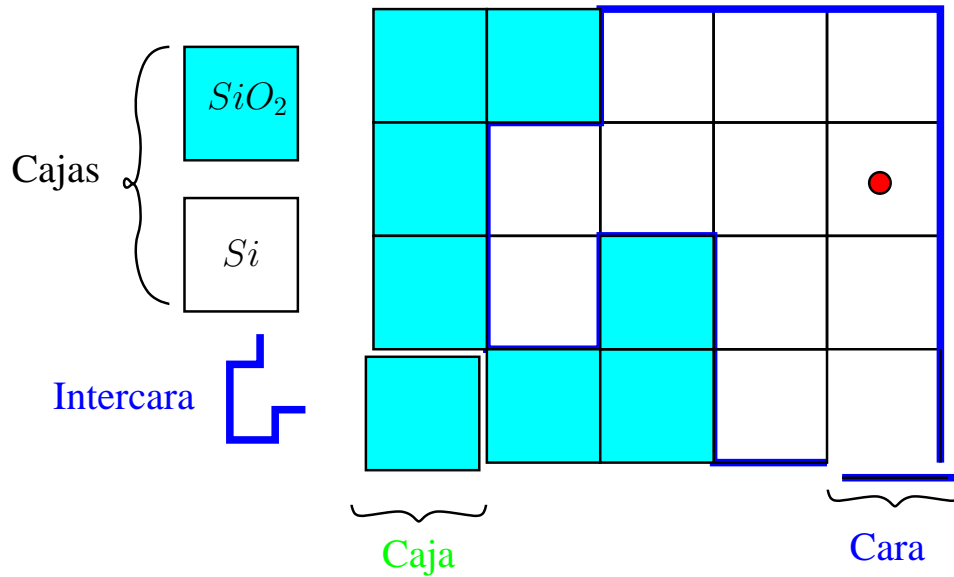


Figura 8.29: Las intercaras de DADOS son el conjunto de caras entre cajas con materiales distintos.

un gradiente de éstos hacia la superficie que tendería a restablecer dicho equilibrio, se define longitud de recombinación L_r como la distancia negativa del eje x que se necesitaría para lograr el valor en equilibrio, tal y como se ve en la figura 8.30. El significado microscópico de L_r se puede asociar con la probabilidad para los I_s y V_s de que sean capturados en las intercaras:

$$P_{\text{trap}} = \frac{\lambda}{L_r}. \quad (8.15)$$

Cuanto más pequeño es L_r más se comporta la intercara como un sumidero de partículas. Para el caso concreto de la intercara $SiO_2 - Si$ los resultados experimentales confirman que se trata de un sumidero casi perfecto con $L_r < 5$ nm (Lim et al., 1995; Cowern et al., 1999a). Estos valores de longitud de recombinación se pasan a DADOS como parámetros externos, listados en la tabla 8.21.

Captura

Las intercaras captura I_s y V_s con la probabilidad expresada en la ecuación 8.15. Cuando el valor de L_r es 0 esto supone que la probabilidad de captura es 1 y por tanto DADOS consideraría la intercara como un sumidero perfecto. Los valores concretos utilizados están disponibles en la tabla 8.21.

Más detalles
en sec. A.3.2,
pág. 332.

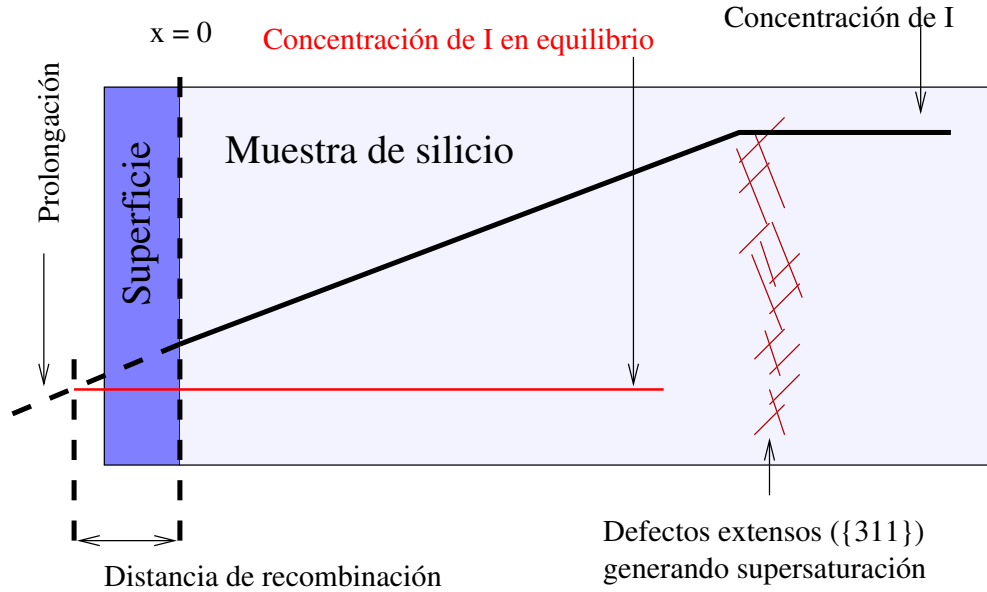


Figura 8.30: La longitud de recombinación (L_r) sirve para modelizar cómo de buena es una intercara atrapando defectos puntuales. L_r es la distancia entre la intercara y el punto en el cual la prolongación de la concentración de los defectos puntuales se encontraría con su valor en equilibrio.

| Símbolo | Nombre ddp | Valor |
|---|------------|--------------------------------------|
| $D_{\text{emission},0}^{\text{interface}}(V)$ | DOFS_V | $800 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_f(V)$ | Eform_V | 3.8 eV |
| $L_r(V)$ | RecLnm_V | 0 nm |
| $D_{\text{emission},0}^{\text{interface}}(I)$ | DOFS_I | $50000 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_f(I)$ | Eform_I | 3.8 eV |
| $L_r(I)$ | RecLnm_I | 0 nm |

Tabla 8.21: Parámetros de DADOS relacionados con la recombinación y emisión de defectos puntuales (I , V).

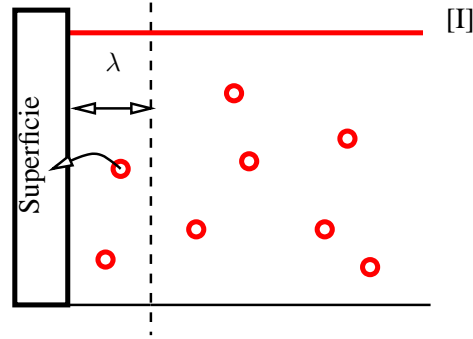


Figura 8.31: El flujo de I que se recombina en la superficie será el número de intersticiales contenidos en una distancia de salto de la intercara ($[I] \times \lambda$) que salten a la superficie por segundo ($1/6 \times \nu_m$) cuando la probabilidad de recombinación es 1.)

Emisión

El flujo de I s que van hacia la intercara, como se muestra en la figura 8.31, es

$$F = \frac{1}{6} \nu_m \times [I] \times \lambda.$$

Más detalles
en sec. A.3.3,
pág. 333.

En equilibrio la superficie debe emitir la misma cantidad de intersticiales que recombina, y consecuentemente la frecuencia de emisión será:

$$\nu_{\text{emisión}}^{\text{interface}}(I) = \underbrace{\frac{2}{a^2} Y Z}_{\text{Sitios en la intercara}} \times P_{\text{trap}} \times \nu_{\text{emisión},0}^{\text{interface}}(I) \times \exp\left(-\frac{E_m(I) + E_f(I)}{k_B T}\right), \quad (8.16)$$

donde $E_f(I)$ es la energía de formación de I . Consecuentemente la concentración de I s en equilibrio será:

$$[I]^* = \frac{2}{\lambda} \times \nu_{\text{emisión},0}^{\text{interface}}(I) \times \exp\left(-\frac{E_f(I)}{k_B T}\right). \quad (8.17)$$

La emisión de defectos puntuales desde las intercaras es modelizada con un temporizador (un timer) que se programa cuando la simulación comienza, y se reprograma cada vez que se emite un defecto puntual o se cambia la temperatura. Estos temporizadores son calculados con una distribución aleatoria que, en promedio, da el periodo de emisión requerido.

▷ 8.7.2

Impurezas

Las intercaras tienen parte de la responsabilidad de mantener el nivel de impurezas por debajo o igual a su solubilidad en cada material, y lo consiguen

atrapando dichas impurezas y emitiendo algunas de ellas con las frecuencias adecuadas, para así lograr un adecuado equilibrio de emisión/captura. El modelo de interacción de las intercaras con las impurezas de DADOS ha sido implementado siguiendo las ideas directrices del modelo de Oh-Ward (Oh and Ward, 1998). En dicho modelo el flujo de dopantes hacia las intercaras es

$$F = C_s h \left(1 + \frac{e}{a} \frac{\sigma_F}{\sigma_{max}} - \sum_{\text{species}} \frac{\sigma_F}{\sigma_{max}} \right) - h \kappa \sigma_F, \quad (8.18)$$

donde

$$h = a \sigma_{max}$$

y

$$\kappa = (e/a) \times (C_{max}/\sigma_{max});$$

σ_{max} es la densidad total de atrape y a y e denotan las probabilidades de absorción y emisión. C_s describe a su vez la concentración de solubilidad del dopante en el material. Los valores para la solubilidad, para h y para e/a son calculados de acuerdo a un plot de Arrhenius

$$\begin{aligned} C_{\text{Sol}}(X) &= C_{0,\text{Sol}}(X) \times \exp\left(-\frac{E_{\text{Sol}}(X)}{k_B T}\right) \\ h(X) &= h_0(X) \times \exp\left(-\frac{E_h(X)}{k_B T}\right), \\ (e/a)(X) &= (e/a)_0(X) \times \exp\left(-\frac{E_{(e/a)}(X)}{k_B T}\right) \end{aligned} \quad (8.19)$$

donde X sería la especie particular (B, As, F, C, \dots), y los prefactores un energías requeridos para aplicar la anterior fórmula se explicitan como parámetros de DADOS, cuyos valores actuales se encuentran disponibles en la tabla 8.22. También se considera la dependencia de la solubilidad con el nivel de Fermi, como se explica en la sección 8.8.9.

Emisión

A partir de la ecuación 8.18, podemos poner el flujo total como

$$F = F_{\text{cap}} - F_{\text{emis}},$$

donde

$$F_{\text{emis}} = h \kappa \sigma_F$$

o, introduciendo en la fórmula el número de átomos atrapados

$$F_{\text{emis}} = h \times \frac{e}{a} \times g_{\text{mat}} C_{\text{Sol}} \times \frac{N}{N_{\text{max}}}. \quad (8.20)$$

| Símbolo | Nombre ddp | Valor |
|------------------------|-----------------------|---------------------------|
| $C_{0,\text{Sol}}(As)$ | Csol0_As | $2.2e+22 \text{ cm}^{-3}$ |
| $E_{\text{Sol}}(As)$ | Esol_As | 0.47 eV |
| $h_0(As)$ | Surf_h0_As | 0.0231 |
| $E_h(As)$ | Surf_Eh_As | 0.766 eV |
| $(e/a)_0(As)$ | Surf_e_a0_As | 6.29e-05 |
| $E_{(e/a)}(As)$ | Surf_Ee_a_As | -0.738 eV |
| $\sigma_{max}(As)$ | SurfMaxTrap_As | $2e+14 \text{ cm}^{-2}$ |
| $g_{\text{Nit},0}(As)$ | Segreg_pref_AsNitride | 30 |
| $E_{\text{Nit},g}(As)$ | Segreg_eV_AsNitride | 0 eV |
| $g_{\text{Oxi},0}(As)$ | Segreg_pref_AsOxide | 30 |
| $E_{\text{Oxi},g}(As)$ | Segreg_eV_AsOxide | 0 eV |
| $C_{0,\text{Sol}}(B)$ | Csol0_B | $1e+24 \text{ cm}^{-3}$ |
| $E_{\text{Sol}}(B)$ | Esol_B | 1 eV |
| $h_0(B)$ | Surf_h0_B | 0.0166 |
| $E_h(B)$ | Surf_Eh_B | 0.486 eV |
| $(e/a)_0(B)$ | Surf_e_a0_B | 0.178 |
| $E_{(e/a)}(B)$ | Surf_Ee_a_B | -0.086 eV |
| $\sigma_{max}(B)$ | SurfMaxTrap_B | $2e+14 \text{ cm}^{-2}$ |
| $g_{\text{Nit},0}(B)$ | Segreg_pref_BNitride | 30 |
| $E_{\text{Nit},g}(B)$ | Segreg_eV_BNitride | 0 eV |
| $g_{\text{Oxi},0}(B)$ | Segreg_pref_BOxide | 30 |
| $E_{\text{Oxi},g}(B)$ | Segreg_eV_BOxide | 0 eV |
| $C_{0,\text{Sol}}(P)$ | Csol0_Ph | $1e+20 \text{ cm}^{-3}$ |
| $E_{\text{Sol}}(P)$ | Esol_Ph | 0 eV |
| $h_0(P)$ | Surf_h0_Ph | 0.715 |
| $E_h(P)$ | Surf_Eh_Ph | 1.75 eV |
| $(e/a)_0(P)$ | Surf_e_a0_Ph | 0.004 |
| $E_{(e/a)}(P)$ | Surf_Ee_a_Ph | -0.37 eV |
| $\sigma_{max}(P)$ | SurfMaxTrap_Ph | $6.8e+14 \text{ cm}^{-2}$ |
| $g_{\text{Nit},0}(P)$ | Segreg_pref_PhNitride | 30 |
| $E_{\text{Nit},g}(P)$ | Segreg_eV_PhNitride | 0 eV |
| $g_{\text{Oxi},0}(P)$ | Segreg_pref_PhOxide | 30 |
| $E_{\text{Oxi},g}(P)$ | Segreg_eV_PhOxide | 0 eV |
| $C_{0,\text{Sol}}(F)$ | Csol0_F | 0 cm^{-3} |
| $E_{\text{Sol}}(F)$ | Esol_F | 0 eV |
| $h_0(F)$ | Surf_h0_F | 100 |
| $E_h(F)$ | Surf_Eh_F | 0 eV |
| $(e/a)_0(F)$ | Surf_e_a0_F | 0.004 |
| $E_{(e/a)}(F)$ | Surf_Ee_a_F | -0.37 eV |
| $\sigma_{max}(F)$ | SurfMaxTrap_F | $1e+18 \text{ cm}^{-2}$ |

Tabla 8.22: Parámetros de DADOS relacionados con la solubilidad de dopantes e impurezas.

C_{Sol} es la solubilidad del dopante en silicio y g_{mat} su coeficiente de segregación respecto al silicio, que se calculan como

$$g_{\text{mat}} = g_{\text{mat},0} \times \exp -E_{\text{mat},g}. \quad (8.21)$$

Para el Si será trivialmente 1, y para el resto de los materiales tendrá los valores dados en la tabla 8.22.

Captura

La probabilidad de captura será

$$P_{\text{cap}} = \frac{\lambda}{D} h \times \left(1 - \sum_{\text{As,B,P...}} \frac{N}{N_{\text{max}}} + \frac{e}{a} \frac{N}{N_{\text{max}}} \right). \quad (8.22)$$

Cuando $P_{\text{cap}} > 1$ se normalizan las probabilidades de captura y emisión, poniendo

$$P_{\text{cap}}^{\text{new}} = 1$$

y

$$F_{\text{emis}}^{\text{new}} = F_{\text{emis}} / P_{\text{cap}}.$$

Estado estacionario

Con el modelo aquí expuesto, y suponiendo que únicamente tenemos una especie dopante (boro) y concentraciones en estado estacionario, podemos calcular la relación entre el número de átomos de dopante, (boro) y el máximo número de ellos que podría haber:

$$\frac{N_B}{N_{\text{max}}} = \frac{1}{1 + \frac{e}{a} \left(\frac{B_{\text{Sol}}}{B} - 1 \right)}, \quad \frac{e}{a} \leq 1,$$

tal y como se representa en la figura 8.32.

Paso a otros materiales

Las impurezas de boro, arsénico y fósforo están modelizadas en DADOS de tal manera que son capaces de difundir a través de las intercaras y pasar a otros materiales. Puesto que las propiedades de estas especies que se conocen para otros materiales son bastante reducidas, únicamente se ha implementado la difusión de estas partículas, sin interacciones ni formación de clusters, para los materiales de óxido y nitruro. Así mismo no se han implementado defectos puntuales propios del silicio (intersticiales, vacantes) para materiales que no sean el silicio cristalino.

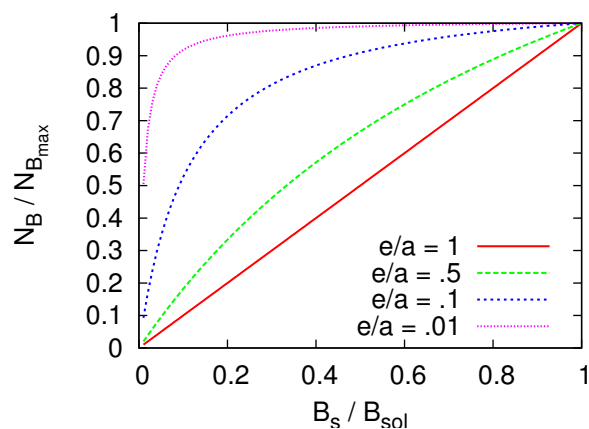


Figura 8.32: Concentración de átomos de boro sustitucional frente a su máxima concentración en estado estacionario y siguiendo el modelo de Oh-Ward.

▷ 8.7.3

Oxidación y nitrificación

La oxidación y nitrificación no están aún implementadas en nuestro simulador kMC si bien podrían ser simuladas, en una primera aproximación, cambiando las longitudes de recombinación y energías de formación para los defectos puntuales propios del silicio I y V .

Teoría en la
sec. 7.7.2,
pág. 95.

▷ 8.8

Modelo de carga

Los defectos puntuales (I , V) y las impurezas (B , As , O , C , P , F) pueden aparecer con distintos estados de carga. Por ejemplo, un intersticial puede ser neutro (I^0) pero también puede encontrarse con un estado de carga positivo o negativo (I^+ e I^-). A su vez las vacantes pueden ser neutras, positivas, negativas o doblemente positivas o negativas (V^0 , V^+ , V^{++} , V^- y V^{--}). Para las distintas impurezas se pueden conseguir distintos estados de carga, de los cuales se muestra una lista completa en la tabla 8.23.

Teoría en la
sec. 7.4.3,
pág. 85.

Se podría haber modelizado la carga de muchas formas, de las cuales la más intuitiva habría sido añadir una especie de “etiqueta” a cada partícula indicando su carga (0, -1, +1...). No obstante, puesto que la migración (junto a otras propiedades), depende del estado de carga, queda claro que se necesita un conjunto completo de parámetros para cada estado de carga por separado.

Es por ello que en DADOS se ha seguido un modelizado alternativo, en vez de

| | | | | |
|---|---|-----------------|---|---|
| $I \begin{cases} I^+ \\ I^0 \\ I^- \end{cases}$ | $V \begin{cases} V^{++} \\ V^+ \\ V^0 \\ V^- \\ V^{--} \end{cases}$ | $As \{ As^+ \}$ | $As_i \begin{cases} As_i^+ \\ As_i^0 \end{cases}$ | $As_v \begin{cases} As_v^+ \\ As_v^0 \\ As_v^- \end{cases}$ |
| $B \{ B^- \}$ | $B_i \begin{cases} B_i^+ \\ B_i^0 \\ B_i^- \end{cases}$ | $C \{ C^0 \}$ | $C_i \{ C_i^0 \}$ | $In \{ In^- \}$ |
| $In_v \begin{cases} In_v^0 \\ In_v^- \end{cases}$ | $In_i \begin{cases} In_i^0 \\ In_i^- \end{cases}$ | $P \{ P^+ \}$ | $P_i \begin{cases} P_i^0 \\ P_i^+ \end{cases}$ | $P_v \begin{cases} P_v^+ \\ P_v^0 \\ P_v^- \end{cases}$ |

Tabla 8.23: Lista de especies cargadas de DADOS.

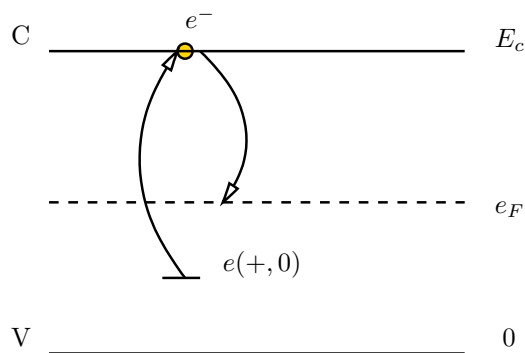
definir una etiqueta de carga para cada especie se han definido tantas partículas diferentes como especies podemos tener en diferentes estados de carga, de forma que para el programa I^+ es diferente de I^- como lo podría ser B de I . Así pues también se definen todas las interacciones una por una de acuerdo al tipo de partícula (I , V , B ...), pero también a su estado de carga, que en nuestro caso se especifica añadiendo la letra P (plus) para cargas positivas, M para negativas (minus), y PP o MM para doblemente positivas o negativas. Con esta notación DADOS llama BiM al B_i^- y $AsVP$ al As_v^+ . Las interacciones permitidas entre partículas, teniendo también en cuenta la carga, se encuentran en las tablas 8.4 y 8.5, y no pueden ser cambiadas por el usuario mediante parámetros externos, ya que se supone que las interacciones obedecen a modelos físicos que no tienen por qué ser ajustados.

▷ 8.8.1

Hipótesis

El modelo de carga implementado en DADOS parte de las siguientes hipótesis y suposiciones:

- Se toma la banda de valencia como origen de energía cero.
- Como las reacciones de carga son más rápidas que las estructurales (Rafferty, 1997), se supone que la carga es adaptada y actualizada de forma instantánea.
- La energía de formación de las especies neutras (como $E_f(I^0)$), y de aquellas que se forman a partir de especies neutras y dopantes siempre ionizados (como $B^- + I^0 \rightarrow B_i^-$), no depende del nivel de Fermi, y por tanto DADOS

Figura 8.33: Energía de formación de I^+ .

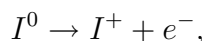
toma, como es natural, las energías para estas especies como parámetros y recalcula el resto a partir de ellas. Concretamente se usa esta hipótesis en la sección 8.8.2, para calcular las energías de formación de los defectos puntuales e impurezas en otros estados de carga.

- La dependencia de los valores de carga con la temperatura y el estrechamiento del gap son proporcionales a la anchura de banda. Con esta suposición se pueden establecer relaciones de proporcionalidad para calcular las posiciones en el gap a diferentes temperaturas y condiciones de dopado a partir de los parámetros iniciales para una única temperatura, como se muestra en la sección 8.8.3 para la dependencia con la temperatura, y en la 8.8.10 para los efectos relacionados con el estrechamiento del gap.
- Los átomos sustitucionales se encuentran siempre ionizados y en igual estado de carga. Aunque esta suposición no es estrictamente necesaria facilita bastante los cálculos, y es cierta a las temperaturas normales de simulación usadas en DADOS. De esta forma los dopantes sustitucionales se encuentran como As^+ , C^0 , In^- y P^+ .

▷ 8.8.2

Energías de formación para especies cargadas

Tomemos la formación de I^+ a modo de ejemplo. La reacción necesaria para generar este estado de carga es



donde vamos a denotar la energía necesaria para extraer un electrón de I^0 y obtener $I^+ + e^-$ como $e(+, 0)$, valor que medimos desde la banda de valencia (ver figura 8.33).

Así pues, necesitamos $E_c - e(+, 0)$ para ionizar I^0 y obtener I^+ , pero entonces tenemos un sistema con un electrón extra que, cuando sea devuelto, nos

| Símbolo | Nombre ddp | Valor |
|----------------------|------------|---------|
| $e(-, -)(V^{--})$ | e_VMMO | 1.06 eV |
| $e(0, -)(V^-)$ | e_VMO | 0.6 eV |
| $e(+, 0)(V^+)$ | e_VPO | 0.03 eV |
| $e(+++, +)(V^{+++})$ | e_VPP0 | 0.13 eV |
| $e(0, -)(I^-)$ | e_IMO | 1 eV |
| $e(+, 0)(I^+)$ | e_IP0 | 0.4 eV |
| $e(0, -)(B_i^-)$ | e_BiMO | 0.8 eV |
| $e(+, 0)(B_i^+)$ | e_BiPO | 1.04 eV |
| $e(+, 0)(As_i^+)$ | e_AsiPO | 0.1 eV |
| $e(0, -)(As_V^-)$ | e_AsVMO | 0.77 eV |
| $e(+, 0)(As_V^+)$ | e_AsVPO | 0.3 eV |
| $e(0, -)(In_i^-)$ | e_IniMO | 0.6 eV |
| $e(0, -)(In_V^-)$ | e_InVMO | 0.7 eV |
| $e(0, -)(P_V^-)$ | e_PhVMO | 0.7 eV |
| $e(+, 0)(P_V^+)$ | e_PhVPO | 0 eV |
| $e(+, 0)(P_i^+)$ | e_PhiPO | 0.6 eV |

Tabla 8.24: Parámetros para los niveles electrónicos a $T = 0$ K.

reportará la energía $E_c - e_F$, siendo e_F el nivel de Fermi. Por último tendremos que

$$E_f(I^+) = E_f(I^0) + E_c - e(+, 0) - E_c + E_f = E_f(I^0) + E_f - e(+, 0)$$

y consecuentemente con esta relación, la concentración entre I^+ e I^0 , e I^0 e I^- será:

$$\frac{[I^0]}{[I^+]} = \exp\left(\frac{e_F - e(+, 0)}{k_B T}\right)$$

$$\frac{[I^-]}{[I^0]} = \exp\left(\frac{e_F - e(0, -)}{k_B T}\right)$$
(8.23)

Los niveles electrónicos (para $T = 0$ K) son introducidos como parámetros de DADOS, y se encuentran listados en la tabla 8.24.

► 8.8.3

Dependencia con la temperatura

La anchura del gap con la temperatura viene dada por la siguiente expresión (Sze, 1981)

$$E_g(T) = E_{gT=0K} - \frac{A_g T^2}{B_g + T},$$
(8.24)

| Símbolo | Nombre ddp | Valor |
|-----------------|------------|--------------------------------------|
| $E_g(T = 0K)$ | Eg0 | 1.17 eV |
| A_g | Agap | 0.000473 eV/K |
| B_g | Bgap | 636 K |
| $N_c(T = 300K)$ | Nc300 | $3.2 \times 10^{19} \text{ cm}^{-3}$ |
| $N_v(T = 300K)$ | Nv300 | $1.8 \times 10^{19} \text{ cm}^{-3}$ |
| $\exp N_c$ | expNc | 1.5 |
| $\exp N_v$ | expNv | 1.5 |

Tabla 8.25: Parámetros relacionados con la dependencia de los niveles de carga con la temperatura.

y usando la hipótesis de que la dependencia de todos los niveles de carga con la temperatura es la misma que para la anchura del gap, podemos recalcular estos niveles en otras temperaturas con la siguiente proporción

$$e(+, 0)(T) = e(+, 0)_{T=300K} \times \frac{E_g(T)}{E_{gT=300K}}.$$

Las expresiones para la densidad de estados de las bandas de conducción y valencia siguen un patrón similar:

$$\begin{aligned} N_c(T) &= N_c(T = 300K) \times \left(\frac{T}{300}\right)^{\exp N_c}, \\ N_v(T) &= N_v(T = 300K) \times \left(\frac{T}{300}\right)^{\exp N_v}, \end{aligned} \quad (8.25)$$

y, por último, las densidades para los niveles intrínsecos y densidades intrínsecas serán:

$$\begin{aligned} e_i(T) &= \frac{E_g(T)}{2} + \frac{k_B T}{2} \times \ln\left(\frac{N_v}{N_c}\right), \\ n_i(T) &= \sqrt{N_c N_v} \times \exp\left(-\frac{E_g(T)}{2k_B T}\right). \end{aligned} \quad (8.26)$$

Los parámetros usados anteriormente se encuentran expuestos en la tabla 8.25.

▷ 8.8.4

Atracciones y repulsiones de carga

Podemos suponer que las atracciones y repulsiones electrostáticas afectan a entornos pequeños y grandes de forma separada, de tal forma que se han implementado las atracciones y repulsiones en distancias pequeñas prohibiendo las interacciones entre partículas con el mismo estado de carga no neutro (+, +) (-, -).

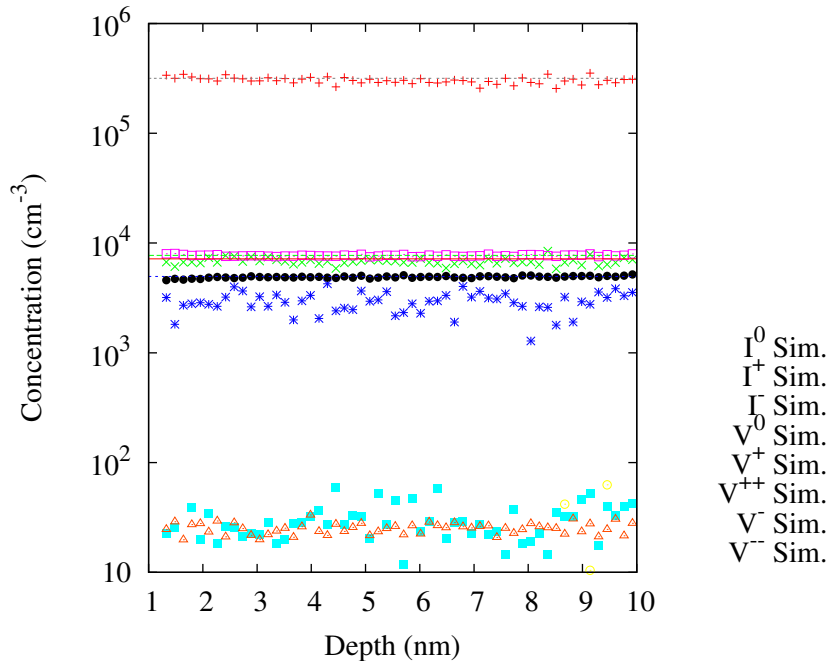


Figura 8.34: Equilibrio entre diferentes estados de carga de I_s y V_s a 600°C de temperatura.

Las interacciones electrostáticas a larga distancia quedan incluidas automáticamente debido al arrastre que hay en la migración producido por la dependencia de ésta con el nivel de Fermi.

► 8.8.5

Probabilidades y estados de carga

Más detalles en sec. A.6.1, pág. 337.

El modelo de carga de DADOS se preocupa básicamente por dos aspectos. Por una parte hay que mantener las concentraciones adecuadas entre los diferentes estados de carga de las mismas especies, concentraciones que dependen de la temperatura, la anchura del gap y del nivel de Fermi, y por otra parte cada estado cargado debe difundir con una difusividad adecuada, que es diferente según la carga.

Como ya hemos explicado, DADOS implementa cada estado de carga como una partícula diferente, por lo que mantener para cada estado la difusividad adecuada es un problema trivial, ya que DADOS moverá cada una de ellas de acuerdo a los parámetros que pongamos y, por tanto, de manera natural habrá difusividades distintas.

Así pues, el problema real del modelo de carga es mantener la proporción entre las concentraciones de los diferentes estados. Por ejemplo, la figura 8.34 muestra algunos perfiles de partículas cargadas en equilibrio a 600°C . En general, esta

proporción no es homogénea, puesto que depende del nivel de Fermi, que no tiene por qué ser el mismo en todo el cristal. Para mantener la concentración, DADOS tiene que calcular las proporciones relativas para cada estado de carga de cada especie, definiéndose la probabilidad de una partícula X de estar en un estado de carga j respecto a su estado neutro como

$$P_0^j = \frac{[X^j]}{[X^0]}, \quad j = 0, -, --, +, ++, \quad (8.27)$$

valor que es fácilmente calculable a partir de la ecuación 8.23.

Así, la probabilidad de que una partícula X se encuentre en un estado de carga j será:

$$P^j = \frac{P_0^j}{\sum_j P_0^j}. \quad (8.28)$$

Una vez que se conocen todas las probabilidades para cada estado de carga, hay dos mecanismos para lograr que se respeten dichas proporciones:

1. Actualizar el estado de carga de la partícula cada vez que salta: lo llamaremos actualización dinámica.
2. Repasar todas las partículas cada cierto tiempo y actualizar todas para que se encuentren en la proporción adecuada de carga: lo llamaremos actualización estática.

La actualización dinámica presenta dos problemas:

- Para el primero nos serviremos de un ejemplo. Si suponemos que A^+ tiene una frecuencia de salto muy grande y A^- una muy pequeña, tendremos que cada vez que salte A^+ se calculará la probabilidad de que pase a A^+ y, como esto se hace muy a menudo, más pronto o más tarde la partícula pasará a ser A^- . Como A^- apenas salta es mucho más probable que tengamos un nuevo A^+ saltando y transformándose en A^- , por lo que al final tendríamos una concentración artificialmente alta de la especie lenta. Por ello, si nos limitáramos a usar P_j como probabilidad de cambio para este caso dinámico el modelo funcionaría mal, y por tanto, nos vemos obligados a definir una probabilidad *dinámica* como

Más detalles
en sec. A.6.2,
pág. 338.

$$P_{\text{dyn}}^j = \frac{P^j \nu_m(X^j)}{\sum_i P^i \nu_m(X^i)}, \quad (8.29)$$

donde $\nu_m(X^i)$ es la frecuencia de salto (responsable de la migración) para la partícula X en el estado de carga i .

- El segundo problema se relaciona con las partículas que presentan una muy baja (o nula) frecuencia de salto. Puesto que la actualización dinámica sólo

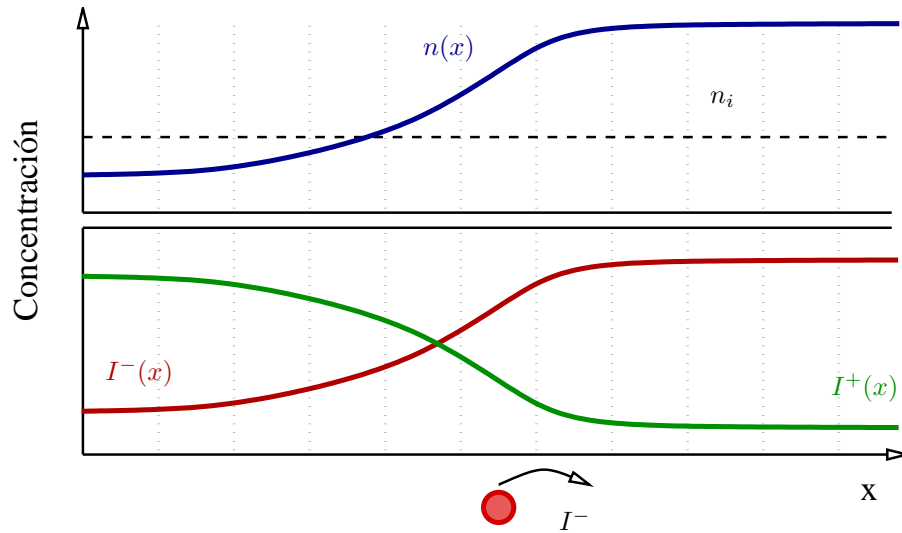


Figura 8.35: La concentración de la especies cargadas varía con la concentración electrónica.

tiene lugar cuando una partícula salta, tendríamos que aquellas que salten poco (o nunca) no estarán bien actualizadas. La forma de resolver este problema es actualizar todas las partículas de tanto en tanto, es decir, la actualización estática de la que ya hemos hablado antes. Dado que este otro tipo de actualización requiere muchos recursos (tiempo de cálculo), es necesario calcular muy cuidadosamente cuándo hay que realizar una de estas actualizaciones globales, tema que se tratará a continuación.

Criterios para la actualización estática

El criterio para actualizar todos los estados de carga de los defectos y calcular los nuevos valores de η resulta ser un compromiso entre la velocidad y la precisión de la simulación. DADOS lo modeliza a través de un contador que intenta calcular cómo de actualizada está la simulación.

Cada vez que aparece o desaparece un dopante eléctricamente activo de la simulación, (por ejemplo, en la reacción de ruptura $B_i \rightarrow B + I$ aparece B), se incrementa el contador, con lo que éste tiene una estimación del número de cambios de carga⁵. Cuando el número del contador es mayor que un cierto $x_{\text{staticupdate}}$, el programa actualiza la carga de todas las partículas. No obstante, el número $x_{\text{staticupdate}}$ no permanece fijo a lo largo de toda la simulación, sino que es incrementado o decrementado dependiendo de la variación relativa de la carga

⁵El contador siempre se incrementa, tanto al aparecer como al desaparecer partículas eléctricamente activas, ya que para una desactivación y reactivación no es fácil saber si fue la misma partícula o dos partículas distintas.

| Símbolo | Nombre ddp | Valor |
|-----------------|------------------|-------|
| ChargeVariation | ChargeVarPercent | 0.25 |

Tabla 8.26: Parámetros usados en DADOS para actualizar la carga de manera estática.

entre una actualización y la siguiente. El usuario, gracias al parámetro externo `ChargeVariation`, que se muestra en la tabla 8.26, puede establecer el valor máximo deseado de variación entre una actualización y la siguiente, de tal forma que cuando la variación relativa es menor que el parámetro, se incrementa $x_{\text{staticupdate}}$ haciendo al sistema menos sensible a las variaciones de carga, y en cambio cuando la variación es mayor que la deseada se aumentaría la “sensibilidad”, disminuyendo el valor de $x_{\text{staticupdate}}$.

▷ 8.8.6

Deriva eléctrica

Introducción

La inclusión de la carga en el simulador kMC DADOS supone, de manera natural, la necesidad de introducir un campo eléctrico relacionado con las variaciones locales de la carga, y por tanto la existencia de fuerzas que actúen sobre las especies cargadas, que generarán una deriva en la migración de las especies que se encuentren sometidas a dichas fuerzas.

Una forma de introducir esta deriva en el modelo es modificando las probabilidades de salto, de tal forma que éstas tomen en cuenta que las partículas cargadas bajo una fuerza eléctrica no saltarán de forma isotrópica en el espacio.

Concretamente, una partícula en el interior de un campo eléctrico podrá saltar en ambas direcciones, pero es más probable que lo haga en aquella que sea más favorable al campo eléctrico, por lo que, si P^{\rightarrow} y P^{\leftarrow} son las probabilidades de que salte en sentido positivo o negativo en el eje x tendremos que

$$\frac{P^{\rightarrow}}{P^{\leftarrow}} = \exp\left(\frac{qE_x\lambda}{k_B T}\right),$$

y cuando la partícula se encuentre en un estado j

$$\frac{P^{\rightarrow}}{P^{\leftarrow}} = \exp\left(\frac{jqE_x\lambda}{k_B T}\right) = \exp\left(\frac{QE_x\lambda}{k_B T}\right), \quad Q = 0, \pm q, \pm 2q, \dots \quad (8.30)$$

Se puede derivar de la ecuación 8.30 que

$$v_x = \lambda(P^{\rightarrow} - P^{\leftarrow})\nu_m/6,$$

lo que, en una aproximación de primer orden, nos da la conocida relación

$$v_x = QD\mathcal{E}_x/(k_B T).$$

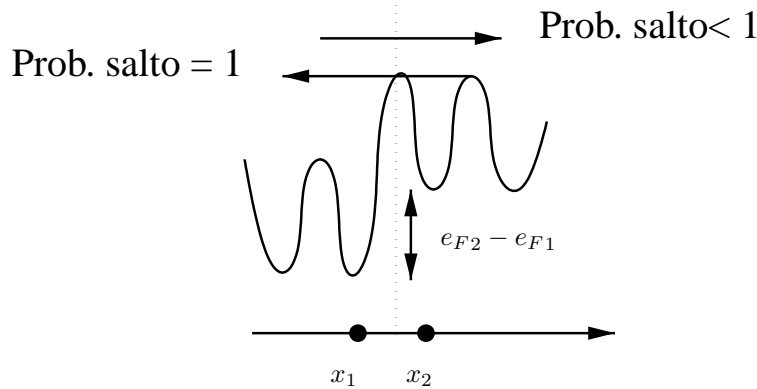


Figura 8.36: Barreras de migración para un I^+ .

Para implementar la diferente probabilidad de salto según la dirección se han usado barreras de migración.

Barreras de migración

Las reacciones de las partículas cargadas al campo eléctrico han sido implementadas como *barreras de migración*, de tal forma que cuando una partícula en un determinado estado de carga intenta migrar a una región en la que es más probable que exista en otro estado diferente, habrá una barrera a dicha migración. Así, pues las barreras estarán relacionadas con las concentraciones relativas de cada especie, y por tanto con P_0^j .

Imaginemos ahora una partícula I^+ que se mueve desde x_1 hasta x_2 . Cuando $P_0^+(x_2) > P_0^+(x_1)$ el salto no ofrece barrera alguna, pero en caso contrario habrá una probabilidad $1 - [P_0^+(x_2)]/[P_0^-(x_1)]$ de que se rechace el salto. Esto indica que la probabilidad de que el I^+ pueda saltar será

$$\frac{P_0^+(x_2)}{P_0^-(x_1)} = \exp \left\{ \frac{[e_F - e(+, 0)]_2 - [e_F - e(+, 0)]_1}{k_B T} \right\}, \quad (8.31)$$

y cuando $e(+, 0)_2 = e(+, 0)_1$

$$\frac{P_0^+(x_2)}{P_0^-(x_1)} = \exp \left[\frac{(e_F)_2 - (e_F)_1}{k_B T} \right]. \quad (8.32)$$

La figura 8.36 muestra un diagrama de energía para el proceso aquí expuesto.

► 8.8.7

Implementación

La implementación del modelo de carga expuesto más arriba ha tenido en cuenta, además de todos los aspectos físicos que hemos ido comentando, la eficacia

computacional del modelo. Concretamente la velocidad con la que se actualiza la carga es bastante crítica puesto que si no se presta atención a dicho aspecto se puede ralentizar una simulación hasta un 1000 %, haciendo que la inclusión del modelo de carga suponga un incremento de 10 veces el tiempo de simulación, cuando ahora mismo una simulación con carga tarda, como mucho, el doble que una sin carga.

Otro aspecto a tener en cuenta es que, debido a que internamente DADOS divide artificialmente el espacio en cajas, hay que hacer promedios a la carga en cada caja para evitar fenómenos de “todo o nada” (si una partícula cae en una caja muy pequeña tendría una concentración de carga altísima, y las de al lado ninguna), pero sin llegar a perder los fenómenos locales en los promedios. Cuando estos promedios no se hacen bien aparecen efectos raros en las regiones con variaciones bruscas de la carga.

Por último hay que dictaminar cómo de frecuentemente se desea actualizar la carga. Hacer pocas actualizaciones lleva a simulaciones rápidas, pero poco precisas puesto que la carga no sigue bien la evolución de la simulación, mientras que hacer muchas actualizaciones hará la simulación más precisa, pero ralentizará ésta hasta límites no tolerables. Así pues, como ya se ha comentado, es necesario un compromiso entre velocidad y precisión.

Cálculo y suavizado de la distribución electrónica

Para calcular la distribución electrónica el modelo de carga de DADOS itera sobre todas las cajas de la simulación y almacena el número de cargas (dopantes sustitucionales, que se consideran siempre activos), junto con el número de I s y V s por caja (que serán usadas como *carga de compensación*). Después de conseguir esta información comienzan los promedios: DADOS recolecta para cada caja quienes son sus cajas vecinas hasta un radio de promedio que es tomado como el radio de Debye

$$r_D = \sqrt{\frac{T}{4\pi e^2 N}}$$

Dicho radio depende de la concentración de dopantes N , que se estima a partir de la información previa de las cajas y sus vecinas. Una vez que ya se tiene el radio de promedio para cada caja, y que han sido descartadas aquellas que, o bien estaban vacías, o bien no presentaban cargas, DADOS calcula la contribución de cada caja promediando con las vecinas y compartiendo así la carga con ellas. Por último, en este cálculo de la concentración de carga N , se aplica una compensación con el número de intersticiales y vacantes de cada caja⁶, restando el número de estas partículas de la carga. Esta es una aproximación, pero que tiene en cuenta el hecho de que los I s y V s intentarán neutralizar a los dopantes adoptando el estado opuesto.

⁶Para simular que las regiones dañadas presentan alta resistividad.

Cálculo del nivel de Fermi

Una vez que se tiene la carga promediada, es necesario calcular las cantidades η relacionadas con el nivel de Fermi en cada caja:

$$\eta_F = \frac{e_F - e_i}{k_B T}.$$

Para ello se calcula primero la concentración de portadores mayoritarios de acuerdo a las siguientes ecuaciones

$$\begin{aligned} n_n &= \frac{1}{2}N + \sqrt{\frac{1}{4}N^2 + n_i^2}, & N \geq 0 \\ p_p &= -\frac{1}{2}N + \sqrt{\frac{1}{4}N^2 + n_i^2}, & N < 0 \end{aligned}, \quad (8.33)$$

y después se utilizan estos valores para poder calcular η_F :

$$\eta_F = \eta_c + \mathcal{F}_{1/2}^{-1} \left(\frac{n_n}{N_c} \right) \quad (8.34)$$

o

$$\eta_F = \eta_v - \mathcal{F}_{1/2}^{-1} \left(\frac{p_p}{N_v} \right), \quad (8.35)$$

siendo $\mathcal{F}_{1/2}^{-1}$ la función de Fermi-Dirac inversa, que es

$$\mathcal{F}_{1/2}(\eta) = \int_{-\infty}^{\infty} \frac{\eta^{1/2}}{1 + e^{\xi - \eta}} d\xi.$$

▷ 8.8.8

Difusividades efectivas de dopantes

El modelo de carga multiplica las posibilidades de difusión de los dopantes. Por ejemplo, para el caso del boro, que difunde a través del boro intersticial o B_i hay tres posibilidades: B_i^+ , B_i^0 y B_i^- , siendo diferentes las difusividades microscópicas de cada uno de estos estados, que llevan a una difusividad efectiva que es la combinación de todas ellas y depende del nivel de Fermi.

En las secciones siguientes vamos a explicar estos mecanismos, usando el boro como ejemplo, y vamos a ver cómo calcular la difusión efectiva teniendo en cuenta todas las migraciones de cada estado de carga parcial.

A su vez, en la tabla 8.27 se listan todas las energías de migración y prefactores para cada diferente estado de carga. Los valores para los estados neutrales de impurezas fueron ya mostrados en la tabla 8.16 y para los defectos puntuales I y V en 8.3.

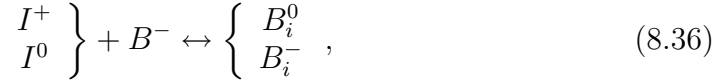
| Símbolo | Nombre ddp | Valor |
|-------------------|------------|--|
| $D_{m,0}(I^+)$ | Dm_IP | $0.005 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(I^+)$ | Em_IP | 1 eV |
| $D_{m,0}(I^-)$ | Dm_IM | $0.005 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(I^-)$ | Em_IM | 1 eV |
| $D_{m,0}(V^+)$ | Dm_VP | $0.001 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(V^+)$ | Em_VP | 0.6 eV |
| $D_{m,0}(V^{++})$ | Dm_VPP | $0.001 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(V^{++})$ | Em_VPP | 0.8 eV |
| $D_{m,0}(V^-)$ | Dm_VM | $0.001 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(V^-)$ | Em_VM | 0.4 eV |
| $D_{m,0}(V^{--})$ | Dm_VMM | $0.001 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(V^{--})$ | Em_VMM | 0.4 eV |
| $D_{m,0}(B_i^+)$ | Dm_BiP | $0.001 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(B_i^+)$ | Em_BiP | 1.1 eV |
| $D_{m,0}(B_i^-)$ | Dm_BiM | $0.001 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(B_i^-)$ | Em_BiM | 0.5 eV |
| $D_{m,0}(In_V^-)$ | Dm_InVM | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(In_V^-)$ | Em_InVM | 5 eV |
| $D_{m,0}(In_i^-)$ | Dm_IniM | $2 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(In_i^-)$ | Em_IniM | 2.2 eV |
| $D_{m,0}(As_V^+)$ | Dm_AsVP | $0.00021 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(As_V^+)$ | Em_AsVP | 1.3 eV |
| $D_{m,0}(As_V^-)$ | Dm_AsVM | $0.00021 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(As_V^-)$ | Em_AsVM | 3.4 eV |
| $D_{m,0}(As_i^+)$ | Dm_AsiP | $0.004 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(As_i^+)$ | Em_AsiP | 1.8 eV |
| $D_{m,0}(P_V^+)$ | Dm_PhVP | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(P_V^+)$ | Em_PhVP | 5 eV |
| $D_{m,0}(P_V^-)$ | Dm_PhVM | $0 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(P_V^-)$ | Em_PhVM | 5 eV |
| $D_{m,0}(P_i^+)$ | Dm_PhiP | $0.01 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_m(P_i^+)$ | Em_PhiP | 2.1 eV |

Tabla 8.27: Parámetros de DADOS (prefactores y energías de migración) para diferentes estados de carga.

La carga en reacciones de formación y ruptura

Las reacciones de formación y ruptura no deben modificar las concentraciones relativas de los distintos estados de carga. Para ello, y en el caso del boro, tomemos en consideración ambos procesos.

Para el caso de la reacción de emparejado



donde definimos la probabilidad de formación como P_f^q , siendo ésta la probabilidad de formación de B_i^q después de la reacción de emparejado de I con B . Esta probabilidad será

$$P_f^q = \frac{P_0^q \nu_{\text{break}}^q}{\sum_k P_0^k \nu_{\text{break}}^k}. \quad (8.37)$$

Como se muestra en el diagrama 8.36, no aparece B_i^+ en las reacciones de ruptura y, por tanto, siendo consecuentes con la recién escrita ecuación 8.37, la probabilidad de formación de esta especie en dicho estado de carga será 0.

Respecto a la reacción de ruptura, la probabilidad de tener I^q después de la ruptura de un B_i será

$$P_{\text{emis}}^q = \frac{P_d^q \delta_q}{\sum_k P_d^k \delta_k}, \quad (8.38)$$

donde δ_q es 1 cuando la reacción $I^q + B^-$ esté permitida, y 0 en caso contrario.

Un ejemplo: La difusión del boro

Más detalles
en sec. A.7,
pág. 339.

Como ya se comentó vamos a suponer para las reacciones del boro que:

- Los estados de carga están en equilibrio.
- No se permiten reacciones entre cargas que se repelan.
- Se conserva la reversibilidad microscópica.
- Las reacciones de ruptura conservan la carga.

Podemos empezar dándonos cuenta de que las reacciones involucradas serán:



y de estas expresiones podemos deducir que la energía de enlace será:

$$E_b(B_i^0) = E_b(B_i^-) + e_{B_i^-} - e_{I^+},$$

y que los prefactores deberán cumplir

$$\frac{D_{0,break}(B_i^-)}{D_{0,m}(I_0)} = \frac{D_{0,break}(B^0)}{D_{0,m}(I^+)} = \frac{D_{0,break}(B^+)}{D_{0,m}(I^{++})}.$$

Por último, la difusividad del boro será

$$D_B[B] = h(D_{B_i^-}[B_i^-] + D_{B_i^0}[B_i^0] + D_{B_i^+}[B_i^+]),$$

donde h es una constante de proporcionalidad con un valor igual a (página 316 de (Fahey et al., 1989))

$$h = 1 + \frac{[B^-]}{2n_i} \left[\left(\frac{[B^-]}{2n_i} \right)^2 + 1 \right]^{-1/2}.$$

Al suponer que $[B] \approx [B^-]$, y por tanto que $D_B \equiv D_{B^-}$ tendremos

$$D_B = D_{B_i^-} \frac{[B_i^-]}{[B^-]} + D_{B_i^0} \frac{[B_i^0]}{[B^-]} + D_{B_i^+} \frac{[B_i^+]}{[B^-]}.$$

Y comparando nuestras expresiones con la notación habitual de simuladores de continuo⁷

$$D_B = D_{xB} + D_{PB} \frac{p}{n_i} + D_{PPB} \left(\frac{p}{n_i} \right)^2$$

encontramos que:

$$\begin{aligned} D_{xB} &= D_{B_i^-} \frac{[B_i^-]}{[B^-]} \\ D_{PB} &= D_{B_i^0} \frac{[B_i^-]}{[B^-]} \left(\frac{[B_i^0]}{[B_i^-]} \right)_{int} \\ D_{PPB} &= D_{B_i^+} \frac{[B_i^-]}{[B^-]} \left(\frac{[B_i^+]}{[B_i^-]} \right)_{int}. \end{aligned}$$

Las energías de migración y formación para los distintos estados de carga del boro están representadas en la figura 8.37.

Boro, carga y saltos largos

El propósito de este apartado es estudiar la distancia de saltos largos, Λ , representada esquemáticamente en la figura 8.38, para la difusión del boro.

Se puede demostrar que la distancia de estos saltos es:

$$\Lambda = \lambda \sqrt{\frac{1}{36\sqrt{2}} \times \frac{\nu_m}{\nu_{break}}} = \lambda \sqrt{\frac{1}{6} \times \frac{D_B}{(D_I)}}. \quad (8.40)$$

Más detalles
en sec. A.7.1,
pág. 342.

⁷Como TaurusTM.

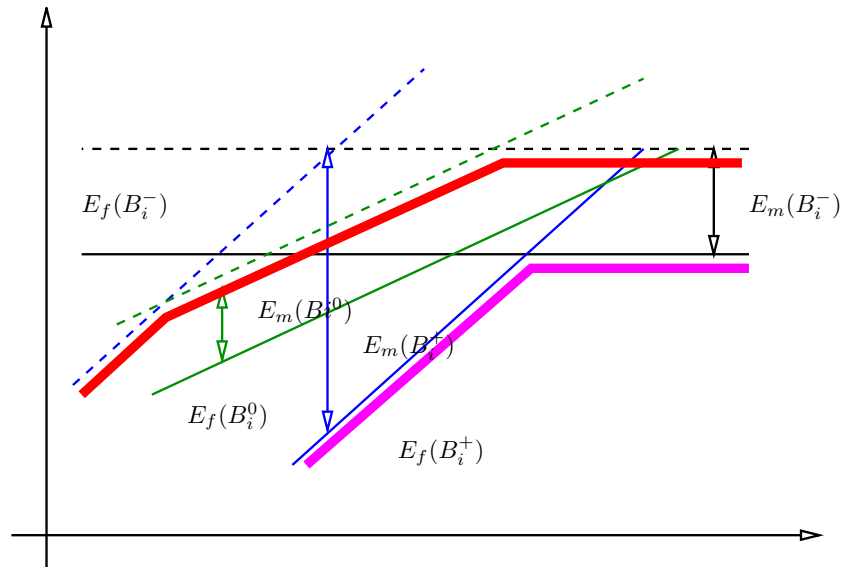


Figura 8.37: Energías de formación, migración y activación del boro dependiendo del nivel de Fermi.

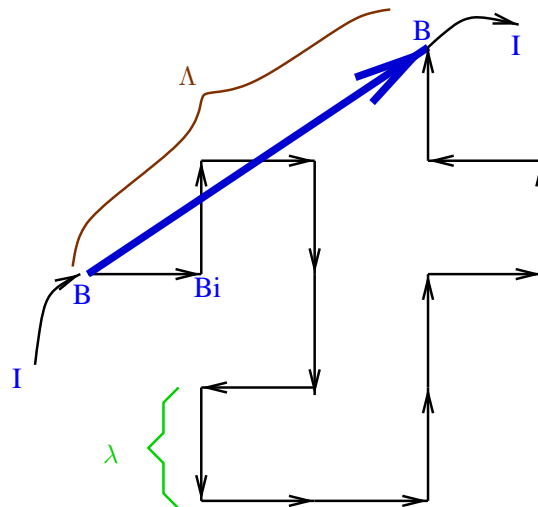


Figura 8.38: Relación entre los saltos de difusión (λ) y los saltos largos (Λ)

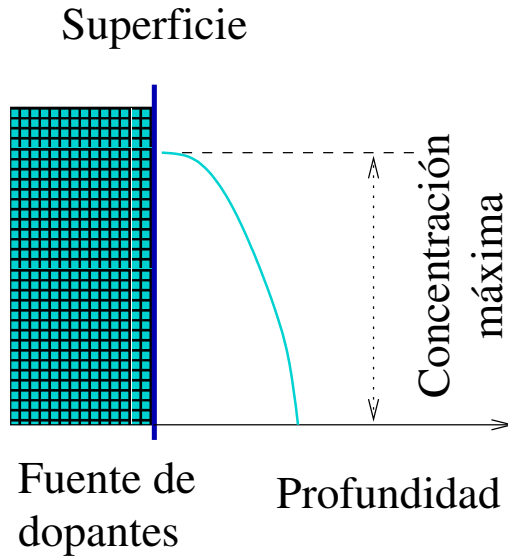


Figura 8.39: Forma habitual de medir la solubilidad (extrínseca) de un dopante: $[A]_{\max}^{\text{extrinsic}}$.

▷ 8.8.9

Dependencia de la solubilidad con el nivel de Fermi

La solubilidad de las especies cargadas dependen del nivel de Fermi, de acuerdo con las referencias (Northrup and Zhang, 1993; de Walle et al., 1993; Sadigh et al., 2002). Concretamente, la solubilidad de un dopante A (al que suponemos con carga negativa) es dada por:

$$[A^-]_{\max}(T) = [A^-]_{\max}^{\text{intrinsic}}(T) \times \exp\left(\frac{e_F - e_i}{k_B T}\right), \quad (8.41)$$

donde

$$[A]_{\max}^{\text{intrinsic}} = \frac{([A]_{\max}^{\text{extrinsic}})^2}{n_i}$$

es la solubilidad de la impureza en condiciones intrínsecas ($e_F = e_i$) y $[A]_{\max}^{\text{extrinsic}}$ lo es para condiciones extrínsecas, cuando $[A]_{\max} \gg n_i$. La figura 8.39 muestra la forma habitual de medir $[A]_{\max}^{\text{extrinsic}}$.

La solubilidad en DADOS está relacionada con los procesos de emisión y recombinación de B_i^0 en la superficie (como ya se ha comentado en la sección 8.7.2). Por ello, el prefactor necesario para la ecuación 8.20 se calcula usando la solubilidad intrínseca. La probabilidad de captura (ecuación 8.22) se calcula usando $D^{\text{eff}}(B)$ para condiciones intrínsecas. Como la concentración de B_i depende del nivel de Fermi, pero las probabilidades de captura y emisión no, la solubilidad del boro tendrá una dependencia natural con el nivel de Fermi, y las condiciones extrínsecas funcionarán de forma automática.

| Símbolo | Nombre ddp | Valor |
|-------------|------------|----------|
| $A_{cn1/4}$ | Acn1_4 | 0 |
| $A_{cn1/3}$ | Acn1_3 | -0.01484 |
| $A_{cn1/2}$ | Acn1_2 | 0.00078 |
| $A_{cp1/4}$ | Acp1_4 | -0.01627 |
| $A_{cp1/3}$ | Acp1_3 | 0 |
| $A_{cp1/2}$ | Acp1_2 | -0.00018 |
| $A_{vn1/4}$ | Avn1_4 | 0.01508 |
| $A_{vn1/3}$ | Avn1_3 | 0 |
| $A_{vn1/2}$ | Avn1_2 | 0.00074 |
| $A_{vp1/4}$ | Avp1_4 | 0 |
| $A_{vp1/3}$ | Avp1_3 | 0.01846 |
| $A_{vp1/2}$ | Avp1_2 | -0.00263 |

Tabla 8.28: Parámetros de DADOS usados para el modelizado del estrechamiento del gap, que dependen de la concentración de dopantes.

► 8.8.10

Estrechamiento del gap

El modelo de carga de DADOS incluye los desplazamientos del mínimo de la banda de conducción y máximo de la banda de valencia inducidos por los dopantes a partir de la formulación de (Jain and Roulston, 1991)

$$\begin{aligned}\Delta\eta_c &= A_{cn1/4} \left(\frac{N^+}{10^{18}}\right)^{1/4} + A_{cn1/3} \left(\frac{N^+}{10^{18}}\right)^{1/3} + A_{cn1/2} \left(\frac{N^+}{10^{18}}\right)^{1/2} \\ \Delta\eta_v &= A_{vn1/4} \left(\frac{N^+}{10^{18}}\right)^{1/4} + A_{vn1/3} \left(\frac{N^+}{10^{18}}\right)^{1/3} + A_{vn1/2} \left(\frac{N^+}{10^{18}}\right)^{1/2},\end{aligned}\quad (8.42)$$

para semiconductores de tipo n y

$$\begin{aligned}\Delta\eta_c &= A_{cp1/4} \left(\frac{N^-}{10^{18}}\right)^{1/4} + A_{cp1/3} \left(\frac{N^-}{10^{18}}\right)^{1/3} + A_{cp1/2} \left(\frac{N^-}{10^{18}}\right)^{1/2} \\ \Delta\eta_v &= A_{vp1/4} \left(\frac{N^-}{10^{18}}\right)^{1/4} + A_{vp1/3} \left(\frac{N^-}{10^{18}}\right)^{1/3} + A_{vp1/2} \left(\frac{N^-}{10^{18}}\right)^{1/2},\end{aligned}\quad (8.43)$$

para los de tipo p . $\Delta\eta_v$ y $\Delta\eta_c$ son los incrementos (o disminuciones) de las bandas de conducción y valencia tal y como se representan en la figura 8.40. Por último, $A_{cp1/3}, \dots$ son los parámetros de éste modelo, conseguidos de la referencia (Persson et al., 1999), y listados en la tabla 8.28. El estrechamiento depende de la concentración de donores y aceptores, que son cantidades bien establecidas por el programa, de tal forma que en cada actualización estática de carga se aplican los adecuados estrechamientos de carga locales de acuerdo a las fórmulas anteriores.

Para el resto de niveles electrónicos que se asientan entre la banda de valencia y la de conducción, se asume una variación proporcional con el estrechamiento

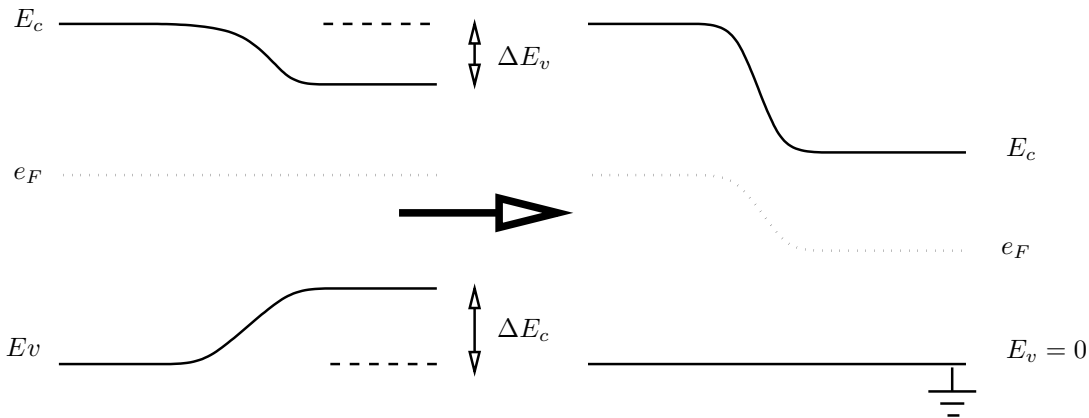


Figura 8.40: La anchura del gap depende de la concentración local de electrones.

del gap, de tal manera que en las regiones en las que η_c y η_v sean corregidas por las cantidades $\Delta\eta_c$ y $\Delta\eta_v$, una cantidad genérica η_x sería corregida de acuerdo a la siguiente expresión:

$$\Delta\eta_x = \frac{\eta_c - \eta_x}{\eta_c - \eta_v} \Delta\eta_v + \frac{\eta_x - \eta_v}{\eta_c - \eta_v} \Delta\eta_c. \quad (8.44)$$

▷ 8.9

Comandos de DADOS

Las simulaciones kMC de DADOS se ejecutan a través de un conjunto reducido de comandos: se introducen partículas en la celda de simulación a través de la lectura de perfiles o la implantación (para la que se llama a un programa externo), y se realiza el recocido con el comando *anneal*. Estos comandos hacen ciertas suposiciones que es necesario conocer para poder usar correctamente el programa, y que vamos a exponer a continuación.

▷ 8.9.1

Insertando partículas en la simulación

Los comandos para insertar partículas en la simulación son el comando de implantación *implant* y de lectura de un perfil *read profile*. Ambos introducen partículas (*Is*, *Vs* o dopantes) en coordenadas concretas. Puesto que la aparición repentina de una partícula en medio de la celda de simulación no tiene sentido físico, se permite que la nueva partícula interactúe con sus vecinos y forme clusters, pockets amorfos, etc., según las partículas que encuentre. Esta interacción

inicial sucede al insertar la partícula, y no necesita tiempo de recocido. Al insertar una partícula se informa al modelo de amorfización de su presencia para que actualice los niveles de dañado. Una vez que el modelo de amorfización conoce la presencia de la partícula es libre de borrarla y regenerarla como partícula oculta, como ya ha sido comentado.

▷ 8.9.2

Lectura de perfiles

UVAS-DADOS puede leer ficheros con perfiles externos para el eje x , que son ficheros de texto con dos columnas, la primera para la profundidad en nm en el eje x y la segunda para la concentración en cm^{-3} . Cada fichero contiene un perfil para una única especie que, al ser leído, genera una distribución aleatoria, pero homogénea, de partículas en el plano yz que va siguiendo la concentración adecuada en cada x .

La lectura de perfiles requiere de muchas situaciones y casos particulares que hay que tener en cuenta. A veces el perfil puede provenir de un perfil implantado durante un cierto tiempo, o bien de alguna concentración preexistente en la muestra de silicio que se introdujo por otros métodos (capas enterradas...). Por ello hay que pasar información a DADOS sobre qué perfiles hay que insertar simultáneamente y en cuánto tiempo. Una vez que el programa empieza a insertar las partículas del perfil lo hace de manera aleatoria, sin seguir un orden artificial en x , hasta que logra la concentración adecuada en cada punto. Esto se consigue limitándose a almacenar temporalmente las posiciones en las que habría que insertar las partículas, pero sin insertarlas aún y, una vez que sabe todas las que tiene que introducir (que pueden también ser de perfiles distintos que ha ido almacenando) aleatoriamente, y teniendo en cuenta el tiempo en el que se pide que las inserte, las va metiendo en la celda de simulación y borrándolas del almacén temporal donde estaban.

Este proceso de lectura de perfiles se realiza mediante el uso de dos comandos distintos: uno que lee los perfiles y va almacenando las partículas que hay que generar, pero sin introducirlas aún, y otro que toma todo el conjunto de partículas calculadas por el primer comando y las inserta en la simulación.

A modo de ejemplo: Si quisiéramos leer un perfil inicial de boro y posteriormente leer los perfiles resultantes de una implantación, haríamos algo como:

1. Con el comando *read profile* leer el perfil de boro y generar las partículas sin insertarlas.
2. Con *implant profile* insertar todas las partículas generadas (de momento sólo el boro) en un tiempo 0.
3. Con *read profile* leer el perfil de I_s .

4. Con *read profile* leer el perfil de V_s .
5. Con *implant profile* insertar las partículas almacenadas temporalmente (que ahora son los I_s y V_s) en la simulación, en un tiempo t igual al que duró la implantación que genero sus perfiles.

Hay que recordar que según se insertan átomos se les permite interaccionar con los vecinos, como se comentó en 8.9.1.

▷ 8.9.3

Implantación iónica

DADOS está diseñado para poder unirse a simuladores externos de implantación iónica, que pasen las cascadas con las coordenadas de las partículas implantadas y los defectos puntuales. En este sentido UVAS-DADOS está unido con MARLOWE, lo que permite disponer de un comando de implantación que requiere los siguientes parámetros:

implanted species , nombre de la especie a implantar (en inglés, boron, silicon, arsenic...)

energy energía del ion implantando, en KeV.

dose dosis a implantar, en cm^{-2} .

dose rate velocidad de implantación, para poder calcular el tiempo que dura la implantación como $\text{dose}/\text{dose rate}$.

window Y,Z Dimensiones de la ventana de implantación. Cuando ésta es más grande que la celda de simulación DADOS la recorta al mismo tamaño que dicha celda.

temperature Temperatura del sustrato de la oblea. Esta temperatura no es un parámetro que se pase a MARLOWE, sino que es usado por DADOS para hacer un recocido que dure un tiempo t a la temperatura pedida. Concretamente, se hace un recocido entre cascada y cascada, de tal forma que para n cascadas el tiempo t entre cada cascada será $\text{dose}/(n \times \text{dose rate})$.

oxide thickness Profundidad de la capa de SiO_2 . Generalmente la capa de óxido nativo es de unos 1.5 nm, que es la cantidad que reserva DADOS entre $x = 0$ y $x = 1,5$ nm para crear una intercara $\text{Si} - \text{SiO}_2$ (como se cuenta en la sección 8.1.3). El cambio de este parámetro afecta sólo a MARLOWE, DADOS seguirá considerando 1.5 nm de óxido independientemente de lo que se ponga aquí.

| Símbolo | Nombre ddp | Valor |
|--------------------------|------------|-------|
| ΔT_{base} | dTBase | 2 |
| n_{init} | nInit | 1 |
| ΔT_{init} | dTLimit | 20 °C |

Tabla 8.29: Parámetros de DADOS utilizados para la generación de las rampas de temperatura.

amorphous implant MARLOWE permite implantar en cristales de silicio cristalino o amorfo, seleccionándose uno u otro con el uso de este parámetro. Al igual que *oxide thickness*, este parámetro solo tiene sentido para el implantador MARLOWE, puesto que DADOS considerará el silicio cristalino o amorfo según la concentración de dañado, y no tendrá en cuenta para nada lo que hayamos puesto aquí.

▷ 8.9.4

Anneal (recocido o difusión)

El comando *anneal* es el más importante de DADOS. Anneal se limita escoger un evento dentro de todos los posibles, ejecutarlo, calcular cuanto tiempo simulado pasó, y repetir el proceso hasta que el tiempo simulado total sea igual al tiempo pedido por el usuario. Las opciones de anneal son:

Initial temperature Temperatura inicial con la que comienza la simulación.

Final temperature Temperatura final a la que acaba.

Time Tiempo total de recocido.

Cuando la temperatura inicial y final son distintas, DADOS simula una rampa de temperatura, haciendo recocidos a distintas temperaturas y distintos tiempos hasta completar el total. DADOS intenta regular la rampa de forma logarítmica, de tal manera que a bajas temperaturas haya pocos recocidos pero durante tiempos largos, y a altas haya muchos pequeños recocidos (cambiar la temperatura supone bastante tiempo de CPU). De esta manera la rampa adopta la forma de escalera con escalones de distintos tamaños, como se observa en la figura 8.41.

Para regular la forma en la que se genera la rampa de temperatura se pueden variar los parámetros expuestos en la tabla 8.29. Los intervalos para la temperatura más alta son

$$(\Delta T_{\text{base}})^{n_{\text{init}}+n},$$

hasta su valor máximo, establecido en ΔT_{limit} .

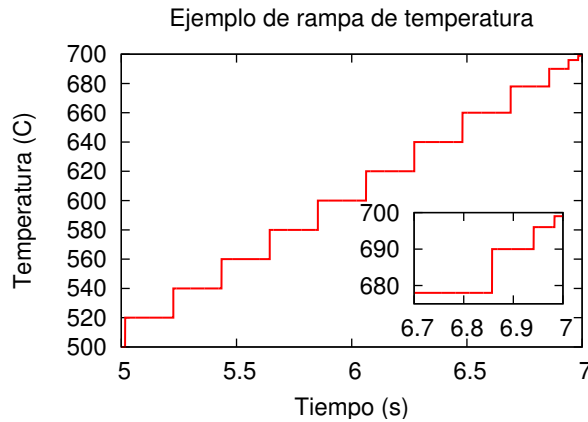


Figura 8.41: Modelo de DADOS para las rampas de temperatura: Pocos pasos largos a temperaturas bajas y muchos cortos a temperaturas altas.

▷ 8.9.5

Generación de salidas

Una vez realizada una simulación es imprescindible obtener la máxima cantidad de información posible de ella. En DADOS esta información se almacena en un conjunto denominado “snapshots”. Cada uno de estos snapshot es una imagen de la simulación tomada en un tiempo concreto, de tal forma que se van almacenando estas tomas en la memoria, conteniendo la siguiente información

- información atomística en 3D: Tipo de partícula (B , B_i , I , ...), nombre del defecto al que pertenece ($\{311\}$, cluster de impurezas, defecto puntual...) y posición tridimensional en nanómetros.
- información de perfiles en 1D: Concentración de cada especie en cada estado de carga y tipo de defecto respecto al eje x .
- Evolución temporal de la supersaturación de intersticiales y vacantes con el tiempo.
- Concentración electrónica en el eje x .
- Campo eléctrico tridimensional en el eje x .
- Diversos histogramas:
 - Histograma de clusters de I_s : Concentración de clusters de I_s presentes en la simulación según el tamaño de cada cluster. Se listan todos los clusters de I_s , es decir, clusters pequeños, $\{311\}$ y bucles de dislocación, en el mismo histograma.

Más detalles
en sec. A.3.4,
pág. 334.

- Histograma de clusters de V s: Concentración de clusters de V s según su tamaño, se incluyen en el mismo histograma tanto los voids como los clusters pequeños de V s.
- Histogramas de pockets amorfos: Concentración de los pockets amorfos frente al número total de partículas que contienen y número de pares IV almacenados.
- Histogramas de clusters de impurezas: Número de clusters de impurezas existentes en la simulación según el número de impurezas e I s (o V s) contenidos, es decir, para el cluster $B_n I_m$ se lista el número que hay para cada n y m . Existe un histograma de este tipo por cada impureza.

El número de imágenes tomadas es regulable por el usuario, siendo bastante importante esta cantidad, puesto que con unos pocos snapshots puede perderse información relevante, pero tomando muchos se incrementa sustancialmente la memoria requerida por la simulación.

Para regular el número de snapshots tomados, DADOS admite los siguientes criterios:

1. Generar uno cada cierto número de eventos (independientemente del tiempo simulado transcurrido).
2. Generar x snapshots por década de tiempo simulado (independientemente del número de eventos simulados, esto es, del tiempo de CPU transcurrido).

Y dichos criterios son regulados con un comando que admite los siguientes argumentos:

- El número de snapshots por década, o 0 para ninguno.
- Cada cuantos saltos o eventos se quiere generar un snapshot, junto con una bandera de verdadero/falso adicional para activar/desactivar esta característica.
- Una bandera de verdadero/falso que informe a DADOS si se quiere guardar o descartar la información atomística en 3D generada. Esta información puede generar una cantidad considerable de datos y espacio en memoria, por lo que en algunos casos se puede descartar y limitarse a usar la información recogida en los histogramas y perfiles, a los que no afecta esta opción.
- Una bandera de verdadero/falso que indique si se desean perfiles globales o incrementales. Un perfil global va almacenando toda la estadística desde el principio de la simulación, el incremental únicamente desde el anterior snapshot.

- El número máximo de snapshots que se desea almacenar, de tal forma que si se generan más de este número, se van descartando los más antiguos y se almacenan los recientes.

Generando los perfiles en una dimensión

Cuando se generan los perfiles unidimensionales, DADOS adopta dos maneras diferentes para hacerlo, según las partículas involucradas en el perfil sean o no móviles.

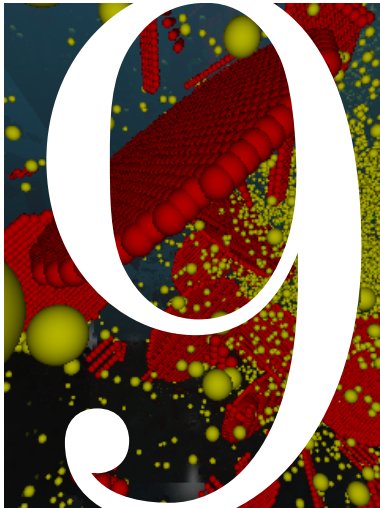
Para las partículas inmóviles es suficiente con contar el número de partículas que hay en un tiempo t concreto y calcular con ello la concentración. En cambio, para partículas móviles este sistema no funciona bien, siendo fácil entender por qué si acudimos al ejemplo de un intersticial. Para un I , la frecuencia con que éste migra es mucho mayor que la frecuencia con la que es emitido desde los clusters o desde la superficie, con lo cual, excepto tras los primeros instantes de implantación, es difícil tener más de uno o dos intersticiales a la vez en la simulación en un tiempo concreto t , ya que muy rápidamente se mueven y son atrapados o recombinados. Por ello, si se tomara la concentración de ellos en un tiempo concreto t , se observaría un pico muy alto en la región en que se encuentra en ese momento y una concentración nula en las demás. Para evitar este problema, DADOS calcula la concentración de las especies móviles utilizando otra alternativa, que consiste en contar el número de saltos que cada partícula ha dado en cada determinada región del espacio entre dos intervalos de tiempo. La anchura de la región del espacio se puede variar a través del parámetro N_{slice} , como se comenta en la sección 8.1.3. Posteriormente, cuando se requiera la creación de un “snapshot”, se calcula la concentración utilizando dicho número de saltos y la migración (conocida) de cada especie. En modo incremental se borra dicha información del número de saltos después de cada imagen, mientras que en modo no incremental se sigue almacenando haciendo un promedio desde el principio de la simulación.

Lógicamente, esta segunda forma de calcular la concentración da un valor de 0 para las especies inmóviles, puesto que éstas no saltan.

Calculando la supersaturación

La supersaturación de los I s o V s es la relación entre la concentración actual y su concentración en equilibrio. Nuestro modelo calcula la supersaturación para los defectos puntuales a partir de la concentración actual de dichas especies, que extrae de su número de saltos, como se ha explicado anteriormente, y consigue la concentración en equilibrio relacionándola con la frecuencia de emisión de las intercaras, que son quienes establecen dicho valor.

Más detalles
en sec. A.3.4,
pág. 334.



Especies implementadas en DADOS

| | |
|---|-----|
| 9.1. El intersticial propio del silicio | 185 |
| 9.2. Vacante | 187 |
| 9.3. Boro | 189 |
| 9.4. Arsénico | 196 |
| 9.5. Flúor | 201 |
| 9.6. Carbono | 204 |
| 9.7. Fósforo | 207 |
| 9.8. Oxígeno | 211 |
| 9.9. Indio | 213 |

Una vez explicados en los capítulos precedentes los aspectos teóricos y modelos prácticos que están implementados en DADOS, vamos ahora a exponer cómo afectan estos modelos y parámetros a cada especie en concreto.

▷ 9.1

El intersticial propio del silicio

Los intersticiales propios del silicio se van a encontrar simulados por DADOS como defectos puntuales con distintos estados de carga, o bien en pequeños clusters, defectos extendidos tipo $\{311\}$ y también como discos de dislocación. Las asociaciones particulares del I con cada uno de los diferentes dopantes se encuentra detallada en la sección del dopante bajo consideración, y la formación de pequeños clusters de I s, $\{311\}$ s y discos de dislocación puede ser consultada en las secciones 8.3.1, 8.3.2 y 8.3.3 respectivamente. Esta sección se limita a describir las interacciones entre I y V .

▷ 9.1.1

Defectos puntuales del I

Los defectos puntuales del intersticial del silicio que han sido implementados en DADOS son:

| Nombre | DADOS | Descripción | Carga |
|--------|-------|---------------------|-------|
| I^0 | I | Intersticial propio | 0 |
| I^- | IM | Intersticial propio | - |
| I^+ | IP | Intersticial propio | + |

▷ 9.1.2

Cómo se introducen I s en la simulación

Durante las simulaciones, los intersticiales pueden ser introducidos por

- Implantación de silicio, o daño causado por la implantación de Si u otro ion. En estos casos la especie introducida es I^0 .
- Lectura de un perfil (externo) de I^0 .
- Emisión de I^0 desde las intercaras.

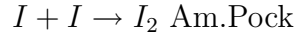
▷ 9.1.3

Interacciones del I

A continuación se detallan todas las reacciones que involucran I s y V s. Las interacciones de I con dopantes u otras impurezas se encontrarán consultando la impureza concreta. Las reacciones que se explicitan están escritas de manera general, sin entrar en todas las posibles combinaciones debidas al nivel de carga. Para tener éstas en cuenta hay que señalar que el modelo usado en DADOS prohíbe

Modelo en la
sec. 8.2.2,
pág. 110.

reacciones repulsivas (+, +) y (-, -). De esta manera, si quisiéramos interpretar la reacción

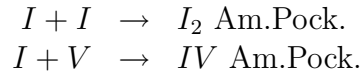


hemos de:

- Considerar todos los posibles estados de carga del intersticial I^0 , I^- e I^+ .
- Permitir únicamente las reacciones no repulsivas, es decir, $I^+ + I^0$, $I^- + I^0$, $I^0 + I^0$ e $I^+ + I^-$.

En ciertos casos el resultado es un “pocket” amorfo, en otros un cluster de impureza. Algunas consideraciones especiales que hay que tener en cuenta para estos casos serán comentadas más adelante.

La reacciones del I son:



Modelo en la sec. 8.2, pág. 109. Los prefactores y energías de migración, niveles electrónicos y el nombre de estos parámetros en el fichero de definición de parámetros del DADOS son:

| | Migración | | Elect. eV | Enlace | |
|-------|----------------------------------|--------------------------|-------------------------------|----------------------------|-----------|
| | cm^2s^{-1} | eV | | cm^2s^{-1} | eV |
| I | Dm_I $D_{m,0}(I)$ 0.005 | Em_I $E_m(I)$ 1 | Sin datos | Sin datos | Sin datos |
| I^+ | Dm_IP $D_{m,0}(I^+)$ 0.005 | Em_IP $E_m(I^+)$ 1 | e_IP0 $e(+,0)(I^+)$ 0.4 | Sin datos | Sin datos |
| I^- | Dm_IM $D_{m,0}(I^-)$ 0.005 | Em_IM $E_m(I^-)$ 1 | e_IM0 $e(0,-)(I^-)$ 1 | Sin datos | Sin datos |

Interacciones con intercaras

Modelo en la sec. 8.7.1, pág. 149.

La partícula que se absorbe y emite desde las intercaras es el I^0 . Los parámetros superficiales para el intersticial son:

| Símbolo | Nombre ddp | Valor |
|---|------------|--------------------------------------|
| $D_{\text{emission},0}^{\text{interface}}(I)$ | D0FS_I | $50000 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_f(I)$ | Eform_I | 3.8 eV |
| $L_r(I)$ | RecLnm_I | 0 nm |

▷ 9.1.4

Disolución de “pockets” amorfos

Modelo en la sec. 8.4.1, pág. 127. Los pockets amorfos se intentan disolver en clusters con 4 intersticiales cada uno, (I_4), siempre que sea posible, si no es así, se emite un I , o bien se forman clusters I_2 o I_3 según el número de I s que haya disponibles.

▷ 9.2

Vacante

La vacante propia del silicio puede encontrarse en las simulaciones de DADOS en diferentes estados de carga, asociada con impurezas, en pequeños clusters de V s, o en “huecos”. La asociación de las vacantes con impurezas está explicada en la sección de cada dopante involucrado, y la formación de pequeños clusters y huecos se encuentra detallada en las secciones 8.3.1 y 8.3.4 respectivamente. Esta sección se limita a comentar las interacciones de V consigo misma y con I .

▷ 9.2.1

Defectos puntuales de V

Los estados de carga de V implementados en DADOS son:

| Nombre | DADOS | Descripción | Carga |
|----------|-------|-------------|-------|
| V^0 | V | Vacante | 0 |
| V^- | VM | Vacante | -1 |
| V^+ | VP | Vacante | +1 |
| V^{--} | VMM | Vacante | -2 |
| V^{++} | VPP | Vacante | +2 |

▷ 9.2.2

Cómo se introducen V s en la simulación

Las vacantes aparecen en la simulación debido a:

- Generación de dañado durante una implantación: Se introduce la especie V^0 .
- Lectura de un perfil externo V^0 .
- Emisión de V^0 desde la superficie o alguna intercara.

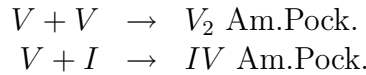
▷ 9.2.3

Interacciones con V

Modelo en la sec. 8.2.2, pág. 110. A continuación se van a detallar las reacciones que involucran defectos puntuales V s y/o I s. Las interacciones de V con dopantes o impurezas se encuentran listadas en el apartado que trate la impureza. Dichas reacciones están escritas de una forma general, que no tiene en cuenta el estado de carga. Debemos resaltar que, en DADOS, se prohíben las reacciones entre estados con igual carga, tipo $(+, +)$ o $(-, -)$. De esta manera, para interpretar la reacción general $V + V \rightarrow V_2$ Am.Pock habrá que:

- Considerar todos los posibles estados de carga de la vacante V^0 , V^- , V^+ , V^{++} y V^{--} .
- Permitir sólo las reacciones no repulsivas, que en este caso serán $V^0 + V^{++}$, $V^0 + V^+$, $V^0 + V^0$, $V^0 + V^-$, $V^0 + V^{--}$, $V^+ + V^-$, $V^+ + V^{--}$, $V^- + V^{++}$ y $V^{++} + V^{--}$.

Con esto, las reacciones involucrando vacantes son:



Modelo en la sec. 8.2, pág. 109. Los prefactores y energías de migración, niveles electrónicos, y nombres en el fichero de parámetros del DADOS son:

| | Migración | | Elect. eV | Enlace | |
|----------|--------------------------------------|--------------------------------|--------------------------------------|----------------------------|-----------|
| | cm^2s^{-1} | eV | | cm^2s^{-1} | eV |
| V | Dm_V $D_{m,0}(V)$ 0.001 | Em_V $E_m(V)$ 0.4 | Sin datos | Sin datos | Sin datos |
| V^+ | Dm_VP $D_{m,0}(V^+)$ 0.001 | Em_VP $E_m(V^+)$ 0.6 | e_VP0 $e(+, 0)(V^+)$ 0.03 | Sin datos | Sin datos |
| V^{++} | Dm_VPP $D_{m,0}(V^{++})$ 0.001 | Em_VPP $E_m(V^{++})$ 0.8 | e_VPP0 $e(+, ++)(V^{++})$ 0.13 | Sin datos | Sin datos |
| V^- | Dm_VM $D_{m,0}(V^-)$ 0.001 | Em_VM $E_m(V^-)$ 0.4 | e_VM0 $e(0, -)(V^-)$ 0.6 | Sin datos | Sin datos |
| V^{--} | Dm_VMM $D_{m,0}(V^{--})$ 0.001 | Em_VMM $E_m(V^{--})$ 0.4 | e_VMM0 $e(-, --)(V^{--})$ 1.06 | Sin datos | Sin datos |

Interacciones con las intercaras

Modelo en la
sec. 8.7.1,
pág. 149.

La superficie y/o las intercaras absorben y emiten V^0 . Sus parámetros respecto a la vacante son:

| Símbolo | Nombre ddp | Valor |
|---|------------|------------------------------------|
| $D_{\text{emission},0}^{\text{interface}}(V)$ | DOFS_V | $800 \text{ cm}^{-2}\text{s}^{-1}$ |
| $E_f(V)$ | Eform_V | 3.8 eV |
| $L_r(V)$ | RecLnm_V | 0 nm |

▷ 9.2.4

Disolución de pockets amorfos

Los pockets amorfos, al disolverse, intentan formar clusters tipo V_3 con las vacantes que les queda. Cuando esto no es posible emiten una V , o forman clusters V_2 , según el número de V s sobrantes.

Modelo en la
sec. 8.4.1,
pág. 127.

▷ 9.3

Boro

El boro se encuentra en el programa DADOS formando defectos puntuales de impurezas o en clusters de impurezas del tipo $B_n I_m$.

▷ 9.3.1

Defectos puntuales con B

Los defectos puntuales con boro que incorpora DADOS son:

| Nombre | DADOS | Descripción | Carga |
|------------|----------|--------------------|-------|
| B^- | B | Boro sustitucional | -1 |
| $BOxide$ | BOxide | Boro en SiO_2 | 0 |
| $BNitride$ | BNitride | Boro en nitruro | 0 |
| B_i^0 | Bi | Boro intersticial | 0 |
| B_i^+ | BiP | Boro intersticial | +1 |
| B_i^- | BiM | Boro intersticial | -1 |

▷ 9.3.2

Introduciendo B en la simulación

Se puede introducir boro en la simulación a partir de:

- Implantación iónica (se introduce B_i^0).

- Lectura de un perfil de B^- .
- Emisión superficial de boro (anteriormente capturado o por un proceso de *in-diffusion*): B_i^0 .

▷ 9.3.3

Interacciones del boro

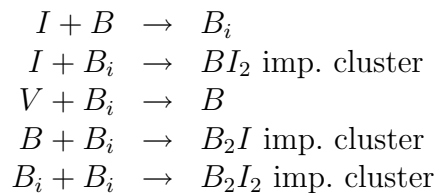
Modelo en la
sec. 8.2.2,
pág. 110.

Las reacciones que se van a detallar a continuación están escritas de una forma genérica que no considera el estado de carga, de tal forma que hemos de comentar que DADOS prohíbe las reacciones repulsivas, aquellas con igual carga, como $(+, +)$ y $(-, -)$. La carga del estado final varía con el nivel de Fermi de la región donde se produzca la reacción, entre todos los estados posibles para dicha reacción. Por ejemplo, para interpretar la reacción $I + B \rightarrow B_i$:

- Se toman todas las cargas posibles para el intersticial del silicio: I^0, I^-, I^+ y para el boro sustitucional: B^- .
- Únicamente se permiten las reacciones no repulsivas, que serán $I^+ + B^-$ y $I^0 + B^-$.
- El resultado de estas reacciones será un B_i , cuyo estado de carga (B_i^+, B_i^0 o B_i^-) dependerá del nivel de Fermi local.

Cuando los criterios anteriores, por la razón que fuera, no sean aplicables, entonces se escribirá la reacción con los estados particulares de carga. Algunos casos especiales cuando aparecen pockets amorfos o clusters de impurezas serán considerados más adelante.

Las interacciones del boro son:



Los prefactores y energías de activación y enlace del boro en silicio, niveles electrónicos y los nombres utilizados en el fichero de parámetros del DADOS se listan a continuación:

Modelo en la
sec. 8.6.1,
pág. 139.

Departamento de Electricidad y Electrónica. Tesis de Ignacio Martín Bragado.

| | Migración | | Elect. eV | Enlace | |
|---------|-------------------------------------|-------------------------------|------------------------------------|------------------------------------|-------------------------------|
| | cm ² s ⁻¹ | eV | | cm ² s ⁻¹ | eV |
| B | Dm_B $D_{m,0}(B)$ 0 | Em_B $E_m(B)$ 5 | Sin datos | Sin datos | |
| B_i | Dm_Bi $D_{m,0}(B_i)$ 0.001 | Em_Bi $E_m(B_i)$ 0.25 | Sin datos | Sin datos | |
| B_i^+ | Dm_BiP $D_{m,0}(B_i^+)$ 0.001 | Em_BiP $E_m(B_i^+)$ 1.1 | e_BiP0 $e(+, 0)(B_i^+)$ 1.04 | Sin datos | |
| B_i^- | Dm_BiM $D_{m,0}(B_i^-)$ 0.001 | Em_BiM $E_m(B_i^-)$ 0.5 | e_BiM0 $e(0, -)(B_i^-)$ 0.8 | Db_BiM $D_{b,0}(B_i^-)$ 0.37 | Eb_BiM $E_b(B_i^-)$ 0.1 |

Los valores de la energía de migración para el boro en otros materiales que no sean silicio son

| | Migración | |
|------------|---|--|
| | cm ² s ⁻¹ | eV |
| $BOxide$ | Dm_BOxide $D_{m,0}(BOxide)$ 0.00312 | Em_BOxide $E_m(BOxide)$ 3.93 |
| $BNitride$ | Dm_BNitride $D_{m,0}(BNitride)$ 0.00312 | Em_BNitride $E_m(BNitride)$ 3.93 |

La figura 9.1 muestra un *plot* de Arrhenius con la difusión teórica del boro calculada con los parámetros listados más arriba para condiciones intrínsecas. La figura 9.2 hace un estudio similar para condiciones extrínsecas.

Interacciones con las intercaras

Los parámetros utilizados por el boro para modelizar sus interacciones con la superficie y/o intercaras son:

Modelo en la
sec. 8.7.2,
pág. 153.

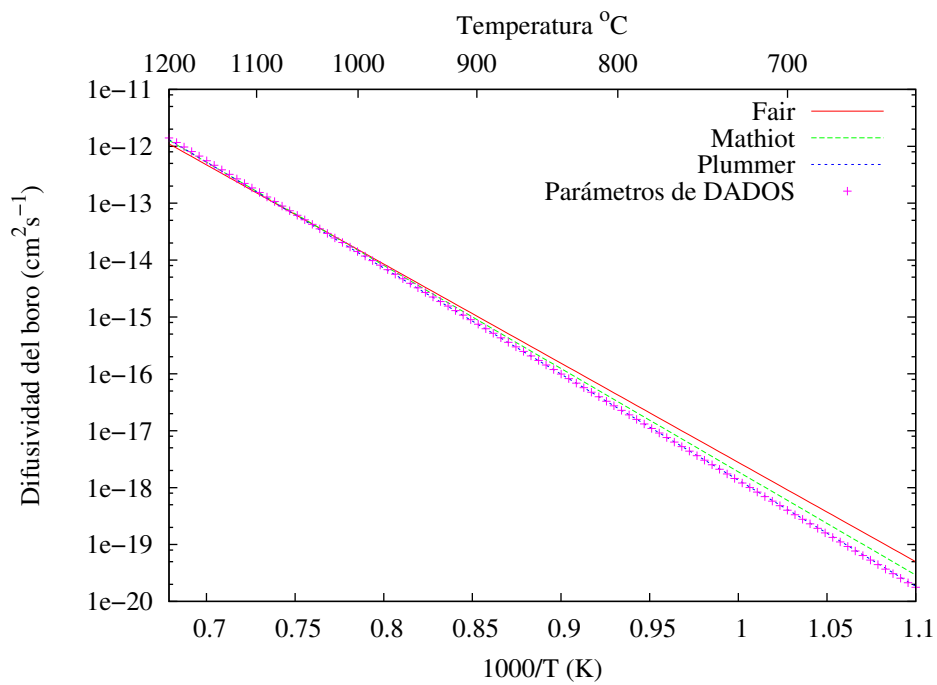


Figura 9.1: Difusión intrínseca del boro. Líneas: Valores experimentales para distintas referencias (Plummer et al., 2000; Fair, 1981a; Mathiot and Pfister, 1984; Hill, 1981; Haddara et al., 2000). Símbolos: Valores teóricos calculados con los parámetros de entrada de DADOS.

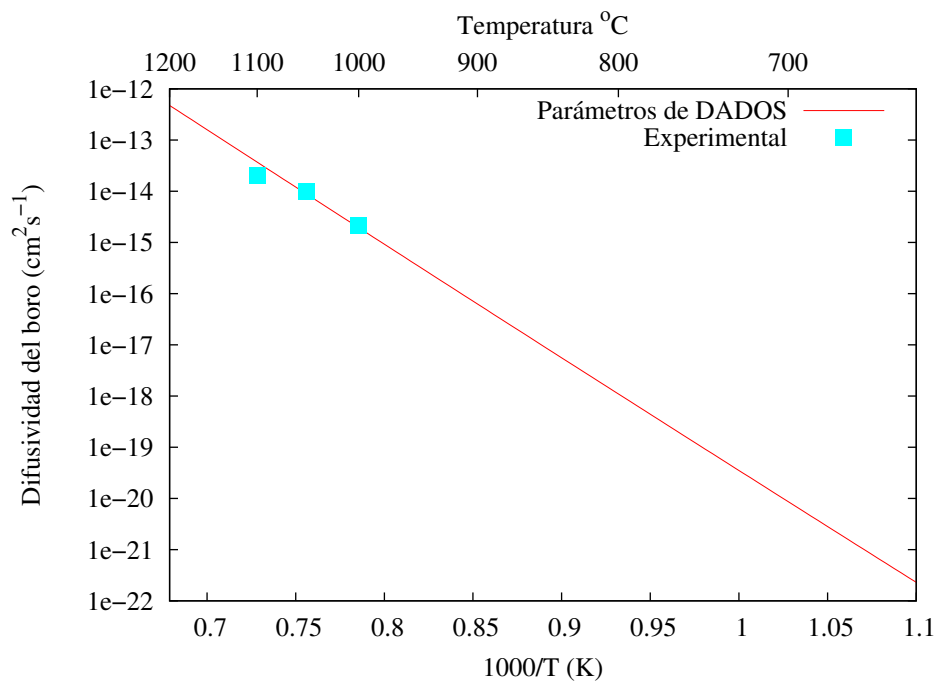


Figura 9.2: Difusión extrínseca del boro. Líneas: Valores teóricos calculados con los parámetros de entrada de DADOS. Símbolos: Valores experimentales tomados de la referencia (Fair, 1981a).

| | Emisión = B_i | Captura = B_i |
|---------------------|--|---|
| Parámetros | Prefactor | Energía |
| Solubilidad | C_{sol0_B} | E_{sol_B} |
| h | $C_{0,sol}(B) = 1e + 24 \text{ cm}^{-3}$ $Surf_h0_B$ $h_0(B) = 0,0166$ | $E_{sol}(B) = 1 \text{ eV}$ $Surf_Eh_B$ $E_h(B) = 0,486 \text{ eV}$ |
| e/a | $Surf_e_a0_B$ $(e/a)_0(B) = 0,178$ | $Surf_Ee_a_B$ $E_{(e/a)}(B) = -0,086 \text{ eV}$ |
| Segregación Óxido | $Segreg_pref_BOxide$ $g_{Oxi,0}(B) = 30$ | $Segreg_eV_BOxide$ $E_{Oxi,g} = 0$ |
| Segregación Nitruro | $Segreg_pref_BNitride$ $g_{Nit,0}(B) = 30$ | $Segreg_eV_BNitride$ $E_{Nit,g} = 0$ |
| Saturación | $SurfMaxTrap_B$ | $\sigma_{max}(B) = 2e + 14 \text{ cm}^{-2}$ |

▷ 9.3.4

Disolución de pockets amorfos

Modelo en la
sec. 8.4.1,
pág. 127.

Los pocket amorfos, cuando se disuelven, intentarán formar clusters de impurezas con dos intersticiales y un boro (BI_2), generando tantos como sea posible con el número de I_s y B_s que tenga atrapados. Cuando esto no sea posible, y según las partículas de B e I_s que le queden, emitirá B_i o depositará un B .

▷ 9.3.5

El B en los procesos de recristalización

La concentración máxima de boro sustitucional se fija en los procesos de recristalización en el valor $[B]_{\max}^{\text{recrys}}$. El resto de los átomos de boro, si sobran, son depositados como $B_n I_m$, donde los subíndices n y m son los valores máximos “Dopantes max. in clusters de B ” y “ I_s o V_s max. en clusters” respectivamente, los cuales se listan en la siguiente tabla:

| Nombre | valor ddp | Valor |
|---------------------------------------|-----------|---------------------------|
| $[B]_{\max}^{\text{recrys}}$ | (interno) | $3.5e+20 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de B | (interno) | 3 |
| Max. I_s o V_s en clusters de B | (interno) | 0 |

▷ 9.3.6

Salto largos del boro

Los saltos largos del boro aparecen de forma natural como consecuencia del proceso atómico de kick out. La figura 9.3 representa los valores teóricos, tomando los parámetros de entrada de DADOS listados más arriba, y usando la ecuación 8.40, frente a los resultados experimentales publicados en la referencia

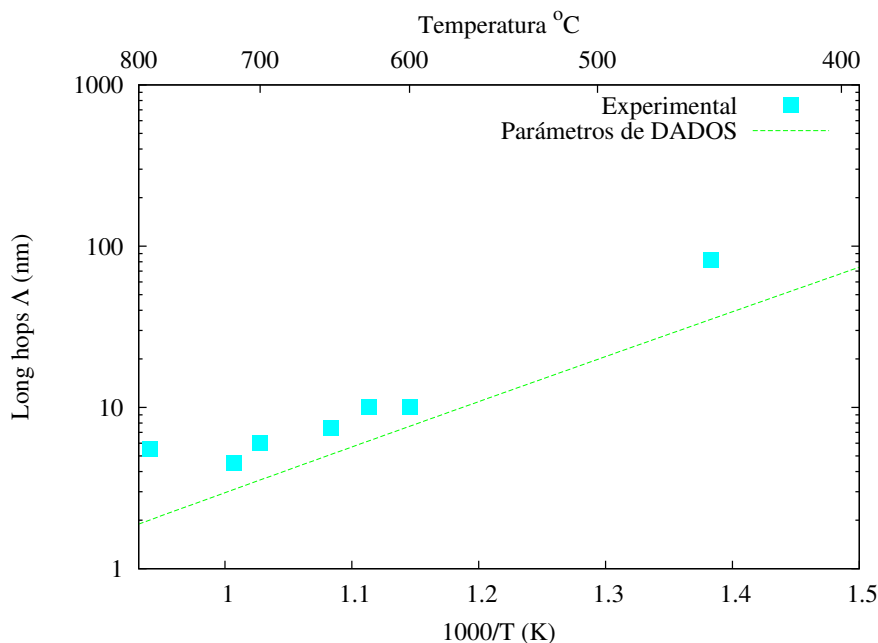


Figura 9.3: Saltos largos del boro (Λ). Líneas: Valores teóricos. Símbolos: Valores experimentales tomados de (Cowern et al., 1991; Cowern et al., 1992).

(Cowern et al., 1991; Cowern et al., 1992).

▷ 9.3.7

Clusters de impureza de B

Los clusters de impureza del boro (BIC) se forman con intersticiales. Las reacciones permitidas son:

Modelo en la
sec. 8.6.2,
pág. 143.

- Crecimiento: atrapando intersticiales, $B_n I_m + I \rightarrow B_n I_{m+1}$, o átomos de boro intersticial, $B_n I_m + B_i \rightarrow B_{n+1} I_{m+1}$.
- Emisión: emisión de intersticiales, $B_n I_m \rightarrow B_n I_{m-1} + I$, o de boro intersticial, $B_n I_m \rightarrow B_{n-1} I_{m-1}$.
- Recombinación recombinación de vacantes, $B_n I_m + V \rightarrow B_n I_{m-1}$.
- Frank-Turnbull: emisión de vacantes, $B_n I_m \rightarrow B_n I_{m+1} + V$.

La energía potencial utilizada para los clusters, E_{pot} , se encuentra indicada en la figura 9.4. Otros valores para los clusters de boro, conseguidos mediante diversas técnicas, así como estudios del proceso de formación de estos clusters, pueden encontrarse en las referencias (Pelaz et al., 1997; Pelaz et al., 1999; Lenosky et al., 2000; Caturla et al., 1998a; Caturla et al., 1998b).

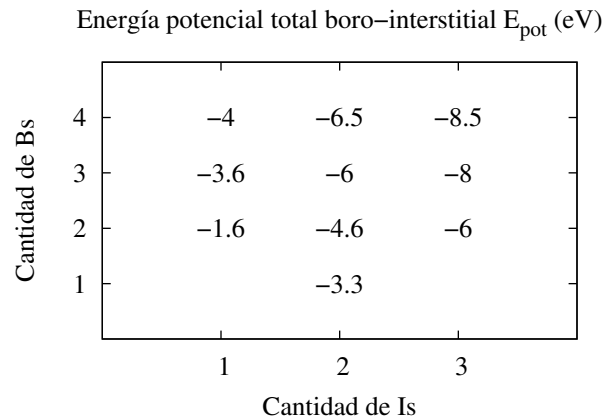


Figura 9.4: Energías potenciales para los clusters de boro con intersticiales.

▷ 9.4

Arsénico

El arsénico se encuentra en DADOS en defectos puntuales de impurezas, o bien en clusters de impurezas del tipo As_nV_m .

▷ 9.4.1

Defectos puntuales con As

Los defectos puntuales de DADOS que incluyen As son:

| Nombre | DADOS | Descripción | Carga |
|-------------|-----------|--------------------|-------|
| As^+ | As | As sustitucional | +1 |
| $AsOxide$ | AsOxide | As en SiO_2 | 0 |
| $AsNitride$ | AsNitride | As en nitruro | 0 |
| As_i^0 | Asi | As intersticial | 0 |
| As_i^+ | AsiP | As intersticial | +1 |
| As_V^0 | AsV | As vacante | 0 |
| As_V^+ | AsVP | As vacante | +1 |
| As_V^- | AsVM | As vacante | -1 |

▷ 9.4.2

Introduciendo As en la simulación

El arsénico se introduce en las simulaciones por:

- Implantación de As_i^0 .
- Lectura de un perfil externo de As^+ .

- Emisión de As_V^0 por parte de las intercaras (previamente capturado o por *in-diffusion*).

▷ 9.4.3

Interacciones del As

Interacciones con defectos puntuales

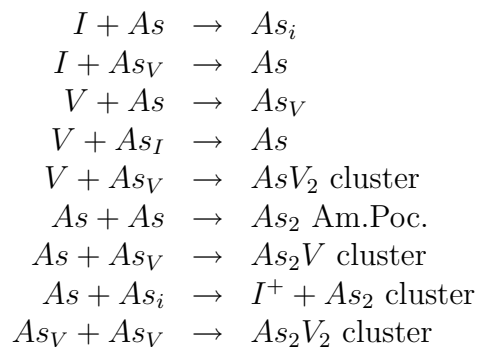
A continuación se detallan, escritas de una forma general que no explicita el estado de carga, las reacciones del arsénico con defectos puntuales. Debemos señalar que DADOS prohíbe las reacciones entre especies que se repelan, es decir, aquellas que tengan la misma carga (+, +) o (-, -). La carga de la especie final variará entre todas las posibles dependiendo del nivel de Fermi local. De esta manera, para interpretar una reacción, como por ejemplo $I + As \rightarrow As_i$ habrá que:

Modelo en la
sec. 8.2.2,
pág. 110.

- Tomar todos los estados de carga posibles del intersticial, I^0 , I^- e I^+ y para el arsénico sustitucional As^+ .
- Permitir únicamente las reacciones no repulsivas, en este caso $I^- + As$ y $I^0 + As$.
- El resultado de estas reacciones será un As_i , cuyo estado de carga (As_i^+ o As_i^0) dependerá del nivel local de Fermi.

Más adelante se comentarán algunas consideraciones especiales para las reacciones que involucran pockets amorfos y clusters de impurezas.

Las interacciones del arsénico con defectos puntuales son:



Los prefactores y energías de migración y enlace para el arsénico en silicio, niveles electrónicos y sus nombres en el fichero de parámetros de DADOS se listan a continuación:

Modelo en la
sec. 8.6.1,
pág. 139.

| | Migración | | Elect. eV | Enlace | |
|-----------------------------------|---|---|---|---|--|
| | cm ² s ⁻¹ | eV | | cm ² s ⁻¹ | eV |
| <i>As</i> | Dm_ <i>As</i> $D_{m,0}(As)$ 0 | Em_ <i>As</i> $E_m(As)$ 5 | Sin datos | Sin datos | |
| <i>As_i</i> | Dm_ <i>Asi</i> $D_{m,0}(As_i)$ 0.004 | Em_ <i>Asi</i> $E_m(As_i)$ 1.8 | Sin datos | Sin datos | |
| <i>As_i⁺</i> | Dm_ <i>AsiP</i> $D_{m,0}(As_i^+)$ 0.004 | Em_ <i>AsiP</i> $E_m(As_i^+)$ 1.8 | e_ <i>AsiP0</i> $e(+,0)(As_i^+)$ 0.1 | Db_ <i>AsiP</i> $D_{b,0}(As_i^+)$ 0.005 | Eb_ <i>AsiP</i> $E_b(As_i^+)$ 0.5 |
| <i>As_V</i> | Dm_ <i>AsV</i> $D_{m,0}(As_V)$ 0.00021 | Em_ <i>AsV</i> $E_m(As_V)$ 1.3 | Sin datos | Sin datos | |
| <i>As_V⁺</i> | Dm_ <i>AsVP</i> $D_{m,0}(As_V^+)$ 0.00021 | Em_ <i>AsVP</i> $E_m(As_V^+)$ 1.3 | e_ <i>AsVP0</i> $e(+,0)(As_V^+)$ 0.3 | Db_ <i>AsVP</i> $D_{b,0}(As_V^+)$ 0.001 | Eb_ <i>AsVP</i> $E_b(As_V^+)$ 1.01 |
| <i>As_V⁻</i> | Dm_ <i>AsVM</i> $D_{m,0}(As_V^-)$ 0.00021 | Em_ <i>AsVM</i> $E_m(As_V^-)$ 3.4 | e_ <i>AsVM0</i> $e(0,-)(As_V^-)$ 0.77 | Sin datos | |

Los valores de la energía de migración para el arsénico en otros materiales que no sean silicio son

| | Migración | |
|------------------|--|--|
| | cm ² s ⁻¹ | eV |
| <i>AsOxide</i> | Dm_ <i>AsOxide</i> $D_{m,0}(AsOxide)$ 0.000294 | Em_ <i>AsOxide</i> $E_m(AsOxide)$ 3.15 |
| <i>AsNitride</i> | Dm_ <i>AsNitride</i> $D_{m,0}(AsNitride)$ 0.000294 | Em_ <i>AsNitride</i> $E_m(AsNitride)$ 3.15 |

La figura 9.5 representa los *plot* de Arrhenius calculados con los valores anteriormente listados, para condiciones intrínsecas.

Interacciones con intercaras y/o superficies

Modelo en la
sec. 8.7.2,
pág. 153.

Los parámetros relacionados con las intercaras se listan a continuación:

Departamento de Electricidad y Electrónica. Tesis de Ignacio Martín Bragado.

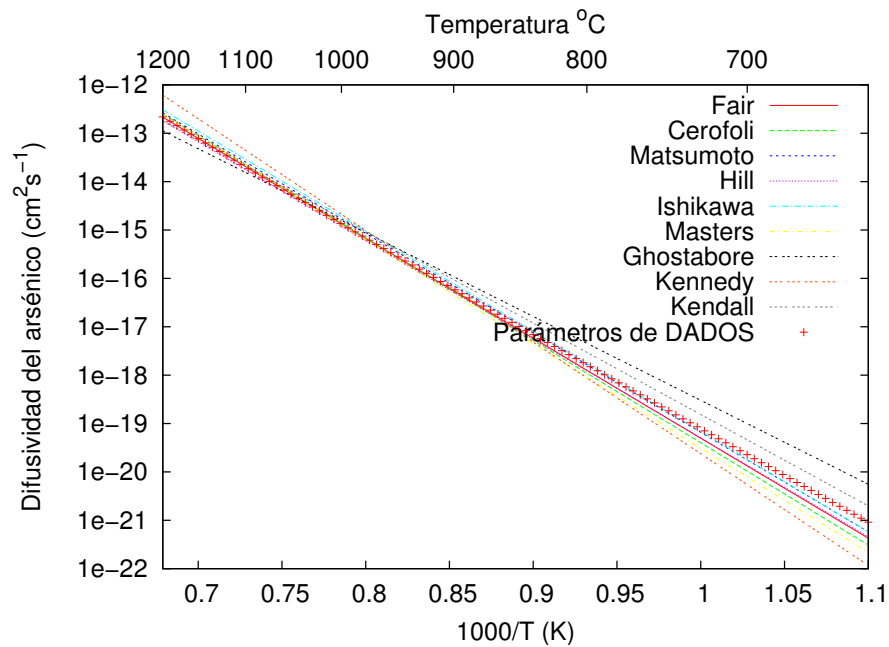


Figura 9.5: Difusión intrínseca del arsénico. Líneas: valores experimentales tomados de las referencias (Fair, 1981a; Ghoshtagore, 1971; Kennedy, 1971; Cerofolini et al., 1986; Matsumoto et al., 1980; Kendall, 1969; Ishikawa et al., 1982; Masters and Fairfield, 1969; Hill, 1981). Símbolos: valores teóricos calculados con los parámetros de entrada de DADOS.

| | Emisión = As_V | Captura = As_V |
|-------------|---|--|
| Parámetros | Prefactor | Energía |
| Solubilidad | C_{sol0_As} | E_{sol_As} |
| | $C_{0,Sol}(As) = 2,2e + 22 \text{ cm}^{-3}$ | $E_{Sol}(As) = 0,47 \text{ eV}$ |
| h | $Surf_h0_As$ | $Surf_Eh_As$ |
| | $h_0(As) = 0,0231$ | $E_h(As) = 0,766 \text{ eV}$ |
| e/a | $Surf_e_a0_As$ | $Surf_Ee_a_As$ |
| | $(e/a)_0(As) = 6,29e - 05$ | $E_{(e/a)}(As) = -0,738 \text{ eV}$ |
| Segregación | $Segreg_pref_AsOxide$ | $Segreg_eV_AsOxide$ |
| Óxido | $g_{Oxi,0}(As) = 30$ | $E_{Oxi,g} = 0$ |
| Segregación | $Segreg_pref_AsNitride$ | $Segreg_eV_AsNitride$ |
| Nitruro | $g_{Nit,0}(As) = 30$ | $E_{Nit,g} = 0$ |
| Saturación | $SurfMaxTrap_As$ | $\sigma_{max}(As) = 2e + 14 \text{ cm}^{-2}$ |

Disolución de los pocket amorfos

Modelo en la
sec. 8.4.1,
pág. 127.

Al disolverse los pockets amorfos intentan formar la mayor cantidad posible de clusters As_4V , siempre que haya arsénico y vacantes suficientes. Cuando esto no es posible, los pockets amorfos liberan As_4 (que en seguida, a partir de un mecanismo de Frank-Turnbull, emiten un I y se transforman en As_4V). Cuando no hay suficientes arsénicos para estas formaciones, el pocket amorfo libera As como defectos puntuales simples.

▷ 9.4.4

Recristalización y As

Al recristalizar las zonas amorfas del silicio se fija la concentración máxima de arsénico dopante en un valor $[As]_{max}^{recrys}$. El resto de átomos de arsénico que pudieran quedar son precipitados como As_nV_m , donde los parámetros n y m son “Dopantes max. de As en clusters” y “ Is o Vs max. en clusters de As ” respectivamente, cuyos valores son listados a continuación:

| Nombre | valor ddp | Valor |
|--------------------------------------|-----------|-------------------------|
| $[As]_{max}^{recrys}$ | (interno) | $1e+21 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de As | (interno) | 4 |
| Max. Is o Vs en clusters de As | (interno) | 0 |

▷ 9.4.5

Clusters de impurezas de As

Modelo en la
sec. 8.6.2,
pág. 143.

El arsénico forma clusters de impurezas junto con las vacantes, permitiéndose las siguientes reacciones:

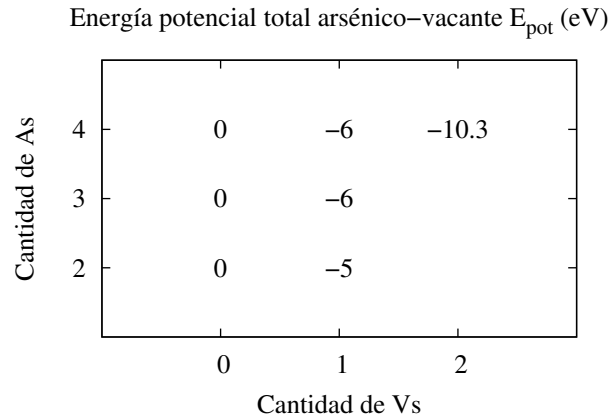


Figura 9.6: Energías potenciales para los clusters de arsénico con vacantes (As_nV_m).

- Crecimiento: capturando V , $As_nV_m + V \rightarrow As_nV_{m+1}$, y As_V , $As_nV_m + As_V \rightarrow As_{n+1}V_{m+1}$.
- Emisión: de V , $As_nV_m \rightarrow As_nV_{m-1} + V$, y As_V , $As_nV_m \rightarrow As_{n-1}V_{m-1} + As_V$.
- Recombinación: de I , $As_nV_m + I \rightarrow As_nV_{m-1}$, y As_i , $As_i + As_nV_m \rightarrow As_{n+1}V_{m-1}$.
- Frank-Turnbull: $As_nV_m \rightarrow As_nV_{m+1} + I$.
- Frank-Turnbull complementario: $As_nV_m \rightarrow As_{n-1}V_{m+1} + As_i$.

Los parámetros de energía E_{pot} necesarios para los As_nV_m se encuentran listados en la figura 9.6.

▷ 9.5

Flúor

El flúor (que está parcialmente implementado en DADOS), se puede encontrar como un átomo intersticial o en clusters de impurezas con vacantes F_nV_m .

▷ 9.5.1

Defectos puntuales de F

Se ha implementado un único tipo de F en DADOS, al que llamamos simplemente F . Puesto que los átomos de flúor sustitucional son menos estables que

los de F en posición intersticial (de Walle et al., 1988), el F implementado se trata en realidad de F_i .

▷ 9.5.2

Introduciendo F en la simulación

El flúor puede aparecer en la simulación debido a:

- Una implantación de F .
- Lectura de un perfil de F .

La superficie no emite F , se limita a recombinarlo.

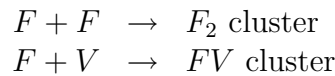
▷ 9.5.3

Interacciones del F

Interacciones con defectos puntuales

Modelo en la
sec. 8.2.2,
pág. 110.

Las interacciones puntuales del F están relacionada con la formación de clusters de F :



Modelo en la
sec. 8.6.1,
pág. 139.

La energía y prefactor de migración, junto con el nombre que estos parámetros tienen en DADOS, son:

| | Migración | | Elect. | Enlace | |
|-----|-------------------------------|-------------------------|-----------|----------------------------|----|
| | cm^2s^{-1} | eV | eV | cm^2s^{-1} | eV |
| F | Dm_F $D_{m,0}(F)$ 0.005 | Em_F $E_m(F)$ 0.8 | Sin datos | Sin datos | |

Interacciones con la superficie e intercaras

Modelo en la
sec. 8.7.2,
pág. 153.

Los parámetros relacionados con la interacción del F en las superficies y/o intercaras son:

| | Emisión = F | Captura = F |
|-------------|------------------------------------|---|
| Parámetros | Prefactor | Energía |
| Solubilidad | C_{sol0_F} | E_{sol_F} |
| | $C_{0,sol}(F) = 0 \text{ cm}^{-3}$ | $E_{sol}(F) = 0 \text{ eV}$ |
| h | $Surf_h0_F$ | $Surf_Eh_F$ |
| | $h_0(F) = 100$ | $E_h(F) = 0 \text{ eV}$ |
| e/a | $Surf_e_a0_F$ | $Surf_Ee_a_F$ |
| | $(e/a)_0(F) = 0,004$ | $E_{(e/a)}(F) = -0,37 \text{ eV}$ |
| Segregación | $Segreg_pref_FOxide$ | $Segreg_eV_FOxide$ |
| Óxido | $g_{Oxi,0}(F) =$ | $E_{Oxi,g} =$ |
| Segregación | $Segreg_pref_FNitride$ | $Segreg_eV_FNitride$ |
| Nitruro | $g_{Nit,0}(F) =$ | $E_{Nit,g} =$ |
| Saturación | $SurfMaxTrap_F$ | $\sigma_{max}(F) = 1e + 18 \text{ cm}^{-2}$ |

Disolución de pockets amorfos con F

Los pockets amorfos, al disolverse, tratarán de formar la mayor cantidad posible de clusters F_6V_2 con los átomos de flúor y las vacantes disponibles. Cuando no haya suficiente cantidad de partículas o vacantes, el pocket amorfo intenta disolverse en F_nV_m , donde $1 \leq m < 2$ y $1 \leq n \leq 6$. Cuando no haya ninguna vacante disponible, el flúor capturado se liberará como defectos puntuales de F .

Modelo en la
sec. 8.4.1,
pág. 127.

▷ 9.5.4

Recristalización y F

Al producirse la recristalización de las regiones amorfas, se fija la concentración del flúor en un valor máximo de $[F]_{\max}^{\text{recrys}}$. El resto de los átomos de flúor, si es que quedara alguno, se depositan como clusters F_nV_m donde n y m son los índices máximos marcados por “Dopantes max. en clusters de F ” y “ I_s o V_s max en clusters de F ” respectivamente. La siguiente tabla muestra dichos valores:

| Nombre | valor ddp | Valor |
|---------------------------------------|-----------|-------------------------|
| $[F]_{\max}^{\text{recrys}}$ | (interno) | $5e+22 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de F | (interno) | 0 |
| Max. I_s o V_s en clusters de F | (interno) | 0 |

▷ 9.5.5

Clusters de impurezas con F

El F está modelizado para formar clusters de impurezas junto con vacantes. No obstante, como el F siempre se encuentra en posición intersticial, no sustitucional (de Walle et al., 1988), el flúor difunde como F_i y crea clusters del tipo $(F_i)_nV_m$. Ahora bien, la notación anterior puede ser hartamente confusa y, por tanto,

Modelo en la
sec. 8.6.2,
pág. 143.

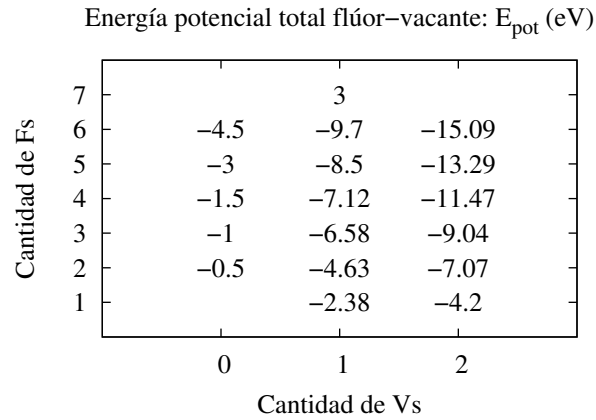


Figura 9.7: Energías potenciales para clusters de flúor con vacantes ($F_n V_m$).

en DADOS nos hemos limitado a llamar F al F_i^1 . Con la anterior notación los clusters de F son simplemente $F_n V_m$.

Las reacciones permitidas para estas agrupaciones de impurezas con vacantes son:

- Crecimiento: por captura de F , $F_n V_m + F \rightarrow F_{n+1} m + 1$, y V , $F_n V_m + V \rightarrow F_n + V_{m+1}$.
- Emisión: de flúor, $F_n V_m \rightarrow F_{n-1} V_m$, y vacantes, $F_n V_m \rightarrow F_n V_{m-1}$.
- Recombinación: $F_n V_m + I \rightarrow F_n V_{m-1}$.
- Frank-Turnbull: emisión de intersticiales, $F_n V_m \rightarrow F_n V_{m+1} + I$.

Las energías de potencial utilizadas para estos clusters se muestran en la figura 9.7.

▷ 9.6

Carbono

El carbono se pueden encontrar en DADOS en forma de defecto puntual de impureza, o bien formando clusters de impurezas con intersticiales $C_n I_m$.

¹Lo cual es una excepción porque por ejemplo, para el boro, llamamos B_i al boro intersticial y B al boro activo sustitucional.

▷ 9.6.1 Defectos puntuales con C

Los defectos puntuales con carbono incluidos en DADOS son:

| Nombre | DADOS | Descripción | Carga |
|---------|-------|-----------------------|-------|
| C^0 | C | Carbono sustitucional | 0 |
| C_i^0 | Ci | Carbono intersticial | 0 |

▷ 9.6.2 Introduciendo C en la simulación

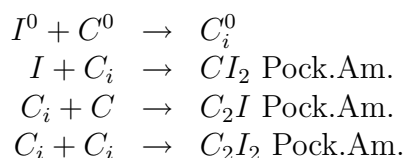
El carbono se introduce en la simulación a través de:

- Implantación de C_i^0 .
- Lectura de un perfil externo de C^0 .

▷ 9.6.3 Interacciones con C

Interacciones con defectos puntuales

Las interacciones con defectos puntuales del carbono son:



Modelo en la
sec. 8.2.2,
pág. 110.

Los prefactores y energías de migración, junto con los nombres definidos para ellos en el fichero de parámetros del DADOS son:

| | Migración | | Elect. eV | Enlace | |
|-------|----------------------------------|-----------------------------|--------------|--------------------------------|----------------------------|
| | cm^2s^{-1} | eV | | cm^2s^{-1} | eV |
| C | Dm_C $D_{m,0}(C)$ 0 | Em_C $E_m(C)$ 5 | Sin datos | Sin datos | |
| C_i | Dm_Ci $D_{m,0}(C_i)$ 0.001 | Em_Ci $E_m(C_i)$ 0.87 | Sin datos | Db_Ci $D_{b,0}(C_i)$ 2.5 | Eb_Ci $E_b(C_i)$ 1.5 |

Modelo en la
sec. 8.6.1,
pág. 139.

Disolución de pockets amorfos y el C

Los pockets amorfos intentan crear clusters de C_3I en su disolución final, siempre que tengan atrapados el suficiente número de C s e I s. En caso de ausencia de I s, pueden emitir C_3 , y ante la falta de un C liberan C_2I . Por último, cuando

Modelo en la
sec. 8.4.1,
pág. 127.

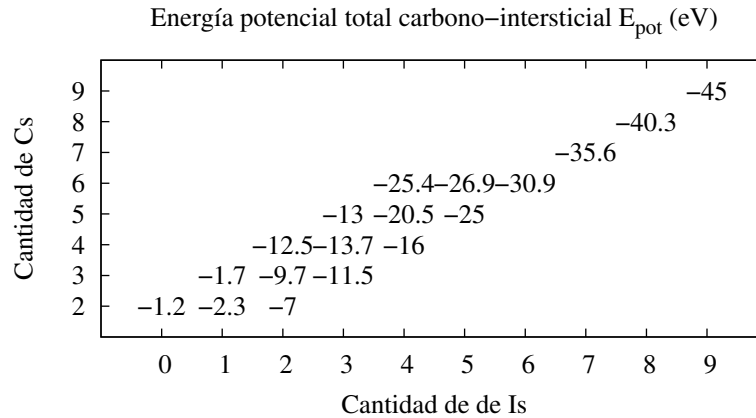


Figura 9.8: Energías potenciales de los clusters de impureza del carbono con I_s (C_nI_m).

ni siquiera estas opciones menos exigentes son posibles, el pocket depositaría C o C_i .

▷ 9.6.4

Clusters de impurezas de C

Los clusters de impureza de carbono se forman aglomerando también I del Modelo en la silicio. Las reacciones permitidas son:

sec. 8.6.2,
pág. 143.

- Crecimiento: atrapando intersticiales, $C_nI_m + I \rightarrow C_nI_{m+1}$, o carbono intersticial $C_nI_m + C_i \rightarrow C_{n+1}I_{m+1}$.
- Emisión: emisión de intersticiales, $C_nI_m \rightarrow C_nI_{m-1} + I$, o carbono intersticial, $C_nI_m \rightarrow C_{n-1}I_{m-1}$.
- Recombinación: recombinación de vacantes, $C_nI_m + V \rightarrow C_nI_{m-1}$.
- Frank-Turnbull: emisión de vacantes, $C_nI_m \rightarrow C_nI_{m+1} + V$.

Las energías potenciales E_{pot} , utilizadas por el programa, se encuentran en la figura 9.8. El carbono es conocido por disminuir la difusividad de los intersticiales del silicio, tal y como se comenta en las referencias (Pinacho et al., 2002; Johnson et al., 1998).

▷ 9.7

Fósforo

El fósforo se encuentra en las simulaciones de DADOS, bien formando defectos de impurezas, o en clusters con vacantes tipo P_nV_m .

▷ 9.7.1

Defectos puntuales con P

Los defectos puntuales definidos en DADOS que involucran fósforo son:

| Nombre | DADOS | Descripción | Carga |
|------------|-----------|-----------------------|-------|
| P^+ | Ph | Fósforo sustitucional | +1 |
| $POxide$ | PhOxide | Fósforo en SiO_2 | 0 |
| $PNitride$ | PhNitride | Fósforo en nitruro | 0 |
| P_i^0 | Phi | Fósforo intersticial | 0 |
| P_i^+ | PhiP | Fósforo intersticial | +1 |
| P_V^0 | PhV | Fósforo vacante | 0 |
| P_V^+ | PhVP | Fósforo vacante | +1 |
| P_V^- | PhVM | Fósforo vacante | -1 |

▷ 9.7.2

Introduciendo P en la simulación

El fósforo puede introducirse en las simulaciones a través de:

- La implantación de P_i^0 .
- La lectura de un perfil externo de P^+ .
- Le emisión superficial o de una intercara de P_i^0 .

▷ 9.7.3

Interacciones del fósforo**Interacciones del fósforo como defecto puntual**

Las reacciones que van a explicitarse más adelante están escritas de una manera general, sin concretar los estados de carga. Para completar dichas reacciones hay que tener en cuenta que DADOS no permite ninguna reacción entre partículas con la misma carga repulsiva (+, +) o (-, -), y que la carga del estado final variará entre todas las posibles dependiendo del nivel de Fermi local de la región en que ocurra la reacción. Así, por ejemplo, para interpretar la reacción general $I + P \rightarrow P_i$ habrá que:

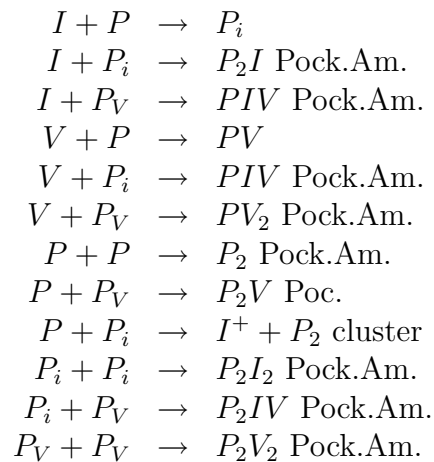
Modelo en la
sec. 8.2.2,
pág. 110.

- Considerar todas las cargas posibles del intersticial I^0 , I^- , I^+ y del fósforo sustitucional P^+ .

- Permitir únicamente las reacciones no repulsivas, en este caso $I^- + P^+$ y $I^0 + P^+$.

- Conocer que el resultado final de la reacción podrá ser P_i^+ o P_i^0 según el nivel de Fermi local.

Las reacciones del fósforo como defecto puntual consideradas en DADOS son:



Los prefactores y energías de migración del fósforo en silicio, prefactores y energías de enlace, niveles electrónicos y todos sus nombres tal como aparecen en el fichero de parámetros de DADOS son

Modelo en la sec. 8.6.1, pág. 139. Departamento de Electricidad y Electrónica. Tesis de Ignacio Martín Bragado.

| | Migración | | Elect. eV | Enlace | |
|---------|-------------------------------------|--------------------------------|-----------------------------------|--------------------------------------|--------------------------------|
| | cm ² s ⁻¹ | eV | | cm ² s ⁻¹ | eV |
| P | Dm_Ph $D_{m,0}(P)$ 0 | Em_Ph $E_m(P)$ 5 | Sin datos | Sin datos | |
| P_i | Dm_Phi $D_{m,0}(P_i)$ 1e-06 | Em_Phi $E_m(P_i)$ 0.8 | Sin datos | Sin datos | |
| P_i^+ | Dm_PhiP $D_{m,0}(P_i^+)$ 0.01 | Em_PhiP $E_m(P_i^+)$ 2.1 | e_PhiP0 $e(+,0)(P_i^+)$ 0.6 | Db_PhiP $D_{b,0}(P_i^+)$ 0.25 | Eb_PhiP $E_b(P_i^+)$ 0.7 |
| P_V | Dm_PhV $D_{m,0}(P_V)$ 0 | Em_PhV $E_m(P_V)$ 5 | Sin datos | Sin datos | |
| P_V^+ | Dm_PhVP $D_{m,0}(P_V^+)$ 0 | Em_PhVP $E_m(P_V^+)$ 5 | e_PhVP0 $e(+,0)(P_V^+)$ 0 | Db_PhVP $D_{b,0}(P_V^+)$ 0.005 | Eb_PhVP $E_b(P_V^+)$ 0.6 |
| P_V^- | Dm_PhVM $D_{m,0}(P_V^-)$ 0 | Em_PhVM $E_m(P_V^-)$ 5 | e_PhVM0 $e(0,-)(P_V^-)$ 0.7 | Sin datos | |

Los valores de la energía de migración para el fósforo en otros materiales que no sean silicio son

| | Migración | |
|------------|---|---|
| | cm ² s ⁻¹ | eV |
| $POxide$ | Dm_PhOxide $D_{m,0}(POxide)$ 14.2 | Em_PhOxide $E_m(POxide)$ 4.86 |
| $PNitride$ | Dm_PhNitride $D_{m,0}(PNitride)$ 14.2 | Em_PhNitride $E_m(PNitride)$ 4.86 |

Interacciones con la superficie y/o intercaras

Los parámetros de superficie y/o intercaras relacionados con la emisión y recombinación del fósforo son:

Modelo en la
sec. 8.7.2,
pág. 153.

| | Emisión = P_i | Captura = P_i |
|-------------|---|--|
| Parámetros | Prefactor | Energía |
| Solubilidad | Csol0_Ph $C_{0,Sol}(Ph) = 1e + 20 \text{ cm}^{-3}$ | Esol_Ph $E_{Sol}(Ph) = 0 \text{ eV}$ |
| h | Surf_h0_Ph $h_0(Ph) = 0,715$ | Surf_Eh_Ph $E_h(Ph) = 1,75 \text{ eV}$ |
| e/a | Surf_e_a0_Ph $(e/a)_0(Ph) = 0,004$ | Surf_Ee_a_Ph $E_{(e/a)}(Ph) = -0,37 \text{ eV}$ |
| Segregación | Segreg_pref_PhOxide | Segreg_eV_PhOxide |
| Óxido | $g_{Oxi,0}(Ph) = 30$ | $E_{Oxi,g} = 0$ |
| Segregación | Segreg_pref_PhNitride | Segreg_eV_PhNitride |
| Nitruro | $g_{Nit,0}(Ph) = 30$ | $E_{Nit,g} = 0$ |
| Saturación | SurfMaxTrap_Ph | $\sigma_{max}(Ph) = 6,8e + 14 \text{ cm}^{-2}$ |

Disolución de pockets amorfos con fósforo

Modelo en la
sec. 8.4.1,
pág. 127.

En el momento de la disolución final de los pockets amorfos que contienen fósforo se intentan generar clusters de tipo P_4V . Si no hay partículas suficientes se generan P_3 o P_nV , con $2 \leq n < 4$. Las partículas finales que queden se depositan en forma de P , P_i o P_V .

► 9.7.4

El fósforo durante la recristalización

Las regiones amorfas que están recristalizando mantienen la concentración de fósforo eléctricamente activo (substitucional) con un valor máximo de $[P]_{\max}^{\text{recrys}}$. Los átomos de P que puedan sobrar son depositados en la forma inactiva P_nV_m , donde los valores máximos de n y m están dados por “Dopantes max. de P en clusters” y “Is o Vs max. en clusters de P ” respectivamente. Estos valores están recogidos en la siguiente tabla:

| Nombre | valor ddp | Valor |
|----------------------------------|-----------|-------------------------|
| $[P]_{\max}^{\text{recrys}}$ | (interno) | $1e+21 \text{ cm}^{-3}$ |
| Dopantes max. en clusters de P | (interno) | 3 |
| Max. Is o Vs en clusters de P | (interno) | 0 |

► 9.7.5

Clusters de impurezas con P

Modelo en la
sec. 8.6.2,
pág. 143.

El fósforo se aglomera con vacantes para formar clusters de impurezas. Las reacciones permitidas para estos clusters son:

- Crecimiento: por captura de V , $P_nV_m + V \rightarrow P_nV_{m+1}$, y P_V , $P_nV_m + P_V \rightarrow P_{n+1}V_{m+1}$.

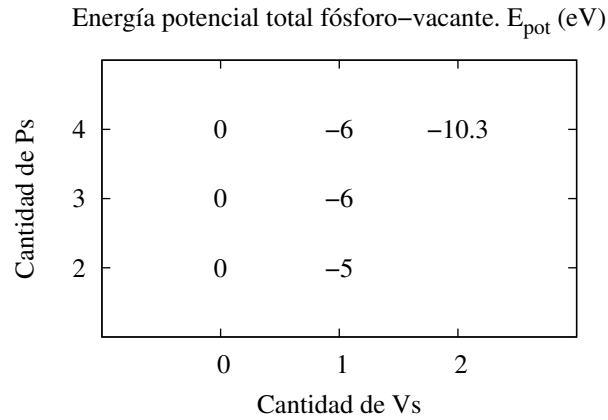


Figura 9.9: Energías potenciales para clusters de fósforo con vacantes (P_nV_m).

- Emisión: tanto de V , $P_nV_m \rightarrow P_nV_{m-1} + V$, como P_V , $P_nV_m \rightarrow P_{n-1}V_{m-1} + P_V$.
- Recombinación: de I , $P_nV_m + I \rightarrow P_nV_{m-1}$ y P_i , $P_i + P_nV_m \rightarrow P_{n+1}V_{m-1}$.
- Frank-Turnbull: emitiendo I , $P_nV_m \rightarrow P_nV_{m+1} + I$.

Los parámetros usados en DADOS como energía potencial para estos clusters se dan en la figura 9.9.

▷ 9.8

Oxígeno

El oxígeno se encuentra en las simulaciones de DADOS en forma de defectos puntuales de impurezas, o bien en clusters tipo O_nV_m .

▷ 9.8.1

Defectos puntuales con O

Los defectos puntuales que contienen oxígeno en DADOS son:

| Nombre | DADOS | Descripción | Carga |
|---------|-------|-----------------------|-------|
| O^0 | O | Oxígeno sustitucional | 0 |
| O_V^0 | OV | Oxígeno intersticial | 0 |

▷ 9.8.2

Introduciendo O en la simulación

El oxígeno se introduce en una simulación a partir de:

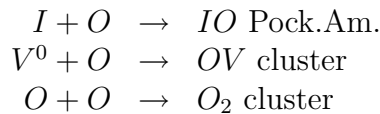
- La implantación de la especie O^0 .
- La lectura de un perfil externo de O^0 .

▷ 9.8.3

Interacciones del oxígeno**Interacciones de defectos puntuales**

Modelo en la
sec. 8.2.2,
pág. 110.

Las reacciones que se escriben a continuación están representadas de una forma general que no explicita el estado de carga del intersticial. Para el oxígeno se permiten todos los estados de carga del intersticial, pero sólo la vacante neutra V^0 . Con esto, las reacciones de defectos puntuales con oxígeno son:



El prefactor y energía de migración, energía y prefactor de enlace y el nombre con el que se accede a estos parámetros en el fichero de parámetros de DADOS

Modelo en la
sec. 8.6.1,
pág. 139.

| | Migración | | Elect. eV | Enlace | |
|-------|---------------------------------|--------------------------|--------------|---------------------------------|----------------------------|
| | cm ² s ⁻¹ | eV | | cm ² s ⁻¹ | eV |
| O | Dm_O $D_{m,0}(O)$ 0.13 | Em_O $E_m(O)$ 2.53 | Sin datos | Sin datos | |
| O_V | Dm_OV $D_{m,0}(O_V)$ 0 | Em_OV $E_m(O_V)$ 5 | Sin datos | Db_OV $D_{b,0}(O_V)$ 0.01 | Eb_OV $E_b(O_V)$ 2.2 |

Interacciones con la superficie y/o intercaras

Modelo en la
sec. 8.7.2,
pág. 153.

DADOS no implementa ninguna reacción del O con las intercaras y/o con las superficies.

Disolución de pockets amorfos con oxígeno

Modelo en la
sec. 8.4.1,
pág. 127.

Al disolverse los pockets amorfos intentan generar la mayor cantidad posible de clusters tipo O_2V con el oxígeno y vacantes que hayan atrapado. Cuando éstos no están presentes se crean clusters OV o se deposita O .

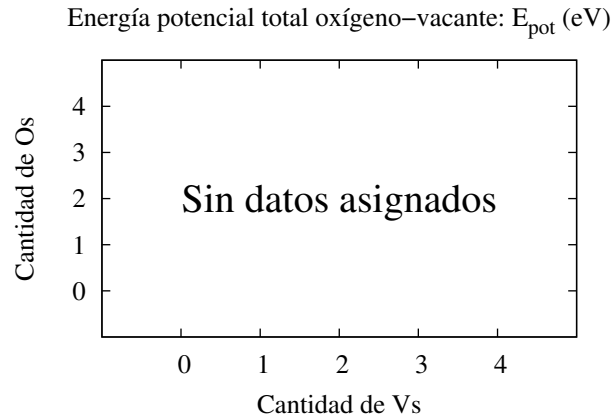


Figura 9.10: Energías potenciales para clusters de oxígeno con vacantes O_nV_m .

▷ 9.8.4

Cluster de impurezas con O

El oxígeno forma clusters de impurezas con vacantes, O_nV_m . Las reacciones Modelado en la permitidas para estos clusters son: sec. 8.6.2,

- De crecimiento, atrapando V ($O_nV_m + V \rightarrow O_nV_{m+1}$).
- De emisión de V ($O_nV_m \rightarrow O_nV_{m-1} + V$), y O_V ($O_nV_m \rightarrow O_{n-1}V_{m-1}$).
- Frank-Turnbull (emisión de I): ($O_nV_m \rightarrow O_nV_{m+1} + I$).
- Recombinación de Is ($O_nV_m + I \rightarrow O_nV_{m-1}$).

pág. 143.

Los valores de energía potencial utilizados para estos clusters pueden verse representados en la figura 9.10.

▷ 9.9

Indio

El indio se encuentra en DADOS implementado en defectos puntuales con esta impureza. No hay clusters de indio (aún).

▷ 9.9.1

Defectos puntuales con In

Los defectos puntuales con indio en DADOS son:

| Nombre | DADOS | Descripción | Carga |
|----------|-------|---------------------|-------|
| In^- | In | Indio sustitucional | -1 |
| In_i^0 | Ini | Indio intersticial | 0 |
| In_i^- | IniP | Indio intersticial | -1 |
| In_V^0 | InV | Indio vacante | 0 |
| In_V^- | InVM | Indio vacante | -1 |

▷ 9.9.2

Introduciendo indio en la simulación

Se puede introducir indio en la simulación con los siguientes procesos:

- Implantación de la especie In_i^0 .
- Lectura de un perfil externo de In^- .

▷ 9.9.3

Interacciones del indio

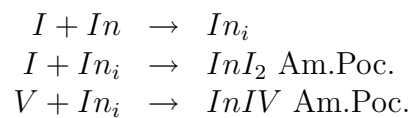
Interacciones de defectos puntuales

Modelo en la
sec. 8.2.2,
pág. 110.

Las reacciones que se detallan a continuación han sido escritas de una forma general que no explicita el estado de carga. Para ver los posibles estados de carga hay que tomar todos los posibles y eliminar las reacciones entre especies repulsivas, con igual carga (+, +) o (-, -). Por último, el estado de carga de la especie creada dependerá del nivel de Fermi local donde se encuentre la partícula. De esta manera, para interpretar la reacción $I + In \rightarrow In_i$ hemos de:

- Considerar todos los estados posibles del intersticial (I^0, I^-, I^+) y del indio sustitucional (In^-).
- Permitir sólo las reacciones no repulsivas, es decir $I^+ + In^-$ y $I^0 + In^-$.
- El resultado de estas reacciones será In_i con un estado de carga, (In_i^+ o In_i^0) que dependerá del nivel de Fermi local.

Las interacciones del indio como defecto puntual son:



En la siguiente tabla se detallan los valores para el prefactor y energía de migración, nivel energético y el nombre que cada uno de estos valores tiene en el fichero de parámetros del DADOS.

Modelo en la
sec. 8.6.1,
pág. 139.

Departamento de Electricidad y Electrónica. Tesis de Ignacio Martín Bragado.

| | Migración | | Elect. eV | Enlace | |
|----------|--------------------------------------|---------------------------------|-------------------------------------|--|---------------------------------|
| | cm ² s ⁻¹ | eV | | cm ² s ⁻¹ | eV |
| In | Dm_In $D_{m,0}(In)$ 0 | Em_In $E_m(In)$ 5 | Sin datos | Sin datos | |
| In_i | Dm_Ini $D_{m,0}(In_i)$ 1.6e-06 | Em_Ini $E_m(In_i)$ 0.9 | Sin datos | Sin datos | |
| In_i^- | Dm_IniM $D_{m,0}(In_i^-)$ 2 | Em_IniM $E_m(In_i^-)$ 2.2 | e_IniM0 $e(0, -)(In_i^-)$ 0.6 | Db_IniM $D_{b,0}(In_i^-)$ 0 | Eb_IniM $E_b(In_i^-)$ 1.6 |
| In_V | Dm_InV $D_{m,0}(In_V)$ 0 | Em_InV $E_m(In_V)$ 5 | Sin datos | Sin datos | |
| In_V^- | Dm_InVM $D_{m,0}(In_V^-)$ 0 | Em_InVM $E_m(In_V^-)$ 5 | e_InVM0 $e(0, -)(In_V^-)$ 0.7 | Db_InVM $D_{b,0}(In_V^-)$ 0.0015 | Eb_InVM $E_b(In_V^-)$ 1.6 |

Interacciones con superficie y/o intercaras

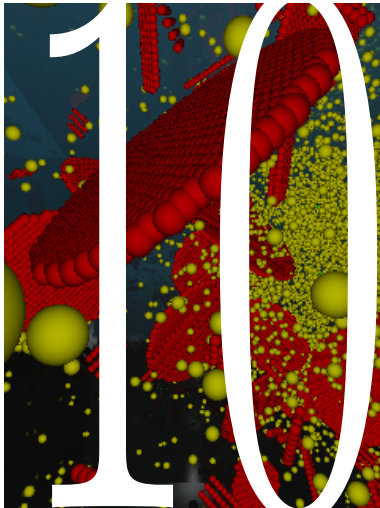
No se ha definido ninguna interacción con la superficie y/o intercaras.

Modelo en la
sec. 8.7.2,
pág. 153.

Disolución de pockets amorfos con indio

Los pockets amorfos, al disolverse, liberan In_V cuando también tienen vacantes, o In en caso contrario.

Modelo en la
sec. 8.4.1,
pág. 127.



Comprobación de los modelos de DADOS

| | |
|---|-----|
| 10.1. Introducción . . . | 219 |
| 10.2. Concentraciones y difusividades en equilibrio | 219 |
| 10.3. Defectos extendidos | 224 |
| 10.4. El modelo de dañado | 229 |
| 10.5. Activación y desactivación de dopantes | 236 |
| 10.6. Intercaras entre materiales | 241 |
| 10.7. Modelo de carga . | 244 |
| 10.8. Conclusiones . . . | 248 |

Este capítulo es una completa recopilación de ejemplos que pueden simularse con DADOS, utilizando el mismo conjunto de parámetros para todos ellos y, de esa forma, comprobar la validez y precisión de los modelos incluidos en este momento en el simulador.

▷ 10.1

Introducción

En este capítulo se van a exponer una serie de ejemplos relacionados con aspectos muy diferentes del simulador DADOS. Esta recopilación de simulaciones, junto con la comparación con resultados experimentales o teóricos, genera un *test* para comprobar como de precisos y realistas son los modelos implementados en DADOS, así como su capacidad predictiva frente a verdaderas situaciones experimentales.

El código y, por tanto, los modelos usados en esta recopilación, es el mismo para todos, y además, excepto cuando se diga lo contrario, todos los modelos están funcionando a la vez (aunque muchos de ellos pueden no ser relevantes para ciertas simulaciones). A su vez, los parámetros usados (es decir, el fichero `datos.ddp`) son los mismos para todos.

Como DADOS es un simulador kMC para la difusión de defectos en silicio, necesita un programa externo que simule los procesos de implantación. Este programa externo es MARLOWE (Robinson and Torrens, 1974), que genera las posiciones y partículas para las cascadas que, posteriormente, son leídas por DADOS. Este simulador de implantación BCA no tiene, en principio, ningún modelo de acumulación de dañado, y se limita a ofrecer la posibilidad de realizar las implantaciones en silicio cristalino o amorfo, opción que debe ser pasada al simulador.

▷ 10.2

Concentraciones y difusividades en equilibrio

▷ 10.2.1

I y V

Bajo este apartado se pretende comprobar las concentraciones en equilibrio de intersticiales y vacantes. En DADOS, la superficie es la encargada de mantener la proporción adecuada de estas partículas, emitiendo y capturando defectos puntuales. Además, en equilibrio, la concentración de *I*s y *V*s tiene que ser homogénea.

Las instrucciones necesarias para realizar los ejemplos son:

- Se crea una celdilla de simulación de $10 \times 25 \times 25 \text{ nm}^3$.
- Recocido a 600 °C durante $1 \times 10^{10} \text{ s}$.

En la figura 10.1 se muestran los resultados de la simulación (símbolos), las

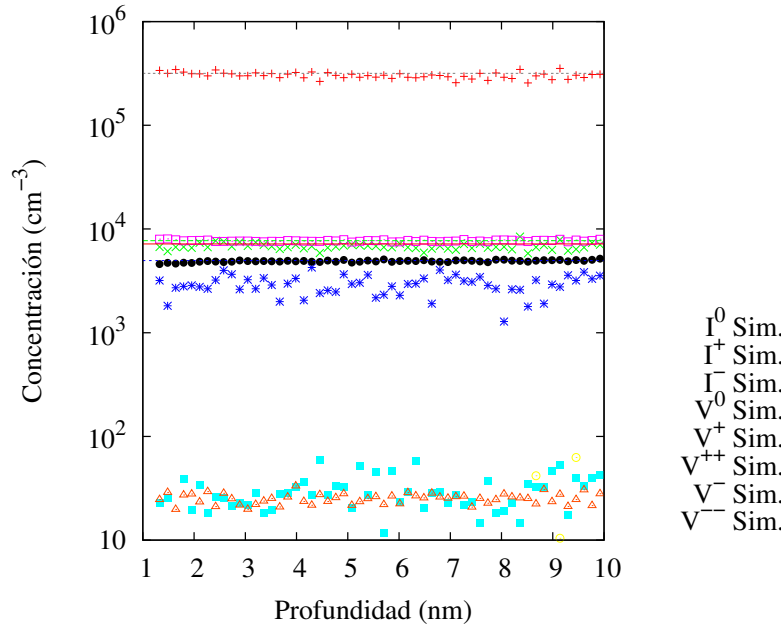


Figura 10.1: Concentración de intersticiales y vacantes en equilibrio a 600°C . Líneas: valores teóricos. Símbolos: resultados simulados.

líneas representan los resultados teóricos. Efectivamente, la concentración es constante y está en los valores correctos.

A su vez, para comprobar la validez de los productos $D(I)[I]$ y $D(V)[V]$ se realizó la siguiente simulación, cuyos comandos son:

- Se crea una celda de $10 \times 50 \times 50 \text{ nm}^3$.
- Recocido durante 10^7 s a 700°C .
- Recocido durante 10^5 s a 800°C .
- Recocido 1500 s a 900°C .
- Recocido 50 s a 1000°C .
- Recocido $0,5 \text{ s}$ a 1100°C .

La figura 10.2 representa los productos $D(I)[I]$ y $D(V)[V]$ frente al tiempo. Las líneas son los resultados de la simulación y los símbolos los valores experimentales, tomados de la referencia (Bracht et al., 1998).

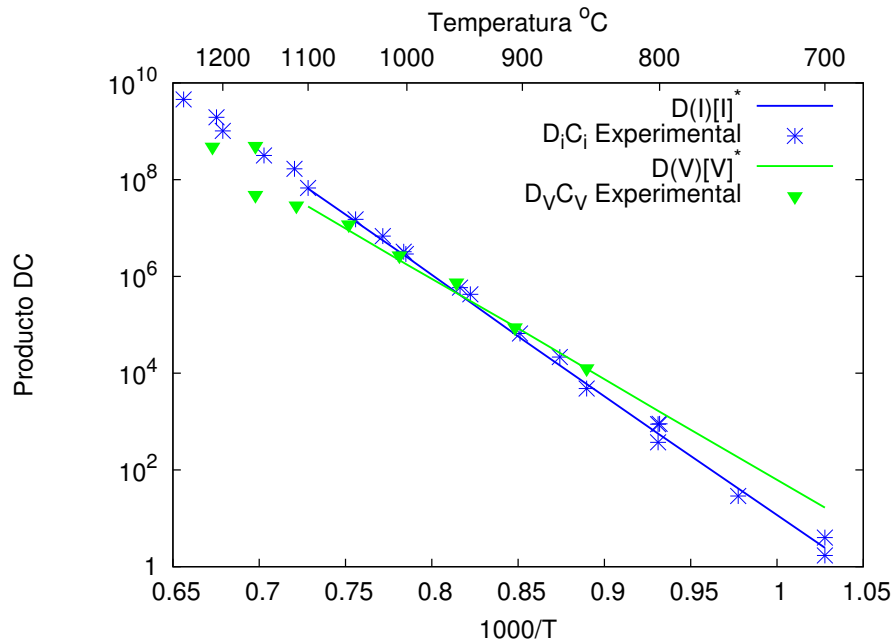


Figura 10.2: Productos $D(V)[V]$ y $D(I)[I]$ a diferentes temperaturas. Símbolos: datos experimentales (Bracht et al., 1995; Cowern et al., 1999b; Giese et al., 2000). Líneas: simulaciones.

▷ 10.2.2

Dopantes

La difusividad de un dopante depende de la naturaleza propia del dopante y del nivel de Fermi local. En principio, se pueden establecer dos niveles de difusividad según sea este nivel de Fermi, conocidos como condiciones intrínseca y extrínseca. Para el caso intrínseco, el nivel de Fermi se corresponde con una concentración de electrones en equilibrio n_i , presente cuando no hay dopantes.

Hablando de manera estricta, sería imposible comprobar cómo difunden los dopantes bajo condiciones intrínsecas, puesto que en cuanto introdujéramos un dopante habría una pequeña modificación del nivel de Fermi y, por tanto, de la difusividad pero, en la práctica, para concentraciones de dopantes bastante más bajas que el nivel de Fermi a una temperatura dada, el semiconductor se comporta como un semiconductor intrínseco.

En cualquier caso, hemos utilizado un aproximación distinta para resolver el problema de la comprobación de la difusividad intrínseca, consistente en poner el nivel de Fermi de forma artificial como el de equilibrio, y correr así las simulaciones. Esta aproximación tiene su interés únicamente con el propósito de comprobar el programa y comparar con situaciones extrínsecas.

La figura 10.3 muestra una simulación de difusión de arsénico en régimen intrínseco. Los símbolos están calculados con un programa auxiliar de difusión

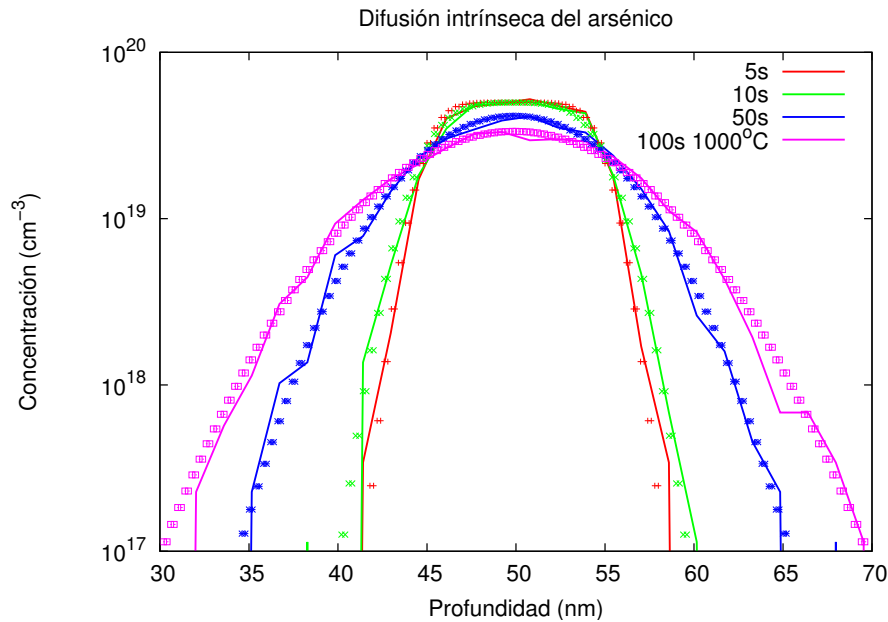


Figura 10.3: Difusión de una punta de arsénico (régimen intrínseco). 100 s. a 1000 °C. Símbolos: simulador de continuo. Líneas: resultados de DADOS.

que usa una aproximación de continuo, las líneas son el resultado de la simulación con DADOS. La figura siguiente 10.4, muestra una punta de boro difundiendo de manera análoga bajo régimen intrínseco. También aquí los símbolos han sido calculados con un programa de continuo auxiliar, y las líneas son las simulaciones realizadas con DADOS.

Los comandos utilizados en estas simulaciones son:

1. Arsénico

- Se crea una celdilla de $100 \times 25 \times 25 \text{ nm}^3$.
- Se habilita el modo intrínseco: el nivel de Fermi es igualado al intrínseco.
- Se deshabilita la formación de clusters de impurezas.
- Se lee un perfil de arsénico: concentración de $5 \times 10^{19} \text{ cm}^{-3}$ desde 45 hasta 55 nm.
- Recocido: 100 s. a 1000 °C.

2. Boro

- Se crea una celdilla de $100 \times 25 \times 25 \text{ nm}^3$.
- Se habilita el modo intrínseco: el nivel de Fermi es igualado al intrínseco.

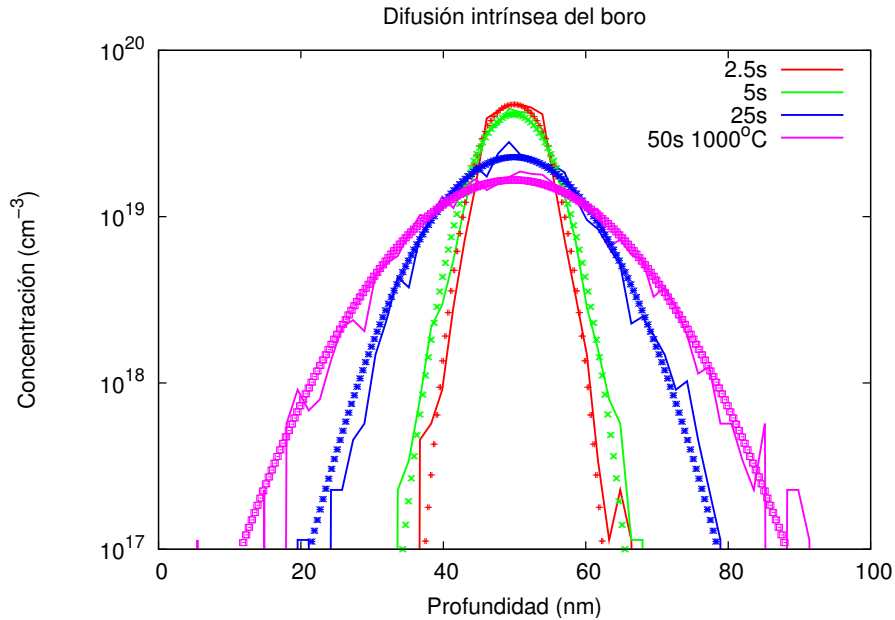


Figura 10.4: Difusión intrínseca del boro. 50 s. a 1000°C . Símbolos: resultados simulados con un simulador de continuo. Líneas: valores generados con DADOS.

- Se deshabilita la formación de clusters de impurezas.
- Se lee un perfil de boro: concentración de $5 \times 10^{19} \text{ cm}^{-3}$ desde 45 hasta 55 nm.
- Recocido: 50 s a 1000°C .

Una comprobación extra para el boro se pudo lograr gracias a la referencia (Covern et al., 1991), que contiene datos experimentales (SIMS) de la difusión de una punta de boro. La concentración máxima de boro en este caso es lo suficientemente baja como para que no haya clusterización, y para asegurar un régimen intrínseco, y la temperatura lo suficientemente alta para evitar los saltos largos. Los resultados de la simulación, junto con los datos experimentales, están expuestos en la figura 10.5.

Los comandos empleados son:

- Se crea una celda de $120 \times 200 \times 200 \text{ nm}^3$.
- Se lee un perfil de boro (tomado de la referencia (Covern et al., 1991)).
- Recocido: 300 s a 900°C .

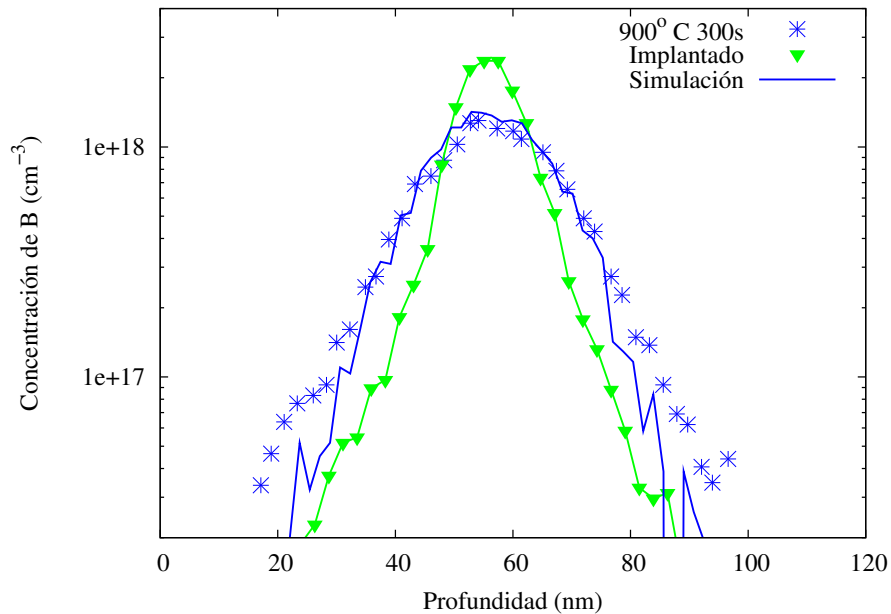


Figura 10.5: Difusión de una punta de boro a 900°C durante 300s. Símbolos: valores experimentales antes y después del recocido (Cowern et al., 1991). Líneas: simulación de DADOS.

▷ 10.3

Defectos extendidos

▷ 10.3.1

Clusters de intersticiales y evolución temporal de la supersaturación de I_s

Para comprobar cómo de bien están funcionando los clusters de I_s , responsables en gran parte de la evolución temporal de $S(I)$, se comparó la evolución de ésta en el tiempo con los resultados experimentales obtenidos por (Cowern et al., 1999c).

Las simulaciones se realizaron con las siguientes instrucciones:

1. 600°C :
 - Se crea una celdilla de $350 \times 50 \times 50 \text{ nm}^3$.
 - Se implanta Si en Si. Dosis: $2 \times 10^{13} \text{ cm}^{-2}$. Energía: 40 keV.
 - Rampa de recocido: desde 26°C hasta 600°C en 6 s.
 - Recocido: 100000 s. a 600°C .
2. 700°C :

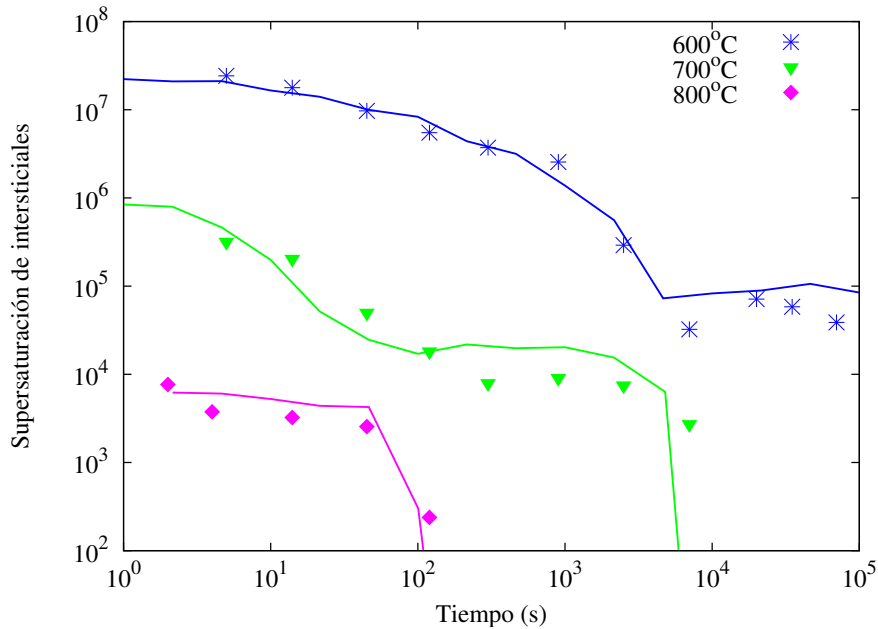


Figura 10.6: Evolución de la supersaturación de I_s con el tiempo. Se ha realizado una implantación de Si en Si de $2 \times 10^{13} \text{ cm}^{-2}$, 40 keV seguida de 100000 s. a 600 °C, 10000 s. a 700 °C o 2000 s a 800 °C. Símbolos: resultados experimentales (Cowern et al., 1999c). Líneas: simulaciones.

- Se crea una celdilla de $350 \times 50 \times 50 \text{ nm}^3$.
- Se implanta Si en Si. Dosis: $2 \times 10^{13} \text{ cm}^{-2}$. Energía: 40 keV.
- Rampa de recocido: desde 26 °C hasta 700 °C en 7 s.
- Recocido: 10000 s. a 700 °C.

3. 800 °C:

- Se crea una celdilla de $350 \times 50 \times 50 \text{ nm}^3$.
- Se implanta Si en Si. Dosis: $2 \times 10^{13} \text{ cm}^{-2}$. Energía: 40 keV.
- Rampa de recocido: desde 26 °C hasta 800 °C en 7.5 s.
- Recocido: 2000 s. a 800 °C.

La figura 10.6 muestra los resultados experimentales (símbolos) tomados de (Cowern et al., 1999c) frente a los simulados (líneas).

▷ 10.3.2

Disolución de los {311}s

Las medidas por microscopía electrónica tomadas en la referencia (Stolk et al., 1997), que muestran la evolución con el tiempo de la densidad de defectos {311} en

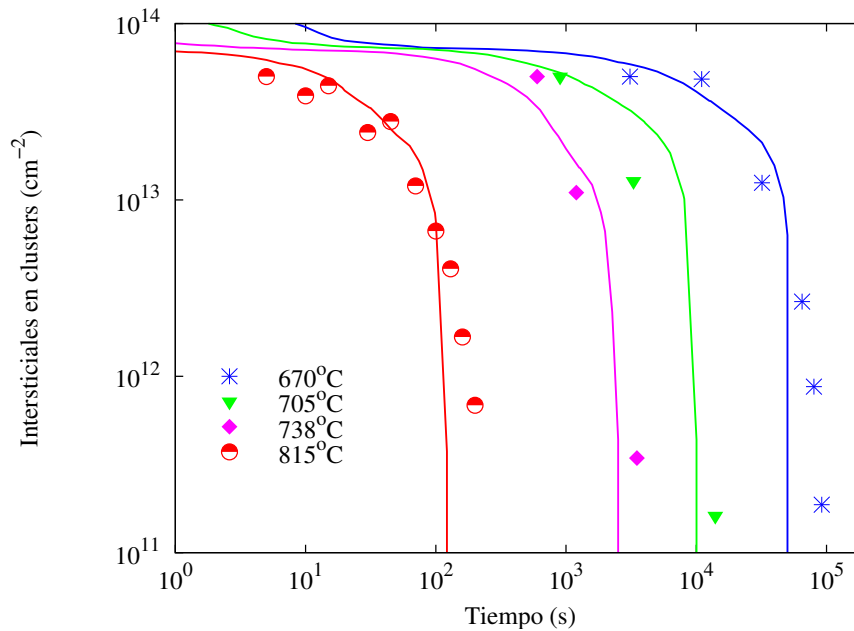


Figura 10.7: Disolución de los defectos tipo $\{311\}$. Se hace una implantación de Si en Si de $5 \times 10^{13} \text{ cm}^{-2}$, 40 keV y unos recocidos posteriores de 200000 s a 670°C , 100000 s a 705°C , 5000 s a 738°C y 400 s a 815°C . Líneas: simulación. Símbolos: resultados experimentales tomados de (Stolk et al., 1997).

función de la temperatura son una buena referencia para comprobar si el modelo de formación y posterior disolución de los $\{311\}$ está funcionando adecuadamente. Concretamente, la figura 10.7 muestra los puntos experimentales (símbolos) frente a los resultados de la simulación (líneas).

Las simulaciones fueron hechas de acuerdo a las siguientes instrucciones:

1. 670°C

- Se crea una celdilla de $350 \times 50 \times 50 \text{ nm}^3$.
- Se implanta Si en Si. Dosis: $5 \times 10^{13} \text{ cm}^{-2}$. Energía: 40 keV.
- Rampa de recocido: desde 26°C hasta 670°C en 10 s.
- Recocido: 200000 s a 670°C .

2. 705°C

- Se crea una celdilla de $350 \times 50 \times 50 \text{ nm}^3$.
- Se implanta Si en Si. Dosis: $5 \times 10^{13} \text{ cm}^{-2}$. Energía: 40 keV.
- Rampa de recocido: desde 26°C hasta 705°C en 7 s.
- Recocido: 100000 s a 705°C .

3. 738 °C

- Se crea una celdilla de $350 \times 50 \times 50 \text{ nm}^3$.
- Se implanta Si en Si. Dosis: $5 \times 10^{13} \text{ cm}^{-2}$. Energía: 40 keV.
- Rampa de recocido: desde 26 °C hasta 738 °C en 8 s.
- Recocido: 5000 s a 738 °C.

4. 815 °C

- Se crea una celdilla de $350 \times 50 \times 50 \text{ nm}^3$.
- Se implanta Si en Si. Dosis: $5 \times 10^{13} \text{ cm}^{-2}$. Energía: 40 keV.
- Rampa de recocido: desde 26 °C hasta 815 °C en 8 s.
- Recocido: 400 s a 815 °C.

▷ 10.3.3

Bucles de dislocación

Para comprobar cómo está funcionando la formación de los bucles de dislocación, se puede acudir a las medidas experimentales por microscopía electrónica realizadas por (Li and Jones, 1998) para una implantación de Si en Si de 100 keV, $2 \times 10^{14} \text{ cm}^{-2}$ y un recocido posterior a 800 °C.

Así pues, la figura 10.8 muestra cómo los bucles de dislocación van evolucionando con el tiempo. Las líneas representan los valores simulados, frente a los símbolos que son los valores experimentales. Hemos necesitado dividir los resultados simulados por un factor, debido a que en el experimento real no se pueden ver los clusters hasta que éstos no rebasan cierto tamaño y, por tanto, en la simulación siempre habrá más clusters de los que se podrán ver en el experimento.

Las instrucciones dadas a la simulación son:

- Se crea una celdilla de $1000 \times 25 \times 25 \text{ nm}^3$.
- Se implanta Si en Si. Dosis: $2 \times 10^{14} \text{ cm}^{-2}$. Energía: 100 keV.
- Rampa de recocido: desde 27 °C hasta 800 °C en 100 s.
- Recocido: 40 min. a 800 °C.

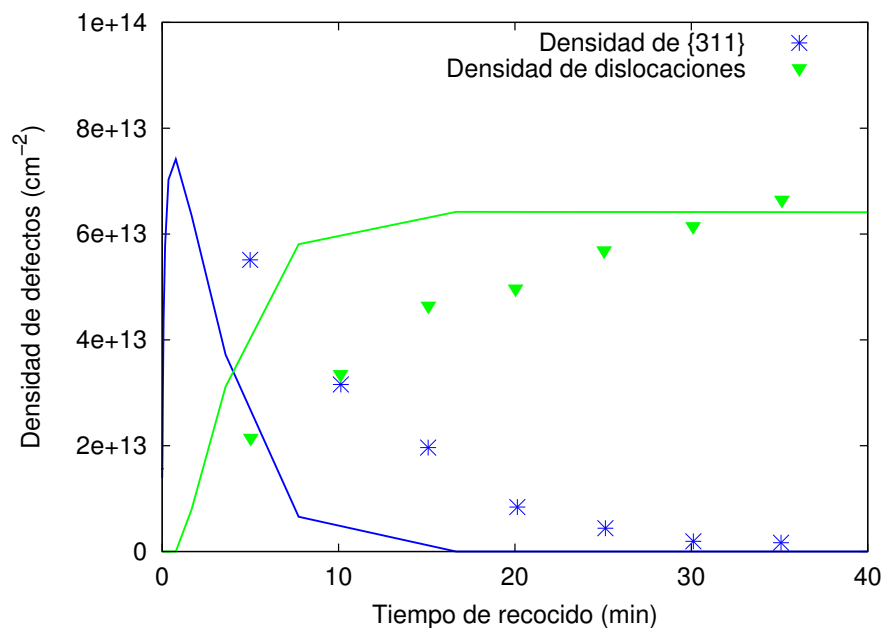


Figura 10.8: Resultados experimentales y simulaciones realizadas para una implantación de Si en Si de 100 keV, $2 \cdot 10^{14} \text{ cm}^{-2}$ y un recocido posterior a 800 °C. Se muestra la evolución de los {311}s y los defectos en función del tiempo, donde el decaimiento de los {311}s se debe a la transformación y posterior absorción por parte de los bucles de dislocación. Líneas: simulación. Símbolos: datos experimentales divididos por un factor de corrección, como se explica en el texto, tomados de (Li and Jones, 1998).

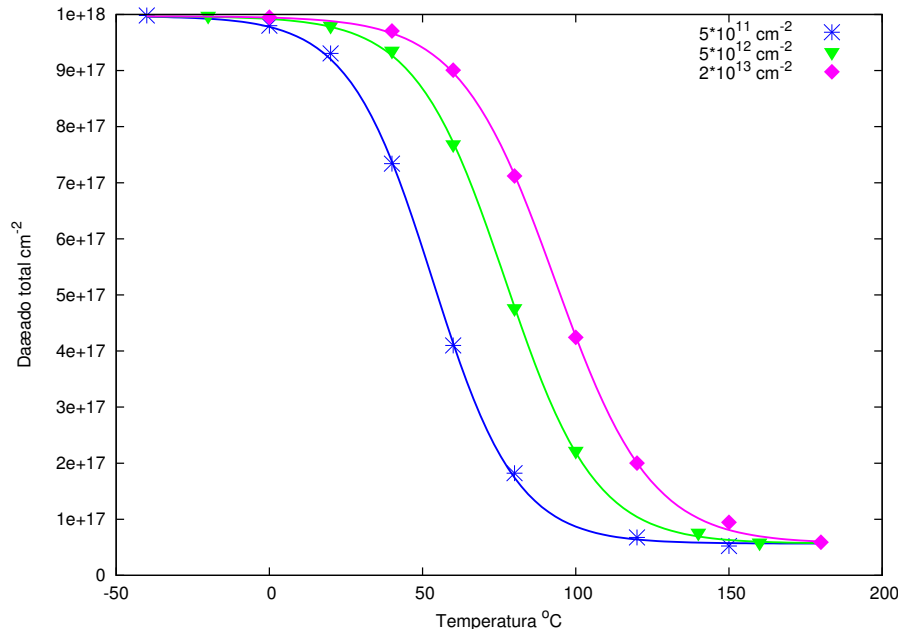


Figura 10.9: Concentración máxima de pares IV en función de la temperatura para distintas velocidades de implantación (en $\text{cm}^{-2}\text{s}^{-1}$) de Si en Si, 80 keV de energía y una dosis de 10^{15}cm^{-2} . Líneas: ajustes a la ecuación 10.1. Símbolos: resultados de DADOS.

▷ 10.4

El modelo de dañado

▷ 10.4.1

Acumulación de dañado

La sección 8.4.2 propone un modelo casi-atómico para describir la evolución del dañado que se genera por implantación en el Si, que abarca desde la formación de defectos puntuales aislados, hasta la formación de capas continuas de material amorfo. Posteriormente, a partir de estas estructuras, se pueden formar otras aún más complejas (defectos extendidos, etc.). Ahora bien ¿es capaz el modelo de representar las abruptas transiciones amorfo/cristal experimentalmente caracterizadas por (Goldberg et al., 1995)?

La figura 10.9 muestra la concentración máxima de pares IV en función de la temperatura (-20 hasta 680 °C), para diferentes velocidades de implantación (5×10^{11} a $5 \times 10^{12} \text{cm}^{-2}\text{s}^{-1}$) de Si en Si, con energía 80 keV. Condiciones similares, pero implantando Ge en vez de Si, están representadas en la figura 10.10.

La acumulación de dañado que se muestra en las gráficas anteriormente men-

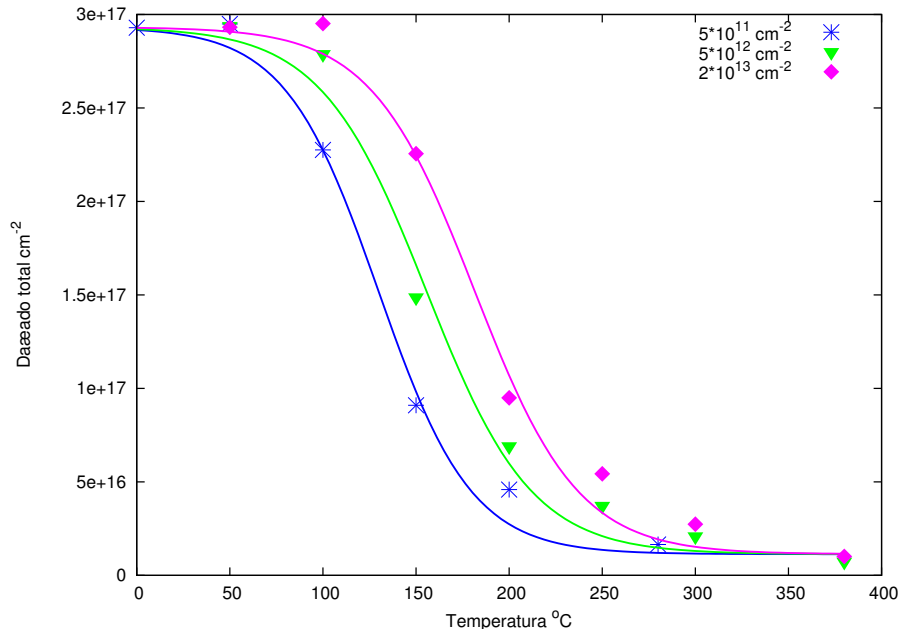


Figura 10.10: Concentración máxima de pares IV en función de la temperatura para distintas velocidades de implantación (en $\text{cm}^{-2}\text{s}^{-1}$) de Ge en Si, 80 keV de energía y una dosis de 10^{15}cm^{-2} . Líneas: ajustes a la ecuación 10.1. Símbolos: resultados de DADOS.

cionadas se puede parametrizar con la siguiente función:

$$IV_{\text{damage}} = a + b \frac{1}{1 + \exp[c \times (T - T_C)]}, \quad (10.1)$$

donde T_C es la temperatura de la transición cristal-amorfo.

La figura 10.11 muestra las temperaturas de transición amorfo-cristal en función de la velocidad de implantación iónica para implantaciones de Si y Ge, de 80 keV de energía. Líneas: simulaciones de DADOS. Símbolos: resultados experimentales tomados de (Goldberg et al., 1995).

Las instrucciones de simulación para cada temperatura (T), y velocidad de implantación son:

- Se crea una celdilla de $200 \times 20 \times 20 \text{nm}^3$.
- Se implanta Si (o Ge) en Si. Velocidad de implantación: desde 5×10^{11} hasta $5 \times 10^{12} \text{cm}^{-2}\text{s}^{-1}$. Energía: 80 keV. Temperatura: T °C (T desde -20 hasta 240 °C).

Para la simulación de la implantación a diferentes temperaturas, el algoritmo programado es:

- El programa BCA MARLOWE genera N cascadas para la especie implantada con la energía de ion solicitada.

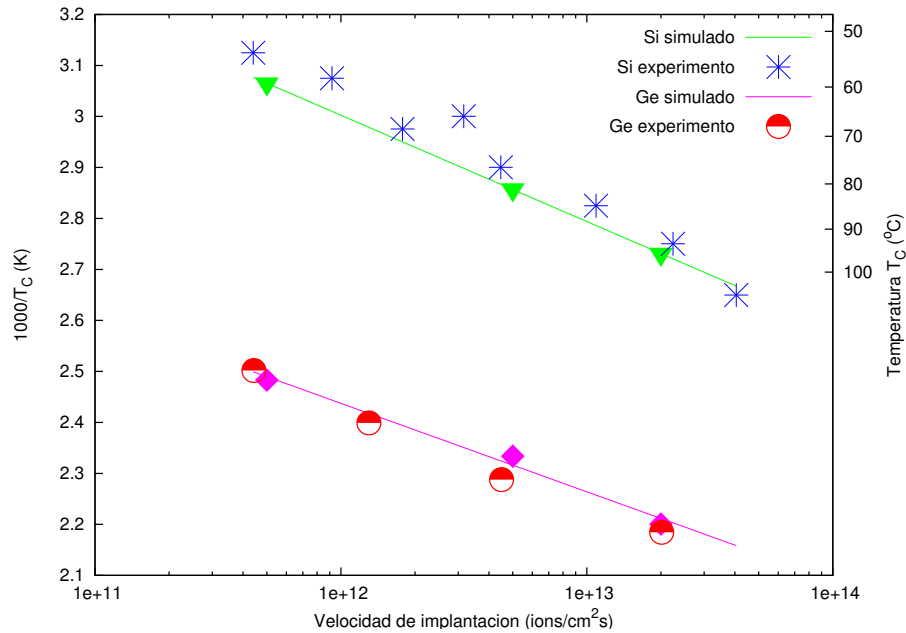


Figura 10.11: Temperatura de transición (T_C) en función de la dosis de Si y Ge implantados en Si. Líneas: simulación. Símbolos: resultados experimentales tomados de (Goldberg et al., 1995).

- Se calcula la dosis implantada por cada cascada, dependiendo de la superficie.
- Se lee cada cascada generada por el código MARLOWE.
- Se inserta la cascada en el simulador DADOS.
- Se hace un recocido a la temperatura solicitada ($T^\circ\text{C}$) durante un tiempo de $t = 1/N \times \text{dose}/\text{doserate}$ segundos.
- Se repite el proceso hasta incluir las N cascadas.

▷ 10.4.2

Amorfización y recristalización

La concentración de dañado límite antes de que el Si cristalino amorfice está situado en $1,5 \times 10^{22} \text{ cm}^{-3}$. Por debajo de este nivel no hay amorfización y, por tanto, tampoco habrá ningún evento de recristalización que limpie el dañado, por lo que dicho dañado tendrá que ser recocido a través de los pocket amorfos y, posteriormente tras la disolución de éstos, a través de clusters de I_s , V_s e impurezas. Un ejemplo de esto se encuentra en la figura 10.12, donde el nivel de dañado se mantuvo por debajo del umbral. Una vez por encima de este nivel las

partes amorfas de silicio van a recrystalizar durante el recocido, y el dañado va a desaparecer, excepto quizás, en las cercanías de la intercara amorfo/cristal. La figura 10.13 muestra este caso.

Los comandos de simulación que se programaron para esto son:

1. Por debajo del umbral:
 - Se crea una celdilla de $50 \times 50 \times 2 \text{ nm}^3$.
 - Se lee un perfil externo de I_s con una concentración de $5 \times 10^{21} \text{ cm}^{-3}$ entre 0 y 30 nm.
 - Se lee un perfil externo de V_s con una concentración de $5 \times 10^{21} \text{ cm}^{-3}$ entre 0 y 30 nm.
 - Recocido: 120 s a 600°C .

2. Por encima del umbral:
 - Se crea una celdilla de $50 \times 50 \times 2 \text{ nm}^3$.
 - Se lee un perfil externo de I_s con una concentración de 10^{22} cm^{-3} entre 0 y 30 nm.
 - Se lee un perfil externo de V_s con una concentración de 10^{22} cm^{-3} entre 0 y 30 nm.
 - Recocido: 120 s a 600°C .

▷ 10.4.3

Interacción del dañado con los dopantes

El recocido de las regiones amorfizadas conlleva la desaparición de todo el dañado (I_s y V_s), que es arrastrado hacia las intercaras o hacia regiones cristalinas vecinas según pasa el frente de recrystalización. Cuando sólo una porción del Si ha sido amorfizada, generalmente aquélla en la estuvo el pico de dañado de la implantación, sólo dicha región queda perfectamente limpia mientras que, otras regiones vecinas, podrían quedar con cantidades de dañado significativas que, no obstante, no bastaron para amorfizarlas. En estos casos, se puede producir una difusión anómala de las impurezas que pudieran haber quedado en la región recrystalizada, ya que las regiones dañadas empezarán a emitir defectos puntuales hacia las limpias, mientras que la región recrystalizada no tiene ningún defecto puntual para emitir, creándose un flujo de I_s o V_s en un único sentido: de la zona dañada a la recrystalizada.

A su vez, el paso del frente de recrystalización arrastra algunas impurezas hacia la superficie, por lo que la forma del perfil de éstas antes y después de la recrystalización es diferente. La figura 10.14 ilustra este fenómeno, donde se ha representado el perfil de una implantación de Arsénico en silicio preamorfizado antes y después del paso del frente de recrystalización.

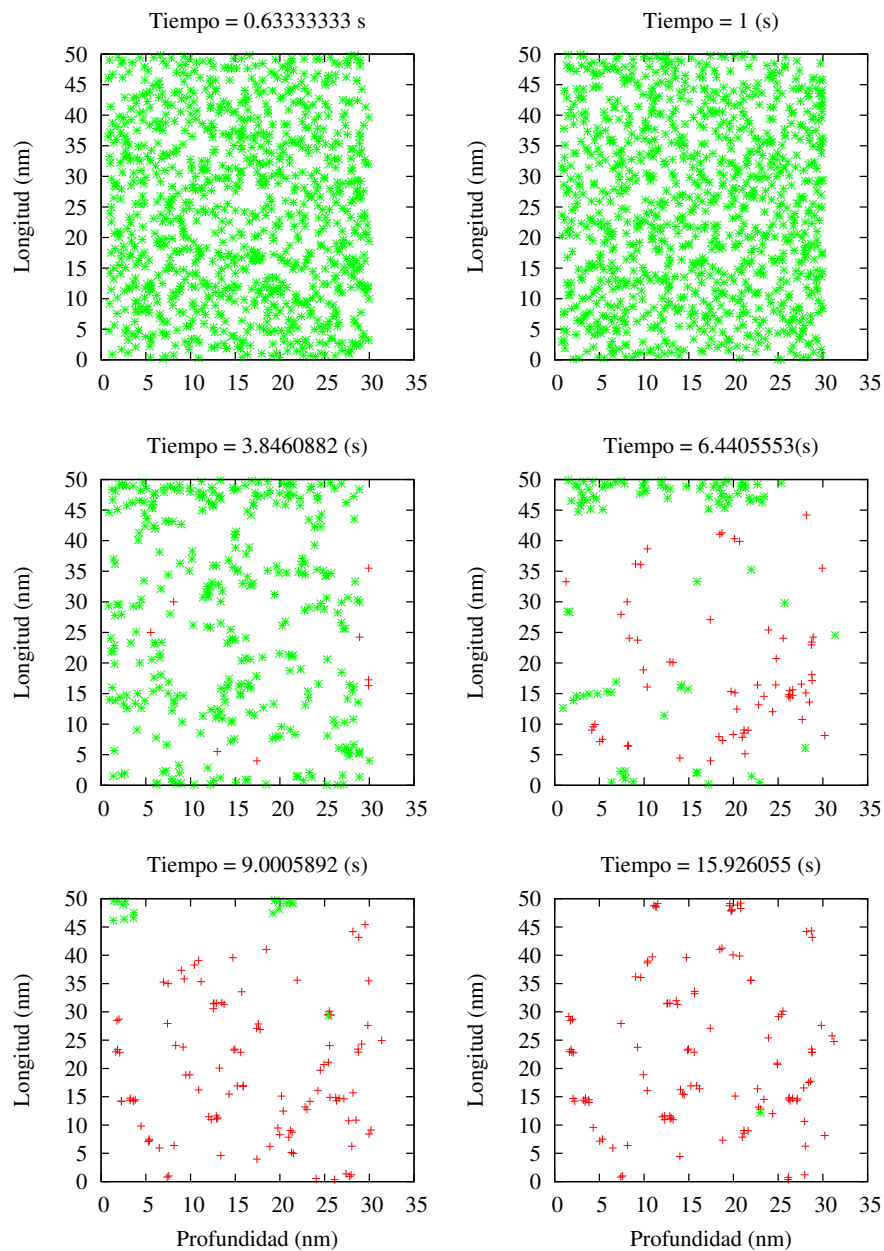


Figura 10.12: Evolución de los defectos puntuales I y V con el tiempo (proyección bidimensional en los ejes x, y). 16 s de recocido a 600°C tras leer una concentración de dañado de $1 \times 10^{22} \text{ cm}^{-3}$. Como esta concentración se encuentra por debajo del umbral de amortización, no queda una zona recristalizada limpia, sin clusters. Cruces (+): clusters. Aspas (\times): pockets amorfos.

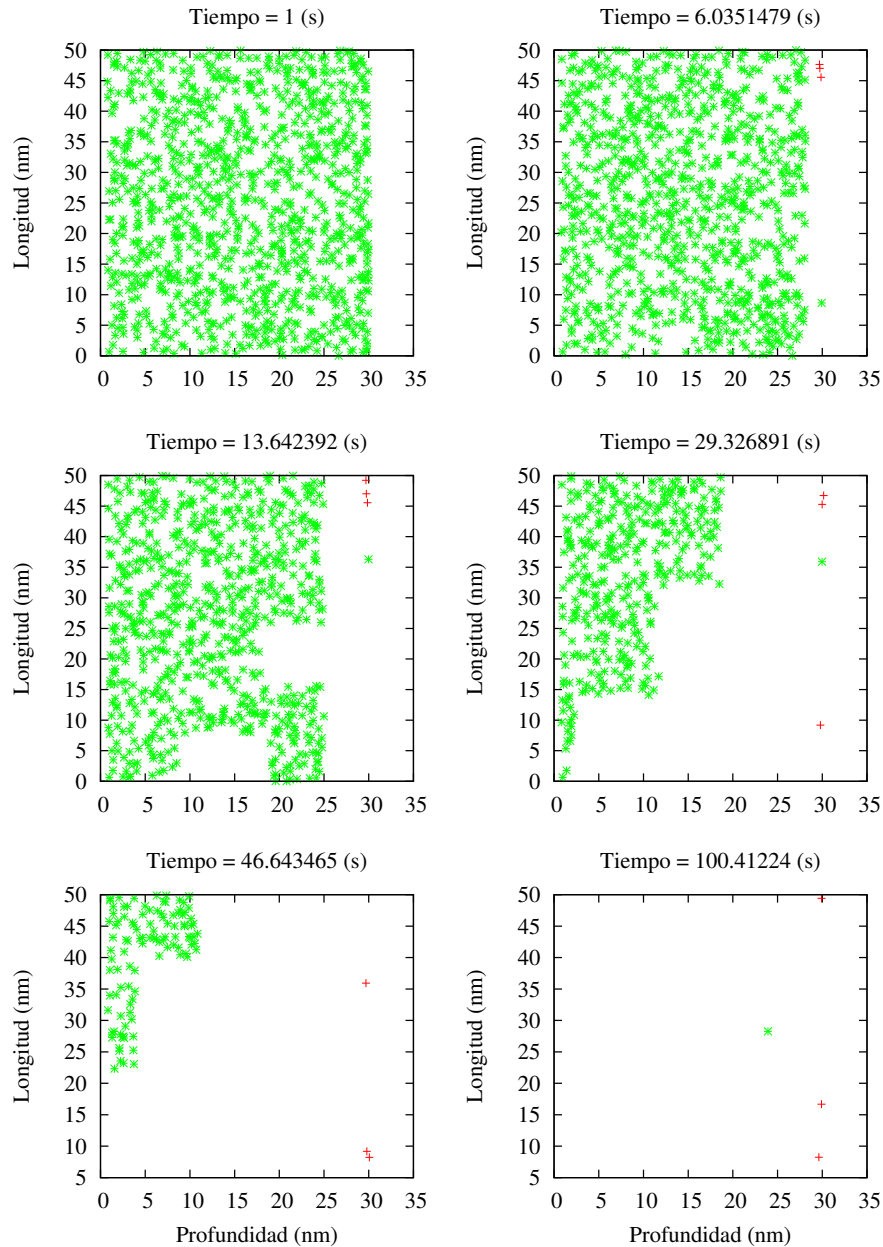


Figura 10.13: Evolución de los defectos puntuales I y V con el tiempo (proyección bi-dimensional en los ejes x,y). 100s de recocido a 600°C tras leer una concentración de dañado de $2 \times 10^{22} \text{ cm}^{-3}$. Al estar esta concentración por encima del umbral de dañado se produce un frente de recrystalización que va limpiando dicho dañado. Cruces (+): clusters. Aspas (\times): pockets amorfos.

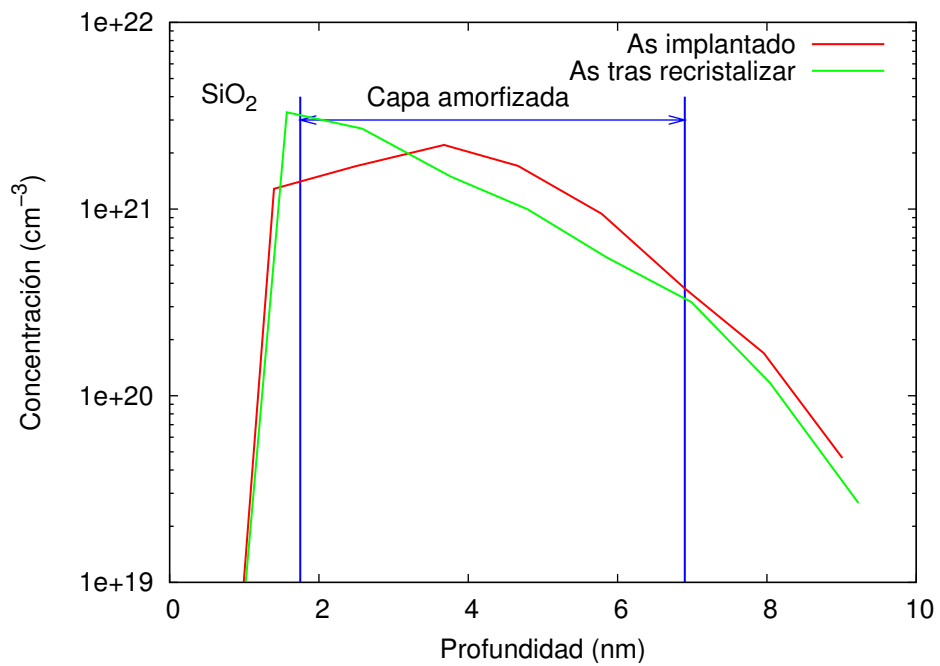


Figura 10.14: Perfil de arsénico antes y después del paso del frente de recristalización. El ejemplo corresponde a una simulación en la que se ha implantado 2 keV de arsénico, con dosis 10^{15} cm^{-2} sobre silicio preamorfizado, y a la que se sometió después a un recocido a 700°C durante 2 horas.

El script que realiza esta simulación es:

- Se crea un celdilla de simulación de $40 \times 25 \times 25 \text{ nm}^3$.
- Se implanta Ge. Dosis: $1 \times 10^{15} \text{ cm}^{-2}$. Energía: 5 keV. (Hay una amorfización parcial)
- Se implanta As. Dosis: $1 \times 10^{15} \text{ cm}^{-2}$. Energía: 2 keV.
- Rampa de recocido: desde 27°C hasta 700°C en 10s.
- Recocido: 2 horas a 700°C .

▷ 10.5

Activación y desactivación de dopantes

▷ 10.5.1

Difusión del boro y formación de clusters de boro intersticial (BIC)s

Cuando la concentración de boro e intersticiales es lo suficientemente alta se pueden formar clusters de boro intersticial. La referencia (Pelaz et al., 1997) muestra un experimento en el que una parte importante de una *spike* de boro permanece inmóvil y eléctricamente inactiva debido a la formación de estos BICs y a que su posterior disolución es algunos órdenes de magnitud más lenta que la difusión del boro activo.

La figura 10.15 muestra una *spike* de boro con una concentración de $4,5 \times 10^{19} \text{ cm}^{-3}$ tras una implantación de Si en Si con una energía de 40 keV y una dosis de $9 \times 10^{13} \text{ cm}^{-2}$. Por último se somete a la muestra completa a un recocido a 800°C .

Las instrucciones que se dieron al simulador fueron:

- Se crea una celdilla de $300 \times 50 \times 50 \text{ nm}^3$.
- Se lee un perfil externo de boro: $4,5 \times 10^{19} \text{ cm}^{-3}$ de concentración entre 100 y 120 nm.
- Se implanta Si en Si. Dosis: $9 \times 10^{13} \text{ cm}^{-2}$. Energía: 40 keV.
- Rampa de recocido: desde 27°C hasta 800°C en 11 s.
- Recocido: 2100 s a 800°C .

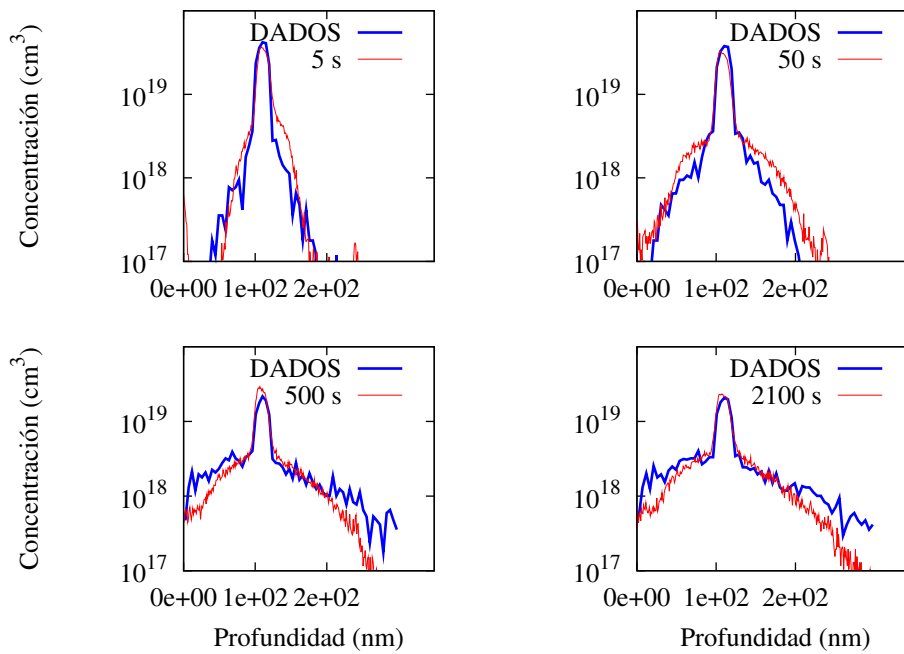


Figura 10.15: Perfiles simulados y experimentales de una *spike* de boro con concentración $4,5 \times 10^{19}$ cm⁻³ tras la implantación de Si en Si a 40 keV de energía y 9×10^{13} cm⁻² de concentración y el recido durante 5, 50, 500 y 2100 s a 800 °C. Línea continua: datos experimentales tomados de (Pelaz et al., 1997). Línea discontinua: simulaciones.

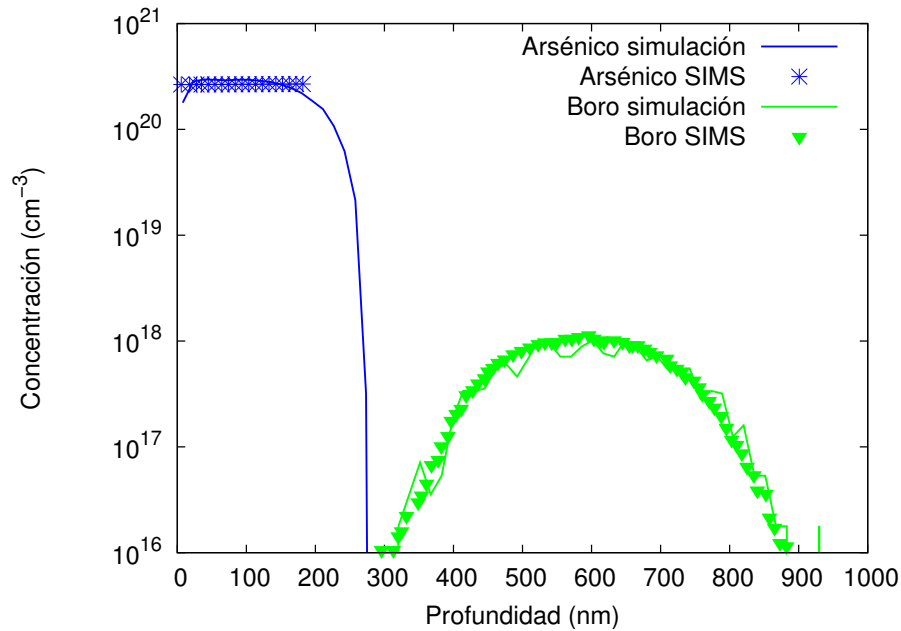


Figura 10.16: Recocido durante 480 s a 750 °C de una capa enterrada de boro (concentración $1 \times 10^{18} \text{ cm}^{-3}$). La difusión del boro es incrementada mediante arsénico implantado y derretido por láser ($4,5 \times 10^{20} \text{ cm}^{-3}$ de concentración de As desde 0 hasta 200 nm). Líneas: simulación. Símbolos: Resultados experimentales de (Rousseau et al., 1998)

► 10.5.2

Desactivación del arsénico

Se ha estudiado la desactivación eléctrica del arsénico con respecto a sus efectos en la difusión temporal extra del boro. Un estudio concreto, (Rousseau et al., 1998), proporciona evidencias convincentes sobre cómo esta difusión extra para el boro se debe a la desactivación del arsénico, a la vez que aporta detalles importantes sobre los mecanismos subyacentes al proceso. El estudio propone en particular que el arsénico se desactiva por la formación de pequeños clusters de As alrededor de una vacante, y que la formación de estos clusters conlleva la emisión de un I , aumentando la supersaturación de I s y por tanto produciendo TED.

La figura 10.16 muestra los resultados experimentales de la referencia (Rousseau et al., 1998) (símbolos), frente a los valores obtenidos mediante simulación (líneas).

Los comandos de de simulación son:

- Se crea una celdilla de $1000 \times 25 \times 25 \text{ nm}^3$.
- Se lee un perfil de arsénico: concentración $4,5 \times 10^{20} \text{ cm}^{-3}$ desde 0 hasta 200 nm.

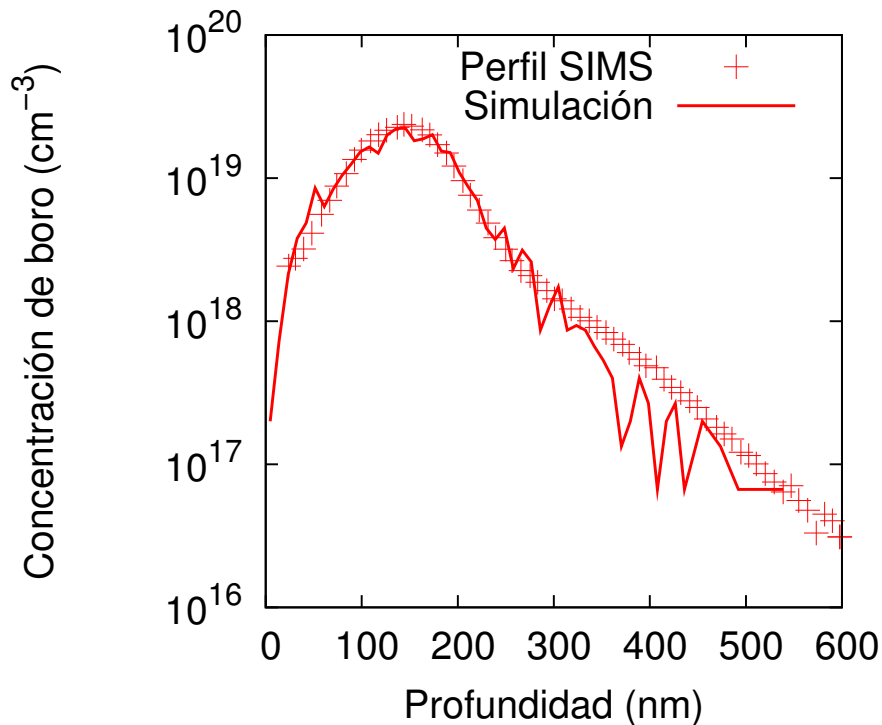


Figura 10.17: Perfiles simulados y experimentales para una implantación de boro de 40 keV de energía y $3 \times 10^{14} \text{cm}^{-2}$ de concentración, tras un recocido de 1 hora a 750°C . Los símbolos son los resultados SIMS experimentales (Huang et al., 1997), y las líneas los procedentes de la simulación.

- Se lee un perfil de boro: concentración de $1 \times 10^{18} \text{cm}^{-3}$ desde 450 hasta 750 nm.
- Rampa de recocido: desde 27°C hasta 750°C en 7 s.
- Recocido: 480 s a 750°C .

► 10.5.3

Implantación y recocido del boro

La referencia (Huang et al., 1997) propone un estudio en el cual se extrae la difusividad del boro producida por un proceso de TED, así como la cantidad de boro inmóvil a través de perfiles SIMS. La figura 10.17 muestra dichos perfiles SIMS (símbolos) junto con los resultados de la simulación realizada con DADOS.

Los comandos programados son:

- Se crea una celdilla de $600 \times 25 \times 25 \text{nm}^3$.
- Se implanta B. Dosis: $3 \times 10^{14} \text{cm}^{-2}$. Energía: 40 keV.

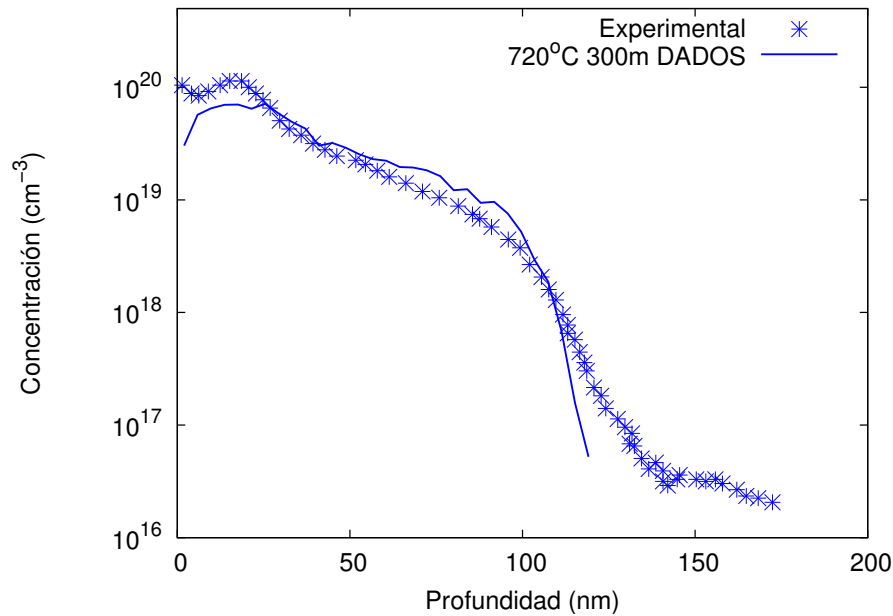


Figura 10.18: Implantación de As con $5 \times 10^{14} \text{ cm}^{-2}$ de dosis y 30 keV de energía, seguido de un recocido de 5 horas a 720 °C. Símbolos: resultados experimentales de (Kim et al., 2000). Líneas: Simulación.

- Rampa de recocido: desde 27 °C hasta 750 °C en 10 s.
- Recocido: 1 h. a 750 °C.

▷ 10.5.4

Implantación y recocido del arsénico

La referencia (Kim et al., 2000) contiene algunos perfiles extraídos mediante SIMS de obleas de silicio que han sido implantadas con una dosis de As de $5 \times 10^{14} \text{ cm}^{-2}$ y posteriormente recocidas a 720 °C durante 300 min.

En la figura 10.18 se encuentran los resultados experimentales (símbolos) así como los de la simulación (líneas) para su comparación. El "script" que produjo la simulación es:

- Se crea una celdilla de $250 \times 25 \times 25 \text{ nm}^3$.
- Se implanta arsénico. Dosis: $5 \times 10^{14} \text{ cm}^{-2}$. Energía: 30 keV.
- Rampa de recocido: Desde 27 °C hasta 720 °C en 10 s.
- Recocido: 5 h. a 720 °C.

En este experimento y simulación ha tenido lugar una amorfización hasta unos 50 nm, recristalización, depósito de una parte de As activa y otra en complejos de AsV de distintos tamaños, barrido con el frente de recristalización, captura en las intercaras, segregación al óxido y una evolución de los $\{311\}$ s a bucles de dislocación que detiene prácticamente la TED.

▷ 10.6

Intercaras entre materiales

▷ 10.6.1

Longitud de recombinación y TED del boro

En la referencia (Covern et al., 1999a) hay algunos experimentos en los que se evalúa la dependencia de la difusión temporal extra del boro (TED) durante un recocido rápido con la distancia de la superficie a la que están presentes defectos tipo *end of range* (básicamente bucles de dislocación). De estos experimentos se saca la conclusión de que la superficie libre actúa como un sumidero para los intersticiales durante la TED.

Los defectos tipo *end of range* (EOR) se crean a dos distancias diferentes de la superficie, concretamente 80 y 180 nm. Estos defectos EOR generan una supersaturación más o menos constante que va difundiendo el boro implantado. Lo que se observa es que, cuando los defectos EOR están más cerca de la superficie, la TED es mayor que cuando están lejos, mostrando de esta manera que la superficie actúa como sumidero de intersticiales.

La figura 10.19 muestra los perfiles SIMS experimentales (símbolos) frente a los resultados de la simulación (líneas).

Las instrucciones que siguió el simulador fueron:

1. $d = 80$ nm:
 - Se crea una celdilla de $300 \times 25 \times 25$ nm³.
 - Se lee un perfil de I_s : concentración de 1×10^{20} cm³ desde 80 hasta 300 nm.
 - Recocido: 0.1 s. a 900 °C (para crear los defectos EOR).
 - Se lee un perfil externo de B similar al que produciría una implantación de 3 keV con dosis 1×10^{14} cm⁻².
 - Recocido: 1 s a 900 °C.
2. $d = 180$ nm:
 - Se crea una celdilla de $300 \times 25 \times 25$ nm³.

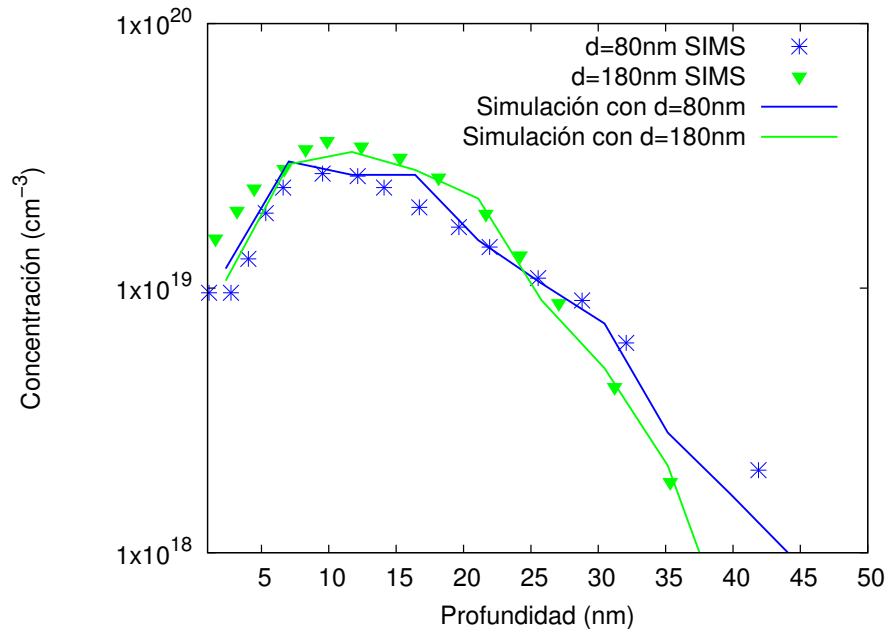


Figura 10.19: TED tras la implantación de B a 3 keV de energía y un recocido de 1 s a 900 °C. El boro difunde según la distancia de los defectos EOR creados previamente a 80 o 180 nm. Símbolos: valores experimentales de (Cowern et al., 1999a). Líneas: simulaciones.

- Se lee un perfil externo de I_s : concentración de $1 \times 10^{20} \text{ cm}^3$ desde 180 hasta 300 nm.
- Recocido: 0.1 s. a 900 °C (para crear los defectos EOR).
- Se lee un perfil externo de B, similar al que produciría una implantación de 3 keV con dosis $1 \times 10^{14} \text{ cm}^{-2}$.
- Recocido: 1 s a 900 °C.

► 10.6.2

Solubilidad de dopantes

En la tabla 8.22 se encuentran los valores que especifican la solubilidad máxima para cada dopante, dependiendo de la temperatura. Con esos datos se calcula la concentración máxima del dopante (en su estado eléctricamente activo) como

$$C_{\text{sol}}(B) = C_{0,\text{Sol}}(B) \times \exp\left(\frac{-E_{\text{Sol}}(B)}{k_B T}\right),$$

tal y como se explica en la ecuación 8.19.

Una forma de obtener esta concentración máxima es a través del proceso denominado in-diffusion, que consiste en poner una fuente de dopante en la superficie del silicio, de tal forma que el dopante vaya difundiendo hacia el interior

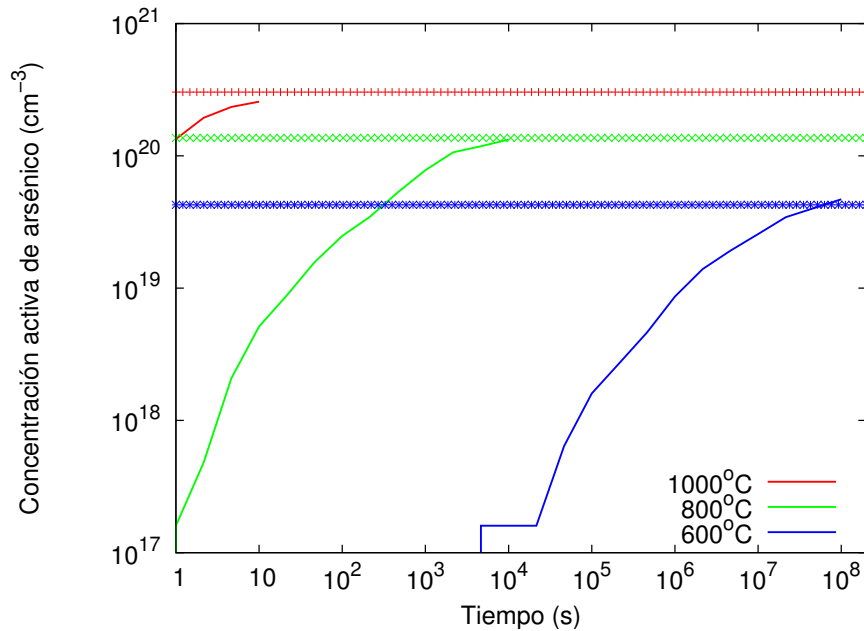


Figura 10.20: Proceso de in-difusión del arsénico a 600, 800 y 1000 °C. Líneas: simulación. Símbolos: solubilidad.

del silicio, hasta lograr saturar éste completamente (concentración homogénea y máxima en todo el silicio).

La figura 10.20 muestra una simulación de este proceso con DADOS, y como la concentración de dopante va subiendo (líneas), hasta que se estabiliza en el valor teórico (símbolos), que es calculado con la ecuación anteriormente citada.

Los comandos utilizados fueron:

- Se crea una celdilla de $15 \times 15 \times 40 \text{ nm}^3$.
- Se habilita la in-difusión para el arsénico.
- Recocido: 1000 s a 1000 °C, 10000 s a 800 °C y 1×10^5 s a 600 °C.

Un proceso similar, pero realizado para el boro en vez del arsénico (figura 10.21), cuyos comandos son:

- Se crea una celdilla de $15 \times 15 \times 40 \text{ nm}^3$.
- Se habilita la in-difusión para el boro.
- Recocido: 1000 s a 1000 °C, 10000 s a 800 °C y 1×10^5 s a 600 °C.

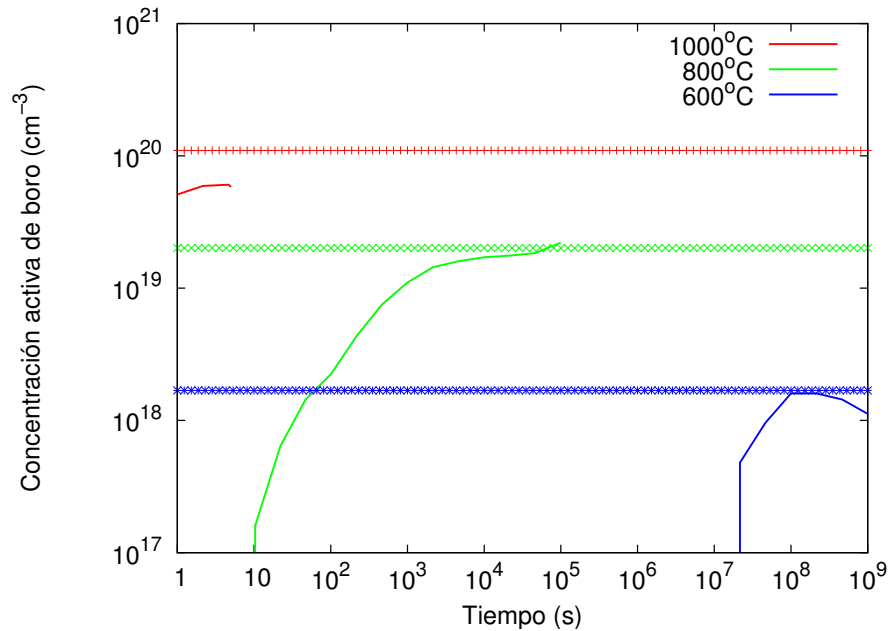


Figura 10.21: In-diffusion del boro a 600, 800 y 1000°C . Líneas: simulación. Símbolos: solubilidad.

▷ 10.7

Modelo de carga

▷ 10.7.1

Difusión extrínseca

La difusividad de una especie depende de la concentración de dicha especie, en principio, a partir de cierto límite en el cual la concentración de dopante eléctricamente activo supera la concentración de electrones en equilibrio para el material, puesto que dicha concentración modifica el nivel de Fermi. En consecuencia, es posible establecer dos límites para la difusividad, uno llamado intrínseco cuando no se supera aún n_i y otro extrínseco para el caso contrario. En el caso extrínseco, la dependencia de la difusividad con el nivel de Fermi modifica significativamente ésta respecto al límite intrínseco.

Un ejemplo de difusividad extrínseca para el arsénico se encuentra en la figura 10.22. Los símbolos están calculados con la ayuda de un sencillo simulador de continuo (que admite condición extrínseca), y las líneas son los resultados de simulación del programa DADOS.

La misma idea se encuentran en la figura 10.23, si bien esta vez para el boro. Nuevamente los símbolos se calcularon con el programa de continuo, frente a las líneas, calculadas usando DADOS.

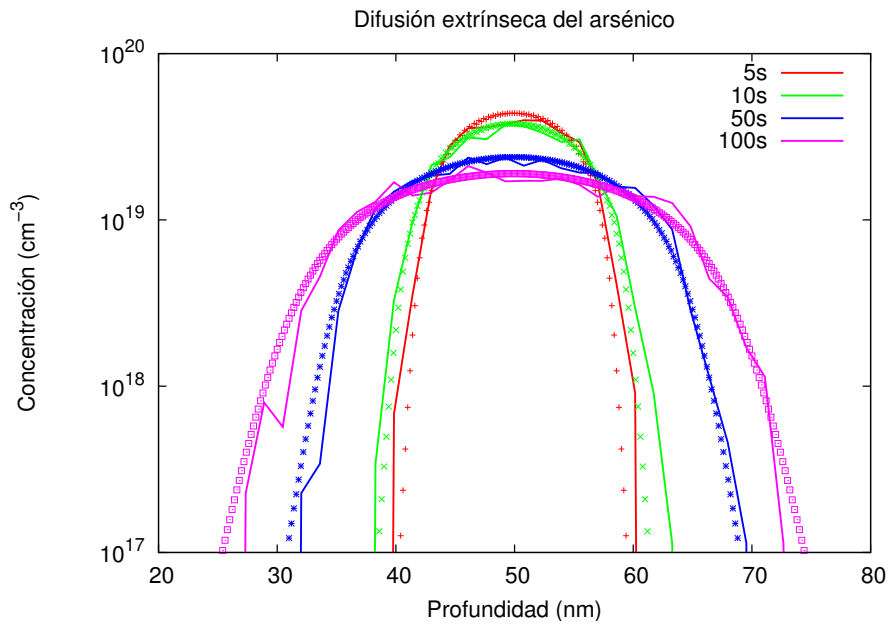


Figura 10.22: Difusión extrínseca del arsénico usando condiciones de Maxwell-Boltzmann a distintos tiempos. Líneas: simulación. Símbolos: valores teóricos.

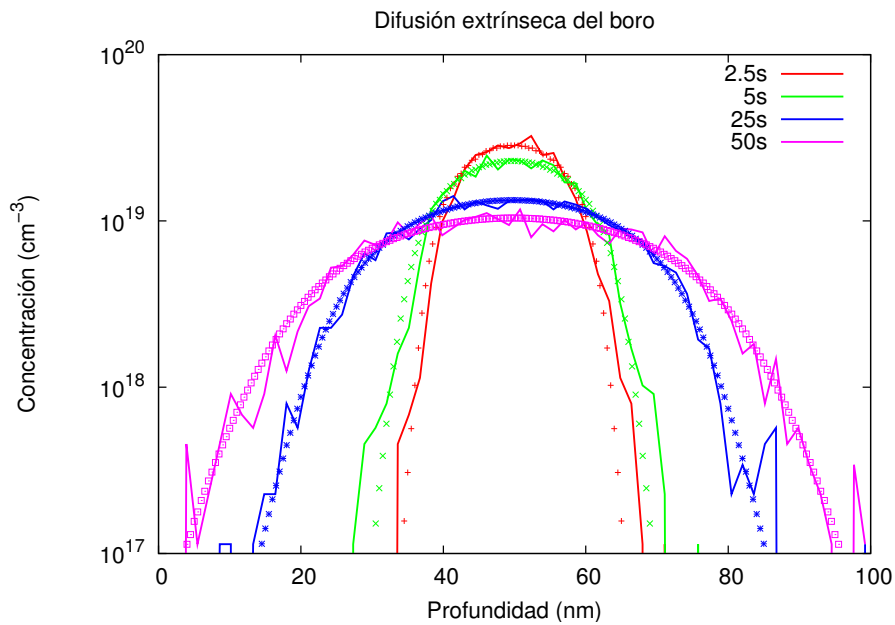


Figura 10.23: Difusión extrínseca del boro para distintos tiempos, usando condiciones de Maxwell-Boltzmann. Líneas: simulación. Símbolos: valores teóricos.

Las simulaciones representadas en las figuras fueron realizadas con las siguientes instrucciones:

1. Arsénico:

- Se crea una celdilla de $100 \times 25 \times 25 \text{ nm}^3$.
- Se habilita la estadística de Maxwell-Boltzmann.
- Se deshabilita la formación de clusters de arsénico.
- Se lee un perfil de As: concentración de $5 \times 10^{19} \text{ cm}^{-3}$ desde 45 hasta 55 nm.
- Recocido: 100 s. a 1000°C .

2. Boro:

- Se crea una celdilla de $100 \times 25 \times 25 \text{ nm}^3$.
- Se habilita la estadística de Maxwell-Boltzmann.
- Se deshabilita la formación de clusters de boro.
- Se lee un perfil de B: concentración de $5 \times 10^{19} \text{ cm}^{-3}$ desde 45 hasta 55 nm.
- Recocido: 50 s a 1000°C .

Ha habido que utilizar estadística de Maxwell-Boltzmann y deshabilitar la formación de clusters de impurezas debido a que el programa de continuo con el que comparamos los resultados es bastante simple, y no admite estadística Fermi-Dirac ni contiene fenómenos de clusterización.

► 10.7.2

Redistribución de dopantes en las uniones *PN*

Al recocer las uniones *PN* se observa una redistribución de dopantes debida a los efectos de carga. Esta redistribución consiste básicamente, en que la concentración de boro disminuye ligeramente fuera de la zona del arsénico para apilarse dentro.

La figura 10.24 muestra un perfil SIMS (en símbolos) tomado de la referencia (Chao et al., 1997) en el que se puede observar esta redistribución. Las líneas son los resultados ofrecidos por DADOS.

Los comandos utilizados son:

- Se crea una celdilla de $300 \times 50 \times 50 \text{ nm}^3$.
- Se lee un perfil de boro: concentración de $1 \times 10^{18} \text{ cm}^{-3}$ desde 0 hasta 300 nm (es decir, toda la muestra).

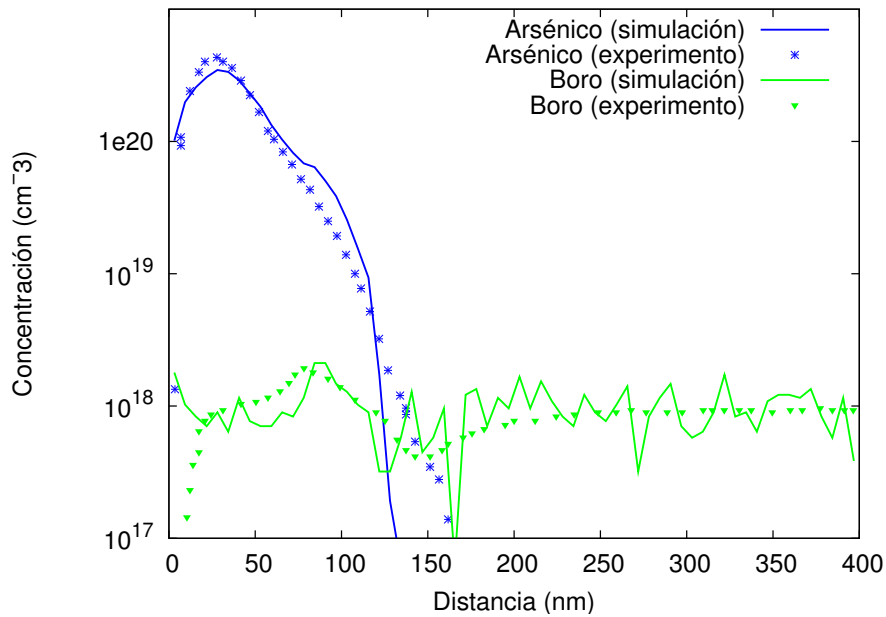


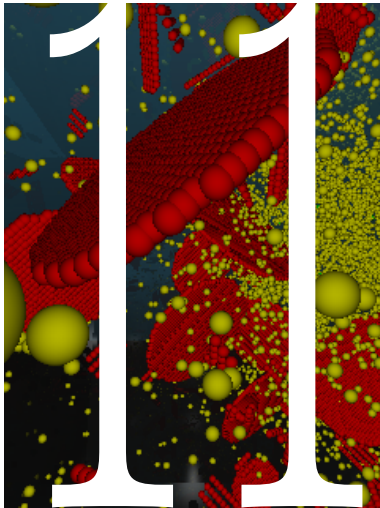
Figura 10.24: Redistribución de dopantes en un unión PN . Se implantó arsénico con una concentración de $2 \times 10^{15} \text{ cm}^{-2}$ con 50 keV de energía, que fue posteriormente recocido a 750°C durante 2 horas. Símbolos: datos experimentales de (Chao et al., 1997). Líneas: simulación.

- Se implanta arsénico. Dosis: $2 \times 10^{15} \text{ cm}^{-2}$. Energía: 50 keV.
- Rampa de recocido: desde 27 hasta 750 °C en 10 s.
- Recocido: 2 h a 750 °C.

▷ 10.8

Conclusiones

Los ejemplos que hemos ido detallando en esta sección se encargan de ilustrar cómo funcionan en la práctica todos los modelos explicados en capítulos precedentes, y cómo éstos son capaces de simular adecuadamente situaciones de procesamiento de silicio bastante complejas. La técnica de Monte Carlo cinética presentada incorpora todos los modelos y mecanismos a la vez, siendo capaz de simular todo el anterior y exhaustivo conjunto de situaciones **simultáneamente y con un único conjunto de parámetros**.



Resultados recientes

| | |
|---|-----|
| 11.1. El di-intersticial móvil del Silicio | 251 |
| 11.2. Cálculos para la movilidad del I_2 | 262 |
| 11.3. Importancia de la correlación de los pares IV | 270 |
| 11.4. Modelizado físico de los efectos del nivel de Fermi | 278 |
| 11.5. Modelizado físico del dañado, amorfización y recristalización | 291 |
| 11.6. Un modelo físico y extenso para el arsénico en Silicio | 301 |

Se exponen en este capítulo los más recientes estudios realizados usando DADOS, que abarcan desde la posibilidad de incluir el cluster de silicio I_2 como partícula móvil, hasta la modelización de la difusión y clustering del arsénico, pasando por un estudio de la importancia de la correlación de los pares de Frenkel en las implantaciones, el modelizado físico del nivel de Fermi y los fenómenos de acumulación de dañado, amorfización y recristalización.

▷ 11.1

El di-intersticial móvil del Silicio

Este estudio está publicado en la referencia (Martin-Bragado et al., 2003).

El di-intersticial móvil del Silicio: Fenómenos de superficie, auto-difusión, formación de clusters y “transient enhanced diffusion” (TED)

Resumen

Bajo la luz de los recientes cálculos teóricos, que predicen la movilidad del di-intersticial del Silicio (I_2), hemos llevado a cabo un análisis de sus implicaciones dentro de un modelo extenso y consistente basado en una técnica atomística de Monte Carlo. En el estudio que se presenta aquí hemos investigado su papel en la recombinación superficial y en la formación de clusters de intersticiales propios, incluyendo también el incremento de difusividad transitoria del boro (TED) en condiciones en las que no existe clustering. Hemos de concluir que, bajo esas condiciones, el I_2 móvil no realiza ningún papel crítico, y que el intersticial propio I puede ser utilizado perfectamente para tener en cuenta todos los efectos del I_2 .

El requerimiento principal a la hora de introducir el I_2 en las simulaciones es la creación de un “cuello de botella”, en dos tamaños de clusters consecutivos, a base de valores de energías relativamente inestables. Además, demostramos que las diferentes longitudes de recombinación medidas bajo condiciones de oxidación e implantación no pueden ser atribuidas a longitudes de recombinación diferentes para el intersticial normal I y el di-intersticial I_2 .

▷ 11.1.1

Introducción

Durante las últimas décadas ha sido adecuadamente probado que la difusión de los dopantes convencionales en silicio está regulada por sus defectos puntuales intrínsecos I y V . La tecnología actual utiliza implantación iónica como proceso principal para introducir dichos dopantes en el silicio. De manera inherente, y como consecuencia de este proceso, se crea una gran cantidad de defectos puntuales que lleva a la formación de diferentes tipos de aglomerados (clusters). Su posterior disolución durante el recocido genera una supersaturación de defectos puntuales que afecta visiblemente a los dopantes implantados, por lo que se necesita un conocimiento profundo de la cinética de disolución de estos defectos para

poder predecir de manera correcta y controlar el perfil final de dopantes en la tecnología nanométrica.

De manera particular, los intersticiales propios (I) extra que se van emitiendo tanto desde los grandes defectos con forma de tiras, llamados $\{311\}$ (Eaglesham et al., 1994), como desde los pequeños clusters (Zhang et al., 1995), se responsabilizan de la *Transient Enhanced Diffusion* (TED) de los dopantes más comúnmente utilizados.

Recientemente, algunos cálculos teóricos han predicho que, además del intersticial propio, el di-intersticial (I_2) también presenta una alta movilidad. Los valores de las energías de migración calculadas dependen de la técnica empleada: dinámica molecular empírica (MD), (Gilmer et al., 1995; Marques et al., 2001) dinámica molecular “tight-binding” (TBMD) (Hane et al., 2000) o aproximación local de la densidad “ab-initio” (LDA), (Estreicher et al., 2001; Chichkine et al., 2002; Chichkine and de Souza, 2002), pero sus rangos están entre los 0.2 y 1 eV. Además, como norma general, la difusividad calculada para el I_2 es más alta que la del intersticial simple.

Por otra parte, hasta donde sabemos, no existen evidencias ni teóricas ni experimentales en contra de una alta movilidad del I_2 pero, a pesar de todo ello, hasta la fecha todos los estudios de simulación de TED y disolución de clusters han supuesto de manera implícita que el I_2 es inmóvil. Una posible razón para ello es que el modelizado del procesamiento de dispositivos avanzados está ya suficientemente saturado de ecuaciones diferenciales que resolver, debido a la complejidad de los mecanismos que necesitan ser tenidos en cuenta. No obstante, la complejidad no resulta un problema excesivo para los simuladores de procesos atomístico de Monte Carlo cinético (kMC) (Jaraiz et al., 2001; Ezaki et al., 2002) cuya principal limitación radica en que únicamente pueden manejar dispositivos con tamaño de algunos cientos de nanómetros. Otra característica atractiva de los simuladores de procesos kMC es que se pueden implementar directamente mecanismos fundamentales, y usar como entrada los parámetros que se derivan de cálculos ab-initio. Por todo ello, estos simuladores sirven como una potente herramienta para probar modelos complejos e investigar posibles hipótesis de simplificación de modelos.

Por las razones que se han mencionado más arriba, trabajos anteriores (Hane et al., 2000; Law et al., 1998) que incluyen la movilidad del I_2 se han centrado principalmente en el papel ejercido por la superficie ante la presencia de I e I_2 , pero no han considerado las posibles interacciones del I_2 con los clusters de I y los dopantes, así como todas las posteriores implicaciones que ello conlleva. En el trabajo que se expone aquí, en cambio, se analizan modelos que, testeados con un simulador de procesos kMC (Jaraiz et al., 1998; Jaraiz et al., 2001), permiten elucidar modelos de interacción que satisfagan el mayor número posible de requerimientos tanto teóricos como experimentales. Entre otras conclusiones, hemos encontrado que las diferentes longitudes de recombinación superficial medidas bajo condiciones de oxidación e implantación no se pueden explicar como

diferentes longitudes de recombinación para el I y el I_2 , pero en cualquier caso encontramos que se puede incluir el I_2 móvil, con los parámetros predichos por los cálculos teóricos, bajo un modelo físico y comprensible en el cual dicho I_2 interactuare con la superficie y con pequeños clusters de intersticiales.

▷ 11.1.2

Comprobando los distintos modelos.

A partir de ahora consideramos que tanto el I como el I_2 son móviles, con unas energías de migración de $E_m(I) = 1$ eV y $E_m(I_2) = 0,5$ eV, respectivamente. Para considerar las interacciones con la superficie, hemos intentado primero comprobar la sugerencia (Hane et al., 2000; Law et al., 1998) según la cual la longitud de recombinación del I_2 es pequeña en comparación con la del intersticial sencillo. Esto quiere decir que el intersticial sencillo se recombinaría en la superficie en muy raras ocasiones, pero en cambio la superficie sería un sumidero perfecto de di-intersticiales. Esta sugerencia es un intento de explicar el hecho de que, en condiciones de oxidación (OED), la longitud de recombinación superficial es grande (cerca de $20\ \mu\text{m}$ en la Ref. (Griffin et al., 1991)), mas para TED esta magnitud es pequeña ($< 10\ \text{nm}$ en las referencias (Lim et al., 1995; Cowern et al., 1999a)). Con dichas longitudes de recombinación, en condiciones de TED, la superficie se comportaría como un sumidero de intersticiales (en la forma de I_2), porque la alta concentración de I s produciría también una alta concentración de I_2 . No obstante, en OED las partículas dominantes serían los I s sencillos, que se recombinarían muy lentamente en la superficie.

Más detalles
en
sec. 11.2.1,
pág. 262.

Ahora bien, las consecuencias, tanto teóricas como experimentales, que conlleva la hipótesis de una longitud de recombinación grande para los I s aún no han sido analizadas en profundidad; esto será lo que haremos a continuación.

Vamos a utilizar el experimento referenciado en (Cowern et al., 1999a) como una prueba para comprobar el modelo de recombinación superficial. En él los autores utilizan una implantación de Ge que amorfiza localmente el silicio y produce una densidad moderada de defectos “end of range” (EOR) (Fig. 11.1). Posteriormente se varía la anchura (W) de la capa amorfizada eliminando el silicio sin afectar a la banda con defectos. Posteriormente se implanta boro, a muy baja energía, en lo que queda de la capa amorfizada, y se pone a $900\ ^\circ\text{C}$ de temperatura. Las muestras son monitorizadas para medir la diferentes concentraciones de intersticiales como una función de la distancia W entre la banda con defectos tipo EOR (que está emitiendo intersticiales) y la superficie. La conclusión es que la concentración de la especie responsable de la difusión del boro disminuye hacia la superficie, pudiendo ser esta especie el I , el I_2 o ambas. Nuestras simulaciones atomísticas, que suponen que los defectos EOR capturan y emiten I e I_2 , muestran que las concentraciones de estas especies no se encuentran en equilibrio local entre sí según se dirigen a la superficie (una suposición que por sencillez adopta (Law et al., 1998)), debido a la muy alta movilidad del I_2 que alcanza la superficie

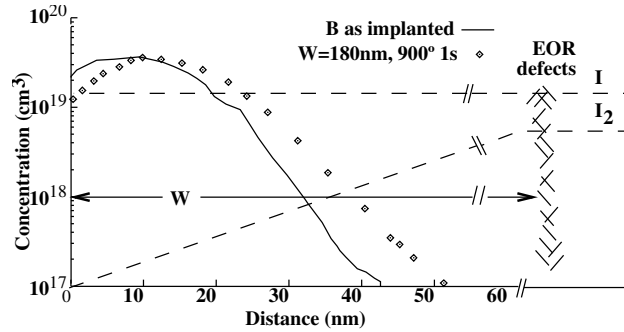


Figura 11.1: Perfiles de profundidad del boro antes y después del recocido de una implantación superficial de boro en silicio preamorfizado con Ge, tomado de la Ref. (Cowern et al., 1999a). Línea continua: perfil de implantación de boro a 3 keV. Símbolos: perfiles SIMS después de 1s de recocido a 900 °C. Líneas punteadas: perfiles simulados de intersticiales (suponiendo que no se recombinan en la superficie) y di-intersticiales según este trabajo, (escala lineal, unidades arbitrarias).

antes de disociarse, y también a la ausencia de clusters u otro tipo de defectos que pudieran actuar como mediadores en la transformación de I_1 a I_2 y viceversa. De esta manera, debido a esta falta de equilibrio local, la concentración de I_1 es uniforme hacia la superficie (Fig. 11.1), y no podría explicar las diferencias en el ensanchamiento del perfil de boro como una función de W . (Cowern et al., 1999a) Muy en cambio, la concentración de I_2 sí que disminuye linealmente a medida que se acerca a la superficie y, por tanto, se podrían intentar explicar los datos experimentales suponiendo que el I_2 interacciona con el boro y controla su difusividad. Habrá que analizar a continuación todas las posibilidades e implicaciones que la anterior hipótesis supone. Para empezar, como primer requerimiento, hay que darse cuenta de que el mecanismo de difusión del boro en equilibrio (esto es, a partir del B_i), tiene que respetarse si se quiere explicar la difusión de este elemento en dichas condiciones (puesto que en equilibrio la concentración de I_2 es despreciable, como se verá más adelante).

Una forma por la cual el I_2 móvil podría contribuir a la difusión del boro sería creando B_i a través de la reacción



(Fig. 11.2). Otra forma es suponer que el BI_2 es también móvil. Si consideramos en primer lugar la primera alternativa, para lograr que el BI_2 se rompa en B_i en vez de volver a dar $B + I_2$, se requiere que (como se puede ver en la Fig. 11.2)

$$E_b(B_i) - E_m(B_i) > E_b(I_2) - E_m(I_2),$$

donde E_b representa la energía de enlace. Valores típicos de los parámetros son $E_b(B_i) = 0,9$ eV, $E_m(B_i) = 0,6$ eV, (Watkins, 2000) $E_b(I_2) = 2$ eV y $E_m(I_2) = 0,5$

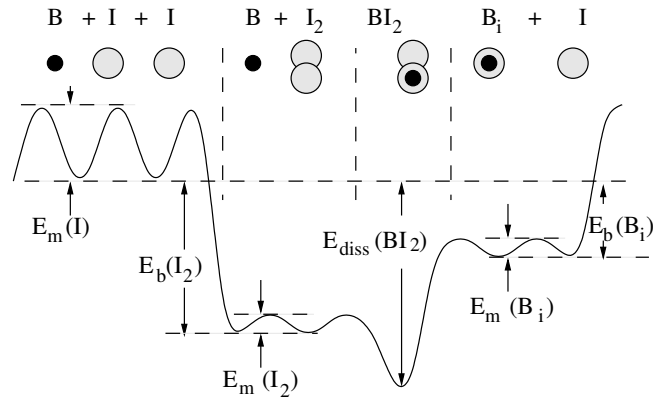


Figura 11.2: Diagrama de energía que muestra los diferentes procesos de emisión y captura que están involucrados en la formación de un BI_2 .

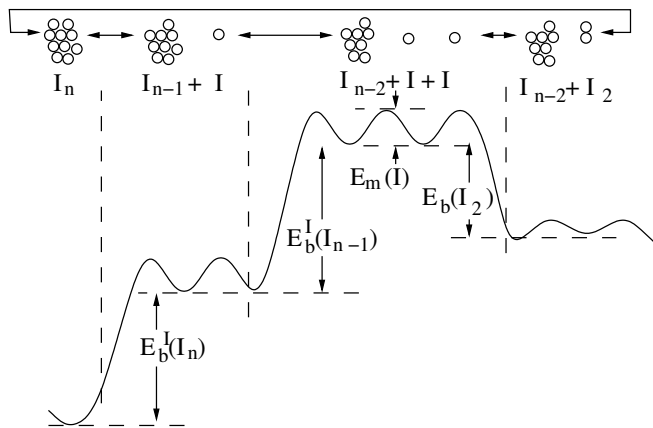


Figura 11.3: Diagrama que muestra las energías involucradas en la emisión y captura de I s e I_2 desde los clusters.

eV. Por tanto es mucho más probable que el BI_2 se rompa dando $I_2 + B$ a que genere $B_i + I$.

A continuación es menester comentar la segunda alternativa, es decir la suposición de que el BI_2 sea móvil por si mismo. Hasta donde sabemos, no se ha publicado ningún estudio sobre la movilidad del BI_2 o del B_2I_2 hasta el momento pero, al menos en principio, uno podría considerarlos como posibles candidatos a ser móviles basándose en el razonamiento de que $I + I$ produce una cluster móvil (el I_2), y el átomo de boro intersticial (B_i) es más móvil que el propio I . Las tres especies consideradas (I_2 , BI_2 y B_2I_2) consisten en dos átomos extra (silicio o boro) dentro de la red de silicio, con lo que si al añadir B al I el conjunto se mueve más que el original ¿por qué no al tener $B + I_2$?. En cualquier caso, regresando a la explicación según la cual el BI_2 móvil podría explicar la dependencia con

la profundidad de una fuente de I_s en los experimentos de TED, el flujo de BI_2 debería ser dominante sobre el de B_i , y esto implicaría que

$$E_f^{TED}(BI_2) + E_m(BI_2) < E_f^{TED}(B_i) + E_m(B_i),$$

donde

$$E_f^{TED}(B_i) = E_b^I(\{311\}) - E_b(B_i)$$

y

$$E_f^{TED}(BI_2) = 2E_b^{311}(I) - E_{diss}(BI_2).$$

E_{diss} representa la energía de disociación y $E_b^I(\{311\})$ es la energía de enlace de un I a un defecto tipo $\{311\}$ (vea la figura 11.3). El valor $E_b^I(\{311\}) = 2,7$ eV es comúnmente aceptado para ello. (Eaglesham et al., 1994) Pero incluso suponiendo que $E_m(BI_2) = 0,2$ eV (que sería el valor mínimo que uno podría esperar) la $E_{diss}(BI_2)$ tendría que ser más grande que 3.1 eV (valor que tendría que ser comparado al valor calculado de 2–2.5 eV, Ref. (Liu et al., 2000)). Además, bajo condicione de equilibrio (donde la presencia de I_2 sea despreciable), la componente fundamental del flujo de boro tiene que deberse al B_i , y esta nueva condición se traduce en que

$$E_{diss}(BI_2) < E_f(I) + E_m(BI_2) + E_b(B_i) - E_m(B_i),$$

donde $E_f(I)$ es la energía de formación de I desde la superficie. Lo anterior implica que $E_{diss}(BI_2) < 4$ eV. Así pues, para satisfacer ambas condiciones a la vez de forma segura, $E_{diss}(BI_2)$ debería encontrarse entre los valores de 3.4 y 3.7 eV, aproximadamente. Una condición extra que se muestra en las simulaciones es la necesidad de prevenir saltos de difusión del B demasiado grandes, que llevarían a perfiles exponenciales, en vez de los típicos perfiles gaussianos (Fig. 11.1) que se observan en los experimentos. (Covern et al., 1999a) Con el BI_2 móvil, estos saltos largos procederían de una ruptura demasiado lenta del $BI_2 \rightarrow B + I_2$. Este nuevo requerimiento, junto con el valor de $E_{diss}(BI_2)$ entre 3.4 y 3.7 eV, definiría una horquilla aceptable para $E_b(I_2)$ de 2.7 a 2.8 eV, que se sitúa muy lejos de los rangos que se calculan para este valor entre 1.8 a 2.2 eV. Además, un valor de $E_b(I_2) = 2,7$ a 2,8 eV correspondería a una energía de activación para los $\{311\}$ s de (vea la Fig. 11.3)

$$E_b(I_{n+2}) + E_b(I_{n+1}) - E_b(I_2) + E_m(I_2) \approx 2,9 \text{ eV}$$

en contradicción con los valores medidos de 3.6 eV. (Eaglesham et al., 1994) Así pues, incluso si se permite un BI_2 móvil, obligar al flujo de BI_2 hacia la superficie a ser mucho mayor que el de B_i , no encajaría con las observaciones de índole experimental. (Covern et al., 1999a)

Resumiendo, debemos concluir que cualquier intento de atribuir las longitudes de recombinación diferentes que se observan en condiciones de OED y TED a

longitudes de recombinación diferentes para el I y el I_2 , es incompatible con otras condiciones teóricas y experimentales. Además, existen indicios de que la longitud de recombinación en OED es diferente dependiendo de la técnica de oxidación utilizada (por ejemplo, los oxinitruros (Rubin et al., 1999; Aoki et al., 2000)), por lo que se puede sospechar que pueden existir efectos químicos en la superficie que afecten a la forma en que se recombinan las partículas en ésta. En consecuencia, en estos momentos no podemos ofrecer un modelo extenso y con bases físicas para las longitudes de recombinación aparentemente diferentes que existen en condiciones de OED y TED.

▷ 11.1.3

Modelo propuesto

Habiendo desechado algunos modelos alternativos, es hora de proponer un modelo que cumpla con todas las condiciones, tanto experimentales como teóricas, que se han expresado en anteriores párrafos.

La superficie, como una conclusión que extraemos del análisis anterior, debe ser un buen sumidero para el I (al menos bajo condiciones TED), debido a que necesitamos esta condición para explicar el ensanchamiento experimental del boro. Respecto a la longitud de recombinación superficial del I_2 , en principio no somos conscientes de ninguna restricción hasta ahora. Por tanto, hemos llevado a cabo algunas simulaciones MD usando potenciales de Tersoff (Marques, 2003) y Stillinger–Weber, y en ellas se encuentra la misma conducta para el I y el I_2 cuando interactúan con la superficie libre del silicio: la superficie se comporta como un sumidero (al contrario que en el sencillo modelo conceptual que sugirió Hu (Hu, 1981)). En nuestro modelo permitimos a la superficie emitir también I_2 , para conseguir reversibilidad microscópica, aunque su energía de activación resulta lo suficientemente alta (como se verá más abajo) para que no se observe dicha emisión desde la superficie. De esta manera el modelo resulta ser bastante sencillo: se incluye un I_2 móvil, con los parámetros predichos por cálculos teóricos, pero bajo condiciones de TED el transporte de boro sigue siendo controlado principalmente por el I normal. Junto con esto, se pone una superficie que capture los I e I_2 . Además suponemos que los I_2 pueden interactuar con clusters de intersticiales pequeños, influenciando su proceso de crecimiento. La energía de activación para emitir un I_2 desde uno de estos clusters, de tamaño n , se puede calcular a partir de la energía de enlace al cluster del I normal, $E_b^I(I_n)$ como se ve en la Figura 11.3:

$$E_b^I(I_n) + E_b^I(I_{n-1}) - E_b(I_2) + E_m(I_2).$$

Utilizando condiciones de reversibilidad microscópica se puede demostrar que el prefactor de emisión de I_2 desde los clusters está dado por

$$\frac{2V_{cap}(I)D_{0,em}^I(I_n)D_{0,em}^I(I_{n-1})D_{0,m}(I_2)}{V_{cap}(I_{n-1})D_{0,m}(I)D_{0,break}(I_2)},$$

Más detalles
en
sec. 11.2.2,
pág. 262.

Más detalles
en
sec. 11.2.4,
pág. 264.

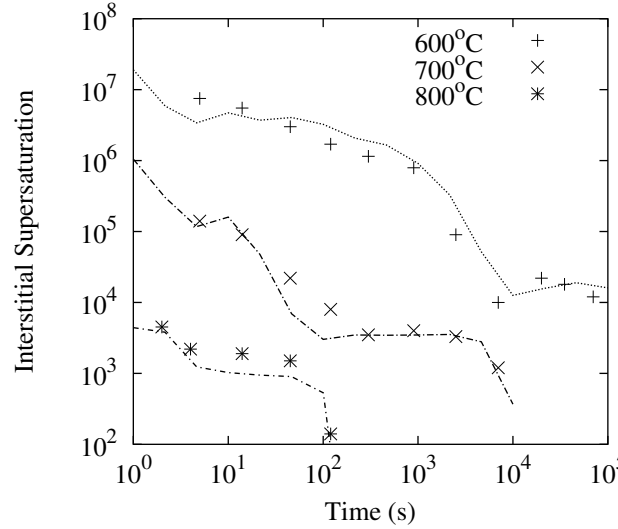


Figura 11.4: Supersaturación de intersticiales en función del tiempo para tres temperaturas de recocido distintas, después de una implantación de Si de 40 keV, $2 \times 10^{13} \text{cm}^{-2}$. Los símbolos representan los valores experimentales, que han sido tomados de la Ref. (Cowern et al., 1999c). Las líneas son resultado de las simulaciones, usando el modelo presentado en esta trabajo, el cual incluye la movilidad del I_2 .

donde $V_{cap}(I_n)$ es el volumen de captura del cluster, $D_{0,em}^I(I_n)$ el prefactor de emisión para tamaño n , $D_{0,m}(I_2)$ el prefactor de migración del di-intersticial, $D_{0,m}(I)$ el prefactor de migración del I y $D_{0,break}(I_2)$ el prefactor de ruptura del di-intersticial.

Suponemos también que no hay interacción del I_2 con los defectos $\{311\}$, porque la interacción sería energéticamente poco favorable, basándose en la reordenación relativa que supondría para ser incluido dentro de la estructura de cadena de los $\{311\}$, y también en que no es necesario para que éstos se disuelvan, ya que lo seguirán haciendo a través de la emisión de I s normales.

Hemos comprobado el modelo resultante, que incluye la movilidad del I_2 , a partir de la simulación del tiempo de evolución medido de la supersaturación de I (medidas TED (Cowern et al., 1999c)), y también a partir de la disolución de los defectos $\{311\}$, que se determina directamente por medidas de microscopía electrónica de transmisión (TEM). (Eaglesham et al., 1994) Para hacer las simulaciones hemos utilizado el programa MARLOWE (Robinson and Torrens, 1974) para generar las cascadas de implantación, y el código atomístico de Monte Carlo cinético (kMC) DADOS (Jaraiz et al., 1998) para simular el ulterior recocido. La figura 11.4 muestra la evolución experimental de la supersaturación de I s (concentración de I s respecto a la que habría en equilibrio, C_I/C_{I^*} , a temperaturas distintas (Cowern et al., 1999c)) que se extrae del incremento de difusión del boro. Los resultados simulados, que también se muestran en la misma gráfica, coinci-

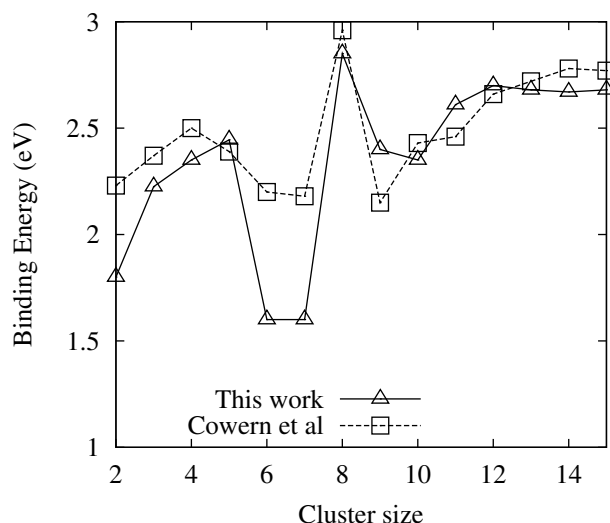


Figura 11.5: Energías de enlace para los clusters de intersticiales, $E_b^I(I_n)$, en función de sus tamaños. Cuadrados: modelo anterior, que sólo utilizaba el I móvil. (Cowern et al., 1999c) Triángulos: modelo actual.

den adecuadamente con los experimentales. Es importante señalar que, cuando comparamos nuestras simulaciones con las medidas TED, (Cowern et al., 1999c) hemos supuesto que el ensanchamiento del boro produce una medida directa de la supersaturación de I_s . Esto es porque, como ya se ha comentado anteriormente, la difusión del boro se produce principalmente a través del B_i , que se forma a partir de $I + B$. De esta forma la presencia de I_2 no incrementa la concentración de B_i , de la que seguimos suponiendo que es la principal responsable de la difusión del boro.

El mismo conjunto de energías de enlace para los clusters I_n ha sido utilizada en todas las simulaciones, y se representa en la figura 11.5, junto con la media de los tres conjuntos (uno por cada temperatura), que se usaron en la referencia (Cowern et al., 1999c). La diferencia más notable consiste en las débiles energías de enlace para los tamaños de clusters 6 y 7, que actúan como un cuello de botella para el crecimiento de los clusters. La presencia de este cuello de botella es necesaria para obtener la forma escalonada que se observa en la evolución temporal de la supersaturación; en caso contrario el I_2 móvil podría puentear el cuello de botella en el tamaño 8, acelerando la formación de $\{311\}$ s. Con nuestras hipótesis de emisión y captura de I_2 desde los clusters pequeños, se requiere un cuello de botella que abarque dos tamaños consecutivos de cluster con una energía de enlace relativamente débil que, intentando seguir la referencia (Cowern et al., 1999c), hemos supuesto que se encontrarían en los tamaños 6 y 7. Como se puede observar en la figura 11.5, se necesitan oscilaciones de las energías bastante grandes, estando así más de acuerdo con las amplitudes de las

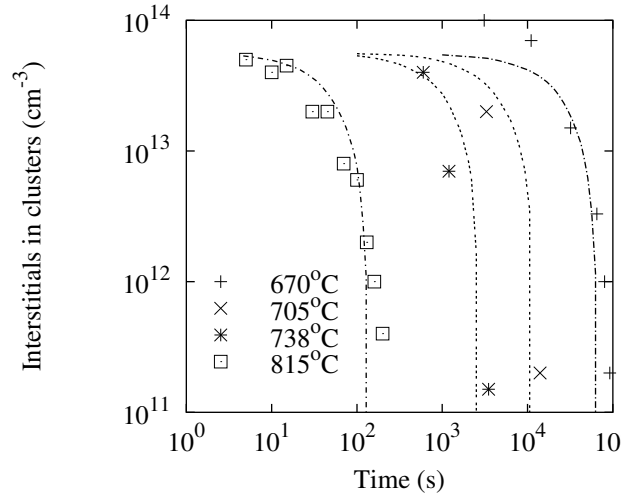


Figura 11.6: Concentración de clusters de intersticiales, después de una implantación de Si de 40 keV, $5 \times 10^{13} \text{ cm}^{-2}$, en función del tiempo y temperatura de recocido. Los símbolos son los datos experimentales extraídos de la referencia (Eaglesham et al., 1994), las líneas representan los valores simulados, utilizado el modelo presentado aquí, que incluye la movilidad del I_2 .

oscilaciones calculadas teóricamente (Chichkine and de Souza, 2002), que también contienen dos clusters poco estables y consecutivos en los tamaños 6 y 7. Así mismo, nuestras simulaciones son bastante sensibles al valor utilizado como energía de enlace del I_2 (el cual fue usado como parámetro de ajuste), y obtuvimos un valor de $E_b(I_2) = 1,8 \text{ eV}$, que está bastante de acuerdo con los cálculos LDA, (Chichkine and de Souza, 2002) pero resulta algo más bajo que los valores obtenidos con técnicas TBMD (2.2 eV). (Hane et al., 2000)

Otra comprobación realizada, figura 11.6, muestra el número total de intersticiales contenidos en clusters tipo $\{311\}$ después de la implantación, en función del tiempo de recocido. Los datos experimentales están cogidos de la referencia (Stolk et al., 1997) y las simulaciones han sido realizadas usando los parámetros representados en la figura 11.4. Hemos supuesto que las energías de enlace de los clusters I_n con $n > 15$ tienden asintóticamente a $E_b^I(\{311\}) = 2,7 \text{ eV}$, que es el valor que mejor ajusta las curvas de disolución. Esta energía de enlace se corresponde con una energía de activación de los clusters $\{311\}$ de 3.7 eV, valor que está de acuerdo con los 3.6 eV de la referencia (Stolk et al., 1997).

Con los valores que este modelo asigna a los parámetros, la contribución que los I_2 aportan a la auto-difusión es despreciable en equilibrio. De hecho, el producto $D_{I_2} C_{I_2}^*$, siendo D_{I_2} la difusividad del I_2 y $C_{I_2}^*$ su concentración en equilibrio, tiene una energía de activación de

$$2E_f(I) + E_m(I_2) - E_b(I_2) = 6,7 \text{ eV}.$$

Esta energía es mucho más grande que los $\sim 4,9$ eV que se miden en los experimentos de difusión de metales (Bracht et al., 1995). Así pues $D_{I_2}C_{I_2}^*$ es mucho más baja que la contribución dominante del intersticial simple $D_I C_I^*$. Muy al contrario, en condiciones de casi-equilibrio, los efectos del I_2 para concentraciones altas de I serán más importantes que los del intersticial sencillo, porque la supersaturación de I_2 ($C_{I_2}/C_{I_2}^*$) y la de Is están relacionadas por la ecuación $C_{I_2}/C_{I_2}^* = (C_I/C_I^*)^2$. En consecuencia, el papel del I_2 (por ejemplo en las posibles reacciones que mantenga con clusters $B_m I_n$) será mucho más acuciado en condiciones TED que en OED.

Merece la pena señalar también que la inclusión de la movilidad del I_2 requiere una revisión no sólo de los parámetros de energía para los clusters de I I_n , como hemos hecho, sino también de los clusters que se formarían entre Is y dopantes. En concreto, los clusters $B_m I_n$, que se piensa que juegan un papel primordial en la difusión del boro y en su desactivación eléctrica, debido a que son capaces de inmovilizar boro. Estos clusters se forman sólo en regiones con muy alta concentración de intersticiales. En el modelo presentado en la referencia (Pelaz et al., 1997), se supone que el origen de la nucleación de los clusters $B_m I_n$ está en el BI_2 . En nuestra visión, la formación de BI_2 a través de $I_2 + B$ podría llevar a una nucleación en regiones con baja concentración de intersticiales por culpa de la migración los I_2 móviles y estables. Así pues, se necesitaría una revisión del proceso de nucleación de los clusters de boro y, en general, de los parámetros energéticos de los $B_n I_m$, que queda fuera del alcance e intención de este estudio. El I_2 móvil, con su alta difusividad a temperatura ambiente, podría ser también un candidato para explicar la desactivación de dopantes a temperatura ambiente que se observa después de una implantación superficial de dañado (Larsen et al., 1996).

Más detalles
en
sec. 11.2.5,
pág. 266.

Más detalles
en
sec. 11.2.7,
pág. 267.

▷ 11.1.4

Conclusiones

Concluyendo, hemos desarrollado un modelo amplio y con base física, que es capaz de reconciliar la alta movilidad del I_2 notificada en cálculos teóricos, junto con otras condiciones teóricas y experimentales. Hemos analizado también algunas de las implicaciones que supondría la inclusión de esta nueva especie móvil. Las energías de ligadura de los clusters, en particular, tienen que reajustarse, al menos en los tamaños pequeños.

No obstante, los intentos de considerar las diferentes longitudes de recombinación que se observan en experimentos OED y TED asignando una eficiencia de captura superficial distinta para los Is y los I_2 , fallan gravemente cuando intentan cumplir con otras restricciones que otros experimentos y cálculos teóricos imponen.

Así pues, la visión global que surge de este estudio es que la movilidad del I_2 , que está teóricamente predicha, se puede incluir, pero no parece jugar un papel

destacado en las simulaciones de difusión del boro (TED), al menos cuando no hay formación de clusters de boro. Esto se debe a que el flujo dominante para el boro sigue siendo controlado por el intersticial sencillo (I), para el cual la superficie es un sumidero perfecto. Desde un punto de vista meramente práctico no sería necesario incluir esta nueva especie (el I_2) en los simuladores de procesos convencionales: el I normal es capaz de asumir las aportaciones del I_2 . En cualquier caso debemos volver a llamar la atención sobre que las conclusiones aquí expuestas únicamente son aplicables bajo las condiciones investigadas en este estudio, es decir, cuando no hay formación de clusters de boro. Quedaría por ver que sucede con la presencia del I_2 móvil cuando hay formación de clusters de boro o desactivación del mismo. Futuros cálculos de las energías de los clusters $B_m I_n$, así como de la posible movilidad del BI_2 y B_2I_2 (de Souza, 2002) (con dos átomos extra, como el I_2), ayudarán sin duda a descubrir los mecanismos involucrados en este tipo de condiciones.

▷ 11.2

Cálculos para la movilidad del I_2

▷ 11.2.1

Difusividad del I_2

La difusividad del I_2 se define como:

$$D(I_2) = D_{0,m}(I_2) \times \exp(-E_m(I_2)/k_B T), \quad (11.1)$$

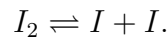
donde la energía de migración y el prefactor son parámetros externos.

▷ 11.2.2

Equilibrio y emisión desde la superficie

Concentración de equilibrio del I_2

La concentración en equilibrio del I_2 ($[I_2]^*$) se extrae utilizando la reacción entre el I y el I_2



El número de clusters de I_2 que se rompen deberá ser el mismo que el número que se forma a partir del choque de dos I s (para mantener el equilibrio), es decir

$$[I_2]^* R_{break}(I_2) = 2[I]^*{}^2 V_{cap}(1) R_m(I),$$

y por tanto

$$[I_2]^* = \frac{2[I]^*{}^2 V_{cap}(1) R_m(I)}{R_{break}(I_2)}. \quad (11.2)$$

Energía de activación y prefactor para la emisión desde la superficie

El número de clusters móviles de I_2 que se capturan en la superficie es:

$$[I_2]^* Y Z \lambda \frac{1}{6} \frac{6}{\lambda^2} D_{0,m}(I_2) \times \exp(-E_m(I_2)/k_B T) P_{rec},$$

y el número de I_2 emitidos es:

$$\frac{2}{a^2} Y Z P_{rec} \frac{6}{\lambda^2} D_{0,surface}(I_2) \times \exp(-E_{act}(I_2)/k_B T).$$

Como ambas expresiones deben tener el mismo valor, es sencillo obtener que

$$[I_2]^* = \frac{6}{\lambda^3} D_{0,surface}(I_2) \exp(-E_{act}(I_2)/k_B T) \frac{1}{D_{0,m}(I_2)}. \quad (11.3)$$

Conocemos $[I_2]^*$ de la ecuación 11.2, y por tanto podemos obtener $D_{0,surface}(I_2)$ y $E_{act}(I_2)$:

$$E_{act}(I_2) = 2E_f(I) + E_m(I_2) - E_b(I_2) \quad (11.4)$$

$$D_{0,surface}(I_2) = \frac{12}{\lambda^3} V_{cap}(1) \frac{D_{0,m}(I)}{D_{0,break}(I_2)} (D_{0,surface}(I))^2 D_{0,m}(I_2). \quad (11.5)$$

▷ La ley de acción de masas nos dice que

$$[I_2] \exp(-E_b(I_2)/k_B T) = [I]^2$$

donde no tenemos en cuenta la diferencia en entropía vibracional entre los dos estados. Esta ecuación implica que habrá una gran concentración de I_2 en equilibrio siempre que la concentración de I s sea lo suficientemente grande (referencia (Gilmer et al., 1995)).

Producto $D(I_2)[I_2]$

Multiplicando $D(I_2)$ por $[I_2]$ a partir de las ecuaciones 11.1 y 11.3 tenemos que:

$$D(I_2)C[I_2] = \frac{6}{\lambda^3} D_{0,surface}(I_2) \times \exp[-(E_f(I_2) + E_m(I_2))].$$

▷ 11.2.3

Comparación entre $D(I_2)[I_2]^*$ y $D(I)[I]^*$

Para obtener cómo de importante es el producto $D[I_2][I_2]^*$ le dividimos por $D(I)[I]^*$.

$$\frac{D(I_2)[I_2]^*}{D(I)[I]^*} = \frac{12}{\lambda^3} \overbrace{V_{cap}(1)}^{3\lambda^3} \times D_{0,surface}(I) / D_{0,break}(I_2) \times \times D_{0,m}(I_2) \times \exp[-(E_f(I_2) + E_m(I_2) - E_f(I) - E_m(I))/k_B T]$$

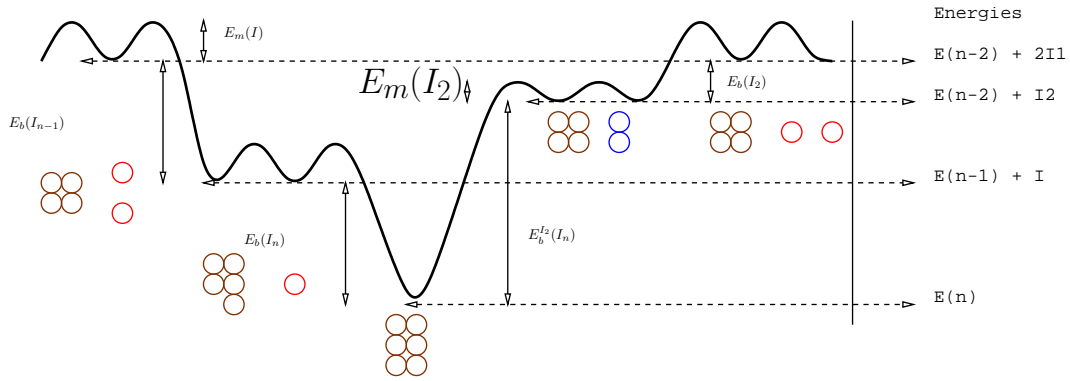


Figura 11.7: Emisión y captura de un I_2 .

es decir

$$\frac{D(I_2)[I_2]^*}{D(I)[I]^*} = 36D_{0,surface}(I)/D_{0,break}(I_2)D_{0,m}(I_2) \times \exp\left[-(E_f(I) + E_m(I_2) - E_m(I) - E_b(I_2))/k_B T\right] \quad (11.6)$$

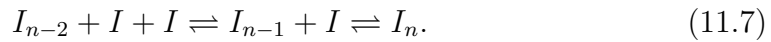
► 11.2.4

Relación entre el I_2 y los clusters de intersticiales

En los párrafos siguientes se va a mostrar como obtener $E_{act,em}^{I_2}(I_n)$ y su correspondiente prefactor, que los clusters de I necesitan para emitir I_2 .

Energía de activación del I_2

La figura 11.7 muestra dos formas de obtener un clusters de intersticiales I_n de un sistema que contenga dos I s y un cluster I_{n-2} .



Por tanto

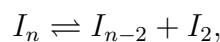
$$E_b^I(I_n) + E_b^I(I_{n-1}) = E_b^{I_2}(I_n) + E_b^I(I_2)$$

y la energía de activación del I_2 será

$$E_{act,em}^{I_2}(I_n) = E_b^{I_2}(I_n) + E_m(I_2) = E_b^I(I_n) + E_b^I(I_{n-1}) - E_b(I_2) + E_m(I_2). \quad (11.8)$$

Prefactores

Para calcular los prefactores de emisión del I_2 hay que investigar las reacciones microscópicas que suceden y asegurarse que éstas preservan la adecuada reversibilidad:



necesitamos tener el mismo número de clusters capturando I_2 que emitiéndole, para así conservar el equilibrio. Por tanto, el número de I_n s que emitirán un I_2 será:

$$\nu_{em}^{I_2}(I_n)[I_n]^*,$$

siendo $\nu_{em}^{I_2}(I_n)$ la frecuencia de emisión de I_2 desde un cluster de tamaño n , es decir

$$\nu_{em}^{I_2}(I_n) = D_{0,em}^{I_2}(n) \times \exp(-E_{act,em}^{I_2}(I_n)/k_B T).$$

El número de clusters que capturan un I_2 y se transforman en un I_n será

$$[I_{n-2}]^* V_{cap}(n-2) [I_2]^* \nu_m(I_2),$$

que resulta ser la concentración de dicha especie, la frecuencia de migración del I_2 y el volumen de captura de un cluster I_{n-2} .

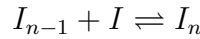
En equilibrio, estas dos contribuciones serán iguales, y podremos obtener la relación:

$$\nu_{em}^{I_2}(I_n) = \frac{[I_{n-2}]^* V_{cap}(n-2) [I_2]^* \nu_m(I_2)}{[I_n]^*} \quad (11.9)$$

A continuación debemos saber el valor de

$$\frac{[I_{n-2}]^*}{[I_n]^*} = \frac{[I_{n-2}]^* [I_{n-1}]^*}{[I_{n-1}]^* [I_n]^*},$$

que se puede extraer utilizando la reacción de emisión de intersticiales



y, por tanto

$$\frac{[I_{n-1}]^*}{[I_n]^*} = \frac{\nu_{em}^I(I_n)}{[I]^* \nu_m(I) V_{cap}(n-1)}. \quad (11.10)$$

En este punto hay que recordar que

$$\begin{aligned} \nu_{em}^I(I_n) &= n D_{0,em}^I(I_n) \times \exp[-(E_b(I_n) + E_m(I))/k_B T] \\ \nu_m(I_2) &= D_{0,m}(I_2) \times \exp(-E_m(I_2)/k_B T) \\ \nu_{break}(I_2) &= D_{0,break} I_2 \times \exp[-(E_b(I_2) + E_m(I))/k_B T], \end{aligned}$$

y por fin, utilizando las relaciones anteriores junto con las ecuaciones 11.9, 11.10 y 11.2 extraemos el valor del prefactor:

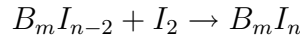
$$D_{0,em}^{I_2}(I_n) = \frac{2V_{cap}(1)(n-1)n(D_{0,em}^I)^2 D_{0,m}(I_2)}{V_{cap}(n-1)D_{0,m}(I)D_{0,break}(I_2)} \quad (11.11)$$

▷ 11.2.5

Emisión de I_2 por parte de los clusters de impurezas

El objetivo de este apartado es calcular las frecuencias de emisión que un cluster de boro e intersticiales tiene a la hora de emitir un I_2 . En cualquier caso, estos cálculos pueden extenderse fácilmente al caso de otras especies, como C o P.

Comenzando con la reacción



el número de clusters de impurezas formados será

$$[I_2][B_m I_{n-2}]v_{cap}(m, n-2)\nu_m(I_2),$$

y el número de rupturas

$$[B_m I_n]\nu_{em}^{I_2}(m, n).$$

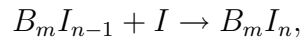
En condiciones de casi-equilibrio se acepta que el número de rupturas sea igual que el de formaciones, y por tanto podemos poner que

$$\nu_{em}^{I_2}(m, n) = [I_2]v_{cap}(m, n-2)\nu_m(I_2)\frac{[B_m I_{n-2}]}{[B_m I_n]}. \quad (11.12)$$

Descomponiendo el último término como

$$\frac{[B_m I_{n-2}]}{[B_m I_n]} = \frac{[B_m I_{n-2}]}{[B_m I_{n-1}]} \frac{[B_m I_{n-1}]}{[B_m I_n]}.$$

La tarea siguiente será calcular estas fracciones. Comenzamos con la reacción



y suponiendo nuevamente casi-equilibrio,

$$[I][B_m I_{n-1}]v_{cap}(m, n-1)\nu_m(I) = [B_m I_n]\nu_{em}^I(B_m I_n),$$

por lo que

$$\frac{[B_m I_{n-1}]}{[B_m I_n]} = \frac{\nu_{em}^I(B_m I_n)}{v_{cap}(m, n-1)\nu_m(I)[I]}.$$

Utilizando las ecuaciones (11.12),(11.2) junto con estas últimas podemos escribir que:

$$\begin{aligned} \nu_{em}^{I_2}(m, n) &= v_{cap}^{I_2}(m, n-2)\nu_m(I_2) \times \\ &\frac{2[I]^* v_{cap}(1)\nu_m(I)}{\nu_{break}(I_2)} \times \frac{\nu_{em}^I(B_m I_{n-1})}{v_{cap}(m, n-2)\nu_m(I)[I]} \times \\ &\frac{\nu_{em}^I(B_m I_n)}{v_{cap}(m, n-1)\nu_m(I)[I]} \end{aligned}$$

Suponemos que

$$v_{cap}^{I_2}(m, n-2) \simeq v_{cap}^I(m, n-1)$$

para simplificar las anteriores ecuaciones y tendremos

$$\nu_{em}^{I_2}(m, n) = \frac{v_{cap}(1)}{v_{cap}(m, n-2)} \times \frac{\nu_m(I_2)}{\nu_{break}(I_2)} \times \frac{\nu_{em}^I(B_m I_{n-1})}{\nu_m(I)} \times \nu_{em}^I(B_m I_n). \quad (11.13)$$

Lo cual lleva al siguiente prefactor para la emisión desde un clusters de impurezas:

$$\text{Prefactor} = \frac{v_{cap}(1)}{v_{cap}(m, n-2)} \times \frac{D_{0,m}(I_2)}{D_{0,break}(I_2)} \times \frac{n(n-1) (D_{0,em}^I(B_m I_n))^2}{D_{0,m}(I)} \quad (11.14)$$

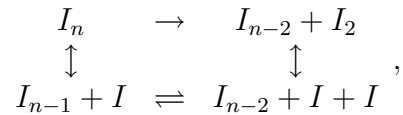
y, teniendo también en cuenta posibles barreras de energía, lleva a la siguiente energía de activación:

$$\begin{aligned} \text{Energía de activación} &= E_m(I_2) + E_{barrier}(m, n-1) + E_{barrier}(m, n) \\ &+ E_b(m, n-1) + E_b(m, n) - E_b(I_2) \end{aligned} \quad (11.15)$$

▷ 11.2.6

El I_2 y su supersaturación

Si tenemos en cuenta las reacciones



podemos mostrar que

$$\frac{[I_2]}{[I_2]^*} = \frac{[I]^2}{[I]^*2} \Rightarrow S_{I_2} = S_I^2. \quad (11.16)$$

Así pues, no se pueden modificar las supersaturaciones de los I_s y los I_2 de forma independiente.¹

▷ 11.2.7

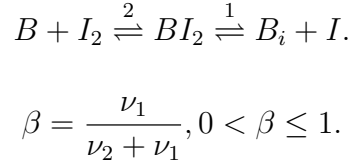
Difusión del boro y movilidad del I_2

Si suponemos que el boro difunde a través del I y el I_2

$$D(B) \propto (D(I)[I] + \beta D(I_2)[I_2]), \quad (11.17)$$

¹Como conclusión, una forma oscilatoria en $E_b(n)$ no ayudaría a incrementar la supersaturación de I_2 , únicamente modificaría la evolución de la distribución de tamaños.

donde β es la fracción de BI_2 s que se rompen como $B_i + I$:



Usando las ecuaciones 11.16 y 11.17 tendremos que:

$$D(B) \propto D(I)[I]^* S_I + \beta D(I_2)[I_2]^* S_I^2 = D(I)[I]^* \left(S_I + \beta \frac{D(I_2)[I_2]^*}{D(I)[I]^*} S_I^2 \right) \quad (11.18)$$

A continuación suponemos que:

- La capacidad de transporte del I cerca del equilibrio está dada principalmente dada por $D_I[I]^*$, y no por $D(I_2)[I_2]^*$.
- El valor de D_B durante la TED está principalmente relacionada con la $D(I_2)$.
- $L_{\text{rec}}(I_2) \ll L_I, r_p$

y podremos escribir:

$$\frac{D_B}{D_B^*} = \frac{D_I[I]^* \left(S_I + \beta \frac{D_{I_2}[I_2]^*}{D_I[I]^*} S_I^2 \right)}{D_I[I]^* + \underbrace{\beta D_{I_2}[I_2]^*}_{\approx 0}} \quad (11.19)$$

La ecuación anterior se transforma en:

$$\frac{D(B)}{D(B)^*} \simeq S_I + \beta \frac{D(I_2)[I_2]^*}{D(I)[I]^*} S_I^2. \quad (11.20)$$

Durante la TED estamos suponiendo que

$$S_I \gg \frac{D(I)[I]^*}{\beta D(I_2)[I_2]^*}$$

para que sean ciertas las hipótesis anteriormente expuestas, por lo que en la TED:

$$\frac{D(B)}{D(B)^*} \simeq \beta \frac{D(I_2)[I_2]^*}{D(I)[I]^*} S_I^2. \quad (11.21)$$

▷ 11.2.8

Disolución los $\{311\}$ e I_2

Para tratar la disolución de los $\{311\}$ s, definimos ϕ como el número de intersticiales que se recombinan en la superficie por unidad de tiempo y área, y así:

$$\begin{aligned}\phi &= D(I)\frac{d[I]}{dx} + 2D(I_2)\frac{d[I_2]}{dx} \\ &= D(I)\frac{[I]}{r_p+L_I} + 2D(I_2)\frac{[I_2]}{r_p+L_{rec}(I_2)} \\ &= D(I)[I^*]S_I \left(\frac{1}{r_p+L_I} + 2\frac{D(I_2)[I_2]^*}{D(I)[I]^*}S_I\frac{1}{r_p+L_{rec}(I_2)} \right)\end{aligned}\quad (11.22)$$

A continuación, utilizando estas hipótesis, la ecuación 11.22 se transforma en:

$$\phi \simeq 2D(I_2)[I_2]^*\frac{S_I^2}{r_p}\quad (11.23)$$

que, con las ecuaciones 11.21 y 11.22 pasa a ser

$$\phi \simeq \frac{2D(I)[I]^*}{\beta r_p} \frac{D(B)}{D(B)^*}.\quad (11.24)$$

Hay que tener en cuenta que esta relación se parece bastante a la expresión

$$\phi \simeq \frac{D(I)[I]^*}{r_p}S_I$$

de (Rafferty et al., 1996), en la cual se identifican S_I y $D(B)/D(B)^*$.

A continuación traducimos β a relaciones de parámetros conocidos:

$$\beta = \frac{\overbrace{\nu_{BI_2 \leftrightarrow B_i+I}}^{\nu_1}}{\underbrace{\nu_{BI_2 \leftrightarrow B_i+I}}_{\nu_1} \underbrace{\nu_{BI_2 \leftrightarrow B+I_2}}_{\nu_2}} = \frac{1}{1 + \frac{\nu_2}{\nu_1}}$$

donde $\beta \simeq 1$ para $\nu_1 \ll \nu_2$.

Puesto que podemos poner $\nu_1 \propto [B][I_2]$ y $\nu_1 \propto [B_i][I]$ tendremos que

$$\frac{\nu_2}{\nu_1} = \frac{[B][I_2]}{[B_i][I]} = S_I \frac{[I_2]^* [B]}{[I]^* [B_i]}.$$

Finalmente β será:

$$\beta = \frac{1}{1 + S_I \frac{[I_2]^* [B]}{[I]^* [B_i]}},\quad (11.25)$$

por lo que podemos suponer que $\beta \simeq 1$ para $S_I[I_2]^*/[I]^* \ll [B_i]/[B]$.

▷ 11.3

Importancia de la correlación de los pares IV

Este artículo ha sido publicado en (Martin-Bragado et al., 2004a) y en (Martin-Bragado et al., 2004b).

Simulaciones de implantación iónica: Un valoración, hecha con Monte Carlo cinético, de las propiedades dominantes en el recocido.

Resumen

Hemos utilizado el método, con bases físicas, de Monte Carlo cinético, junto con la aproximación de colisiones binarias (BCA), para descubrir qué mecanismos de la implantación son relevantes para el modelado de la difusión extra transitoria (TED). Para los casos que hemos estudiado, encontramos que:

1. La correlación espacial de los intersticiales y vacantes en los pares de Frenkel (I,V) no es crítica.
2. La supersaturación de intersticiales en simulaciones donde se incluya el perfil total de I s y V s, o sólo el neto de $I - V$, es la misma.
3. Se pueden usar implantaciones ruidosas (pero rápidas) de BCA directamente, puesto que pese a ello producen simulaciones de recocido lo suficientemente precisas.
4. Cuando haya una concentración de impurezas comparable al exceso neto de $I - V$, hay que usar los perfiles completos de I s y V s para poder reproducir de manera correcta la desactivación y clustering de las impurezas.

Por último, se destacarán algunas implicaciones prácticas de este estudio.

▷ 11.3.1

Introducción

La implantación iónica es actualmente el proceso estándar para establecer los perfiles de los dopantes, los cuales a su vez determinan las características eléctricas de los dispositivos de los circuitos integrados. No obstante, el proceso de implantación crea intersticiales (I) y vacantes (V) extra en el interior del cristal, algunos de ellos a través del reemplazo que el ion genera sobre los átomos de red, y otros a partir de los pares de Frenkel (FP). El recocido posterior a la implantación es capaz de eliminar este dañado, pero la presencia de estos I s no deseados genera una difusión temporal extra, que puede convertirse en un

problema a la hora de procesar especies que difundan por medio del I . (Stolk et al., 1997).

Para tener en cuenta este proceso de TED, los simuladores de procesos suelen generar la distribución de dañado, bien a través de estimaciones analíticas, bien con la aproximación de colisiones binarias (BCA). Este dañado tiene dos componentes, por una parte se encuentra la distribución de I s y V s debida a los pares de Frenkel (FP), y por otra una distribución de I s generada en correspondencia con los iones implantados (para impurezas sustitucionales). El modelo “+1” (Giles, 1991), que se propuso a principios de los 90, supone que todos los I s y V s de los pares de Frenkel se van a recombinar localmente en el volumen, y que por tanto, sólo los I s extra (resultantes de la componente del ion) van a inducir el proceso de TED hasta que sean finalmente capturados en la superficie del silicio. Pese a su aparente simplicidad, este modelo ha sido muy satisfactorio para el boro, por ejemplo, bajo ciertas condiciones, pero en general, se necesita modificar este modelo en un modelo “+n” si se quiere ampliar a otros iones o condiciones. (Pelaz et al., 1998; Hobler and Moroz, 2001).

▷ 11.3.2

Modelo empleado

En este artículo vamos a usar simulaciones realizadas con el método atomístico de Monte Carlo cinético sin red (non lattice kMC), para llevar a cabo un análisis del tema anteriormente expuesto. El método kMC es capaz de realizar simulaciones excepcionalmente detalladas y realistas que ayuden a entender los mecanismos subyacentes y que, de esta forma, permitan descubrir modelos simples y precisos. En nuestro simulador los I s y V s dan saltos aleatorios con una frecuencia derivada de su difusividad, y además están incluidas las interacciones entre partículas, que pueden dar lugar a fenómenos de clustering y re-emisión desde los clusters atrapando y liberando partículas. Las frecuencias de estos procesos se determinan a partir de las energías de enlace específicas que se derivan tanto de cálculos de dinámica molecular o ab-initio, como de experimentos. (Jaraiz et al., 2001) Al contrario que en el método de kMC con red (lattice kMC, ver por ejemplo la Ref. (Takahashi et al., 2003)), nuestro método de kMC tiene en cuenta todas las posibles configuraciones de un cluster de cierto tamaño a través de una única energía efectiva, cosa que es suficiente para los tamaños de simulación y escalas de tiempo que las condiciones de procesado actuales tienen con y sin impurezas (Pelaz et al., 1997; Jaraiz et al., 1996). Concretamente, la energía de activación para la emisión de I s desde clusters de tamaño n se calcula como la suma de la energía de enlace más la de migración de I , y las energías de migración y enlace de este I usadas son las mismas que las indicadas en la Ref. (Martin-Bragado et al., 2003), que consigue un resultado de disolución de los {311}s correcto. Las energías de enlace para las V s se toman de la Ref. (Bongiorno and Colombo, 1998), y los prefactores de emisión se consideran proporcionales

Modelo en la
sec. 8.3.1,
pág. 114.

a la superficie del cluster, en correspondencia con el proceso de captura. En cualquier caso, los valores de E_b afectarán a la escala de tiempos, pero no al número de saltos que las partículas tienen que dar hasta que sean capturadas por la superficie o se recombinen con otra. De hecho, como comentaremos también más adelante, incluso si no se forman clusters de V en absoluto ($E_b = 0$), la evolución de los I s con el tiempo es la misma.

Las coordenadas iniciales de las partículas las calcula un simulador BCA, y alternativamente pueden calcularse de manera aleatoria siguiendo la concentración que le aporte un perfil externo de entrada.

▷ 11.3.3

Temas a tratar

El primer tema que vamos a considerar es la relevancia de la correlación espacial de los FPs: ¿será importante la distancia inicial entre cada I y su correspondiente V o es posible conseguir la misma TED utilizando I s y V s con coordenadas aleatorias que sigan la concentración?. Una vez aclarado esto, habrá que evaluar el peso relativo que tienen cada una de las componentes del dañado: FP e ion. También vamos a mostrar que, para simulaciones de TED, incluso unos perfiles de dañado con bastante ruido, generados con simulaciones de implantación BCA bastante pequeñas y rápidas, pueden ser usados para obtener resultados precisos de la TED. Por último habremos de explorar cómo la presencia de una cantidad sustancial de impurezas puede afectar a los resultados anteriores.

▷ 11.3.4

Simulación empleada

Hemos simulado una implantación de Si con 40 keV, $5 \times 10^{13} \text{cm}^{-2}$ y un recocido posterior a 738 °C durante 6000 segundos como ejemplo de simulación con energía media, dosis media e implantación a temperatura ambiente. La caja de simulación fue de $50 \times 50 \text{nm}^2$, que supuso la implantación de 1250 iones, aproximadamente. Las cascadas se simulan con el programa de BCA MARLOWE (Robinson, 1989), y las coordenadas de los I s y V s son transferidas al simulador atómico de Monte Carlo cinético DADOS (Jaraiz et al., 2001), que se encarga de simular el recocido.

▷ 11.3.5

Correlación espacial de los pares IV

Para comprobar cómo de relevante es la correlación espacial de los FPs, hemos comparado los resultados de la simulación de recocido hechos generando las coordenadas de las partículas mediante BCA, con simulaciones en las cuales los I s y V s de entrada han sido leídos de un perfil. Como el código que traduce los perfiles de concentración en átomos discretos genera las coordenadas de las partículas a

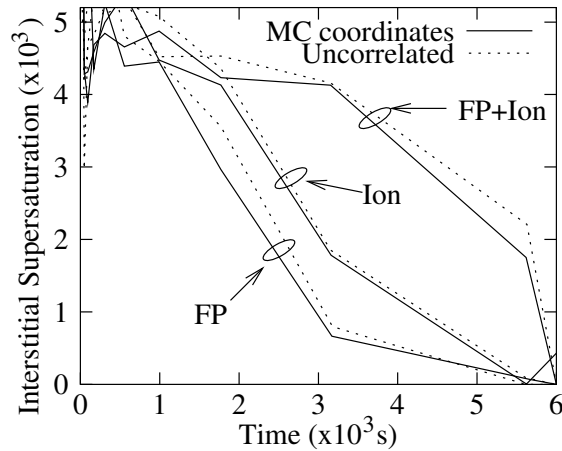


Figura 11.8: Evolución temporal de la supersaturación durante el recocido a 738°C de una implantación de silicio en silicio de 40 keV , $5 \times 10^{13}\text{ cm}^{-2}$. La componente del ion se corresponde con el ion implantado (modelo “+1”) y FP con el dañado (FPs) generado por el ion. Líneas continuas: evolución obtenida leyendo las coordenadas de la implantación BCA. Líneas punteadas: evolución obtenida leyendo los perfiles implantados y generando coordenadas aleatorias.

través de una función de números aleatorios, queda claro que se pierde toda la correlación que pudiere haber en los FP. También hemos comparado simulaciones hechas sólo con el ion con otras en las que sólo se introduce el dañado, para ver las diferentes contribuciones de cada uno de ellos a la supersaturación (el cociente entre la concentración de I_s y la que habría en equilibrio). Esta evolución temporal de la supersaturación de I está en la figura 11.8. El exceso de I_s va disminuyendo a medida que éstos se van recombinando con V_s o bien son aniquilados en la superficie y, como los eventos de recombinación IV son básicamente locales, y por tanto rápidos, es el exceso de I_s que se dirigen hacia la superficie la principal contribución a la supersaturación.

De la figura se ve también que las integrales de la supersaturación para el ion y para el dañado (FPs) dan, más o menos, la integral para el caso total FP+ion, si bien para este caso la supersaturación se retrasa un poco y tiene sus valores altos más desplazados hacia el final. Se puede explicar este resultado como causa del crecimiento de los clusters de intersticiales hacia clusters más grandes y tamaños más estables. Así pues, a la vista de la figura 11.8 podemos concluir que para las condiciones estudiadas:

- La correlación espacial de los pares IV no es relevante, puesto que se pierde al leer los perfiles y aún así se obtiene una supersaturación similar. Esto permite validar el uso de perfiles externos de I_s y V_s , que son mucho más rápidos de generar (cuando se usan aproximaciones analíticas) que las cascadas de implantación de los programas BCA.

- La supersaturación debida a los iones es dominante sobre aquella debida a los FPs, corroborando el modelo “+1”.

Se ha realizado una comprobación extra de la recombinación local de la mayor parte de I_s y V_s llevando a cabo otra simulación en la que los I_s y V_s son sólo el exceso local de I_s o V_s . En este caso, también la supersaturación tiene valores similares en su evolución, demostrando que el número de saltos que hacen los I_s para encontrarse con los V_s es despreciable. Además, en esta simulación el número neto de I_s y V_s es mucho menor que el número total, lo cual permite realizar simulaciones que requieran menos recursos.

► 11.3.6

Implantaciones pequeñas

Cuando se hacen recocidos de perfiles $I - V$ bastante ruidosos, es decir, en los cuales se han simulado muy pocas cascadas (Fig. 11.9), la supersaturación que se obtiene es muy similar a la que produce una simulación de implantación BCA mucho más grande, lo cual indica que el $I - V$ neto necesario para conseguir la supersaturación correcta puede estimarse a partir de una simulación ruidosa, pero muy rápida, de BCA. El proceso de randomizado que genera la difusión suaviza las fluctuaciones y ruidos de la simulación BCA, y el perfil $I - V$ ruidoso sigue aportando información suficiente sobre los mecanismos dominantes que controlan la TED.

► 11.3.7

Importancia de los clusters de vacantes

Cuando se simula el recocido del dañado en la aproximación de continuo es ventajoso, para reducir el número de ecuaciones, no incluir los clusters de V_s , ya que generalmente su presencia no altera significativamente la supersaturación de I_s . En nuestro caso, haciendo simulaciones con y sin el modelo de clusters de V_s , hemos verificado que la evolución temporal de la supersaturación es esencialmente la misma. Para energías de implantación medianas y pequeñas los clusters de V_s se sitúan cerca de la superficie, y el número de saltos que dan los I_s para ser aniquilados en la superficie o en un cluster de V es muy similar.

Para dosis muy bajas, para la cual las cascadas no se solapan, aparecen otros asuntos además de la correlación de los pares de Frenkel: la localización individual de la cascada no está correctamente reproducida por la distribución aleatoria, pero uniforme, de los perfiles de concentración (Jaraiz et al., 2000), por lo que para estas dosis es de prever que simulaciones hechas leyendo perfiles o leyendo coordenadas puedan diferir.

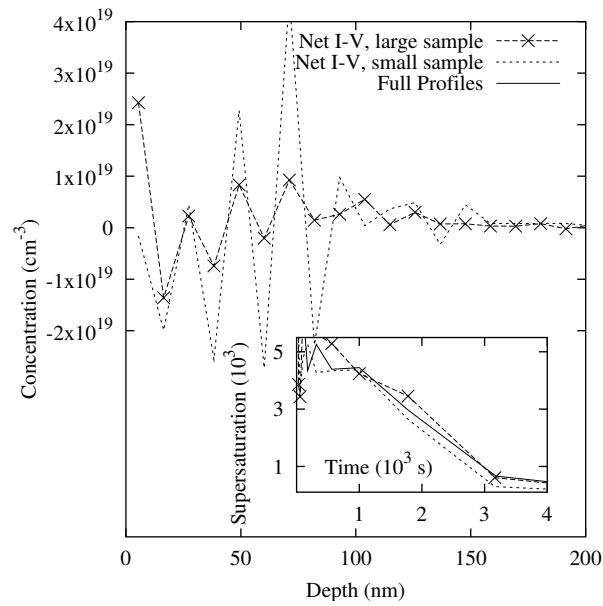


Figura 11.9: Línea punteada: perfil $I - V$ muy ruidoso proveniente de una simulación de implantación BCA pequeña (≈ 110 cascadas) de silicio en silicio 40 keV, $5 \times 10^{13} \text{ cm}^{-2}$. Símbolos: el mismo perfil para 14500 cascadas. Figura insertada: la supersaturación para el muestreo de $I - V$ pequeño (línea punteada), para el largo (símbolos) y para los perfiles completos de I, V , es prácticamente la misma.

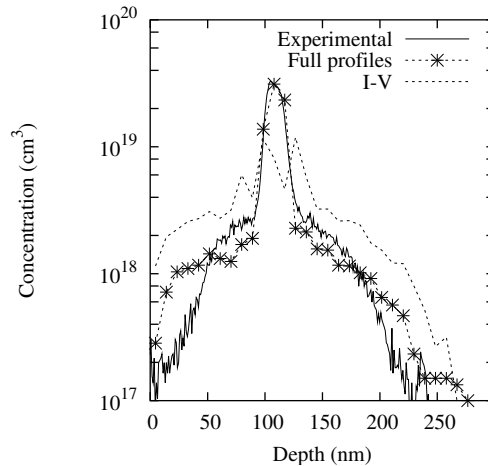


Figura 11.10: Perfiles simulados y experimentales de una *spike* de boro de $4,5 \times 10^{19} \text{ cm}^{-3}$ después de una implantación de silicio de 40keV, $9 \times 10^{13} \text{ cm}^{-2}$ y 50 s de recocido a 800 °C. Línea continua: datos experimentales (Pelaz et al., 1997). Línea discontinua: simulaciones hechas con los perfiles complejos de I, V o sólo con los perfiles netos de $I - V$. Hay que destacar que los perfiles netos de $I - V$ no aportan partículas suficientes para inmovilizar la *spike* de boro.

► 11.3.8

Presencia de impurezas

Por último, hay que ver cómo la presencia de impurezas afecta los resultados del recocido. Como ejemplo vamos a considerar el boro, del cual se sabe que reacciona con I para crear clusters BI (BIC), los cuales tienen una importancia capital en la difusión del boro (Pelaz et al., 1997). De esta manera se han hecho más simulaciones para clarificar cómo la presencia de impurezas pudiera afectar a los resultados anteriormente presentados. La interacción entre los átomos de B y los intersticiales del silicio se describe en términos de “kick-out” y mecanismos de clusterización (Pelaz et al., 1997), donde hemos tomado valores numéricos que produzcan una difusividad acorde con la referencia (Mathiot and Pfister, 1984).

La Fig. 11.10 muestra una *spike* de boro, inicialmente de 100 a 120 nm y con una concentración de $4,5 \times 10^{19} \text{ cm}^{-3}$, recocida durante 50 s a 800 °C después de una implantación de silicio de 40 keV, $9 \times 10^{13} \text{ cm}^{-2}$. Se ha hecho la misma simulación: (a) usando los perfiles completos de la implantación para I y V y (b) utilizando sólo los perfiles netos de $I - V$. El recocido ha sido hecho siempre en una caja de simulación kMC de $80 \times 80 \times 350 \text{ nm}^3$. A su vez, los perfiles completos se han calculado dos veces, una tomando la energía necesaria para generar un FP en el simulador BCA en $E_b = 15 \text{ eV}$ y la otra en $E_b = 20 \text{ eV}$. A la vista de la figura 11.10 se puede concluir, de nuevo, que las coordenadas originales de los I s y V s no son importantes, puesto que los perfiles de I y V dan un recocido

correcto. Además, aunque el dañado que generó la simulación con $E_b = 20$ eV fue un 20 % más pequeño que aquél con $E_b = 15$ eV, el recocido final resultó ser el mismo (aunque por claridad en la figura sólo se muestra uno de los dos). No obstante, el exceso neto de I s y V s ($I - V$) claramente no es suficiente, y se requiere la concentración total de ellos, debido a que la formación de los BICs ocurre únicamente para concentraciones muy altas de I , que no se dan con el perfil neto de I s.

▷ 11.3.9

Conclusiones

Aunque el presente estudio se limita únicamente a un rango determinado de condiciones de implantación y recocido, ha ayudado a clarificar cuáles son los mecanismos más relevantes durante una implantación para la predicción precisa de su TED tras su posterior dañado. Para resumir este estudio diremos que:

- La correlación espacial entre el I y el V de un par de Frenkel no es importante.
- El exceso neto de I s y V s es capaz de dar una supersaturación correcta.
- Simulaciones de implantación muy ruidosas pueden dar resultados correctos.
- La presencia de clusters de vacantes no es relevante: esto reduce el número de ecuaciones en el modelado de especies difundidas mediante I s en los modelos de los simuladores del continuo.
- En simulaciones con especies que difundan a través de I y presenten clusters (como el boro), es necesario utilizar los perfiles completos de I y V , y no sólo el exceso neto, porque para inmovilizar las impurezas en clusters se requiere la concentración total de I s y V s.
- Finalmente, una regla práctica para los simuladores de procesos podría ser: se ejecuta una simulación pequeña y rápida tipo BCA para conseguir los perfiles de I y V , que pueden ser luego usados directamente en simulaciones de kMC con cajas grandes, o bien ser suavizados para su uso en simuladores del continuo.

▷ 11.4

Modelizado físico de los efectos del nivel de Fermi

Este trabajo se encuentra también en la referencia (Martin-Bragado et al., 2004c).

Modelizado físico de los efectos del nivel de Fermi para simuladores de dispositivos nanométricos

Resumen

Se va a exponer una aproximación física para el modelizado del nivel de Fermi que ha sido diseñada para ser precisa, a la par que implementable, en simuladores de procesos de dispositivos. En ella usamos un simulador atomístico de Monte Carlo cinético, junto con un tratamiento continuo de las densidades de carga. El modelo incluye:

- Reacciones de carga y fuerzas eléctricas de acuerdo al nivel de Fermi local.
- Reacciones de emparejamiento y ruptura para partículas cargadas.
- Desactivación/clustering de dopantes.
- Solubilidad dependiente del nivel de Fermi.

El modelo también incluye estadística para semiconductores degenerados, procesos de estrechamiento del gap y compensación eléctrica para el dañado. Los parámetros usados para las partículas cargadas coinciden con cálculos ab-initio y resultados experimentales.

El modelo presentado ha demostrado ser muy eficiente en la simulación realista de procesos de dispositivos. Por último se presentarán algunos ejemplos ilustrativos para dopantes comunes, como el boro y el arsénico, y se discutirá el potencial de esta aproximación para la simulación precisa de dispositivos CMOS de tamaño nanométrico.

▷ 11.4.1

Introducción

La tecnología CMOS actual, de unas pocas décimas de nanómetro, tiene que resolver la complejidad surgida por la necesidad de unos dopados muy altos, un movimiento térmico reducido (condiciones de no equilibrio) y la presencia de efectos tridimensionales. Con estas condiciones resulta ventajoso ser capaz de implementar, de manera directa, los mecanismos físicos subyacentes y, muy en

particular, los fenómenos de difusión extrínseca que generan las altas concentraciones de dopado. (Fahey et al., 1989; Fair, 1981b) Otros temas también relacionados con la carga son la solubilidad, que depende del nivel de Fermi (Northrup and Zhang, 1993; de Walle et al., 1993), la desactivación por medio de clusters de los dopantes (Pelaz et al., 1999), y la compensación eléctrica que crean los altos dañados.

El método de Monte Carlo cinético sin red (non Lattice kMC), está demostrando ofrecer una descripción detallada y comprensible de la física de los procesos tecnológicos, y además es capaz de simular dispositivos de tamaños reales en tiempos razonables (Jaraiz et al., 1998; Jaraiz et al., 2001). En este trabajo se pretende describir un modelo con bases físicas del nivel de Fermi para kMC atómico. Esta aproximación ha sido diseñada para ser implementada en el simulador de procesos DADOS (Moroz et al., 2002; Synopsys, 2003).

A lo largo del artículo nos centraremos primero en los modelos físicos, después revisaremos brevemente la forma de implementarlos y finalizaremos con algunos ejemplos.

▷ 11.4.2

Modelos físicos

En nuestro esquema de simulación las partículas y defectos son tratados de manera atómica, mientras que las concentraciones de carga son tratadas con teoría del continuo. Como las reacciones de carga son mucho más rápidas que las estructurales (Rafferty, 1997), es posible suponer que va a haber equilibrio eléctrico incluso cuando las concentraciones de partículas estén lejos del equilibrio y, consecuentemente, el nivel de Fermi (e_F) estará siempre bien definido.

En esta sección se va a adaptar la formulación de los efectos del nivel de Fermi en la difusión (Fahey et al., 1989) a un escenario atómico, a la vez que se introducirán algunos conceptos que no están presentes de manera explícita en los formalismos de modelos del continuo.

Estados de carga para defectos puntuales sencillos

Tomemos la vacante (V) como ejemplo de un defecto puntual sencillo. La vacante del silicio tiene múltiples estados de carga, que son V^0 , V^- , V^{--} , V^+ , y V^{++} , cuyos valores están listados en la tabla 11.1. En ella estamos representando con e_{V^-} , $e_{V^{--}}$, e_{V^+} , y $e_{V^{++}}$, los niveles energéticos asociados a las transiciones de carga entre V^0 y V^- , V^- y V^{--} , V^0 y V^+ , y V^+ y V^{++} , respectivamente. Con ello, las concentraciones relativas serán:

$$\begin{aligned}
 [V^-]/[V^0] &= \exp((e_F - e_{V^-})/k_B T) \\
 [V^+]/[V^0] &= \exp((e_{V^+} - e_F)/k_B T) \\
 [V^{--}]/[V^0] &= \exp((2e_F - e_{V^-} - e_{V^{--}})/k_B T) \\
 [V^{++}]/[V^0] &= \exp((e_{V^+} + e_{V^{++}} - 2e_F)/k_B T)
 \end{aligned} \tag{11.26}$$

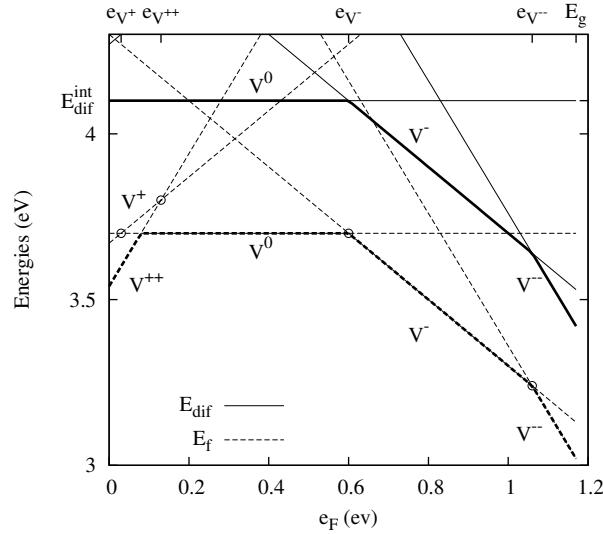


Figura 11.11: Energías de formación (E_f , líneas punteadas) y de difusión ($E_{dif} = E_f + E_m$, líneas continuas) de la vacante de silicio para distintas cargas (V^j) en función del nivel de Fermi (e_F). Se han usado los parámetros listados en la tabla 11.1. Se mide e_F en el borde de la banda de valencia, y se toma $E_g = 1.17$ eV como gap para una temperatura $T = 0$ K. Los valores mínimos para $E_f(V^j)$ y $E_{dif}(V^j)$ están dibujados con líneas gruesas. E_{dif}^{int} se corresponde con la energía de difusión para el silicio en condiciones intrínsecas ($e_F = e_i$) y los círculos indican los cruces de energías correspondientes a los distintos niveles de carga e_{V^j} .

donde $k_B T$ es la energía térmica, y se está suponiendo que todos los estados de carga presentan el mismo factor de degeneración (Fahey et al., 1989). La concentración de V^0 en condiciones de equilibrio (representada como $[V^0]^*$) está controlada por su energía de formación, $E_f(V^0)$, que a su vez es independiente del nivel de Fermi, puesto que es una partícula neutra. Para los otros estados de carga, que no son neutros, la dependencia de la energía de formación con el nivel de Fermi, junto con sus concentraciones en equilibrio, pueden obtenerse de las ecuaciones 11.26. En concreto, las dependencias de las energías de formación con e_F para las vacantes cargadas, utilizando los valores de la tabla 11.1 se encuentran representadas en la figura 11.11, donde hemos tomado el origen de los niveles de energía en el borde de la banda de valencia. La concentración total de vacantes en función de e_F estará determinada, en primer lugar, por el estado de carga con una E_f más baja (línea punteada gruesa en la figura), y se ve también que V^+ es un estado meta-estable, puesto que no representa el mínimo de E_f para ningún valor de e_F .

Las ecuaciones 11.26 siguen siendo válidas fuera del equilibrio lo que implica que la supersaturación de las vacantes, definida como $S_V = [V]/[V]^*$, es la misma para todos los estados de carga: $[V^j]/[V^j]^* = [V^0]/[V^0]^* = S_V$.

| | V^{--} | V^{-} | V^0 | V^{+} | V^{++} | I^{-} | I^0 | I^{+} | B_i^{-} | B_i^0 | B_i^{+} |
|---|----------|---------|-------|---------|----------|---------|--------|---------|-----------|---------|-----------|
| $D_{m,0}(\times 10^{-3} \text{ cm}^2/\text{s})$ | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 1 | 1 | 1 |
| E_m (eV) | 0.4 | 0.4 | 0.4 | 0.6 | 0.8 | 1 | 1 | 0.8 | 0.5 | 0.2 | 1.1 |
| $e_t(T=0)$ (eV) | 1.06 | 0.6 | | 0.03 | 0.13 | 1 | | 0.2 | 0.72 | | 1.02 |
| $[X^0]_0^*(\times 10^{25} \text{ cm}^{-3})$ | | | 1.5 | | | | 10^3 | | | | |
| E_f (eV) | | | 3.7 | | | | 3.9 | | | | |

Tabla 11.1: Parámetros utilizados para los estados cargados de la vacante de silicio (V^j), su intersticial (I^j), y el boro intersticial (B_i^j). D_0 es el prefactor de la difusividad (ecuación 11.28), E_m la energía de migración y e_t el nivel de carga medido desde el borde de la banda de valencia. Estamos suponiendo que e_t escala con la energía del gap como $e_t(T) = e_t(0)E_g(T)/E_g(0)$ (para $T \neq 0$, ver Ref. (Sze, 1981)). Los prefactores para las energías de formación y concentraciones en equilibrio de I^0 y V^0 también se muestran. Otros parámetros que se están usando, pero no están en la tabla, son $D_{bk,0}(B_i^-) = 5 \text{ cm}^2/\text{s}$ y $E_b(B_i^-) = 0,4 \text{ eV}$. $D_{bk,0}$ es el prefactor de ruptura, expresado en unidades de difusividad ($= (\lambda^2/6) \nu_{bk,0}$). $D_{bk,0}(B_i^0)$ y $E_b(B_i^0)(T)$ pueden obtenerse de las ecuaciones 11.32 y 11.33. Por último decir que se ha tomado la distancia de salto λ como la distancia a segundos vecinos (0,384 nm) y el volumen efectivo de captura en $v_{\text{capt}} = 3\lambda^3$, el valor para nuestra implementación.

La frecuencia de migración ν_m de la vacante en un determinado estado de carga j está dada por:

$$\nu_m(V^j) = \nu_{m,0}(V^j) \times \exp(-E_m(V^j)/k_B T) \quad (11.27)$$

donde $\nu_{m,0}$ es el prefactor de migración, y E_m la energía de migración. La difusividad de V^j es, a su vez:

$$D(V^j) = \lambda^2 \nu_m(V^j)/6, \quad (11.28)$$

donde λ es la distancia de salto. En condiciones de equilibrio el producto

$$D(V^j)[V^j]^*$$

será proporcional a

$$\exp(-E_{\text{dif}}(V^j)/k_B T),$$

siendo

$$E_{\text{dif}} = E_m + E_f$$

la energía de difusión, que también se representa en la figura 11.11. El flujo de difusión estará dominado por el valor mínimo de E_{dif} (línea gruesa en la figura 11.11). Si se usan los parámetros de la tabla 11.1 se obtiene un buen ajuste de las medidas de difusividad expuestas en la referencia (Giese et al., 2000) para la V .

Los niveles de carga para los intersticiales del silicio no están aún establecidos de forma concluyente (Fahey et al., 1989; Zhu, 1998; Watkins, 2000), parece

que el I^+ fuera meta-estable, pero con una energía de migración tan baja que probablemente controle la difusión de intersticiales para materiales de tipo p (Harrison, 1998). Además, para simplificar vamos a ignorar el estado I^{++} (Fahey et al., 1989), con lo que se describirá el I utilizando únicamente los estados I^+ , I^0 y I^- , con los valores que se listan en la tabla 11.1. Estos parámetros han sido establecidos para ajustar los datos experimentales tanto de la difusión del I (Bracht et al., 1995) como de la auto-difusión total del silicio. (Bracht et al., 1998).

El modelo de deriva electrónica en kMC

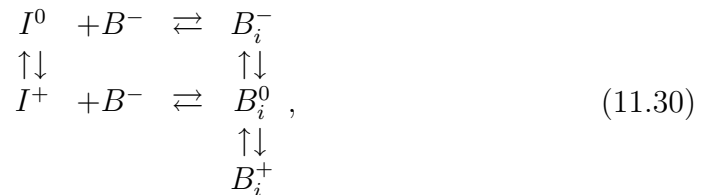
La deriva electrónica aparece como una consecuencia natural de la dependencia local de las energías de formación de las partículas cargadas. Dentro del entorno del método kMC, la relación entre la probabilidad de salto hacia las direcciones positiva y negativa del eje x para un defecto puntual que está migrando y presenta una carga jq será:

$$\frac{P_{+x}}{P_{-x}} = \exp\left(\frac{jq\mathcal{E}_x\lambda}{k_B T}\right) = \exp\left(-\frac{j}{k_B T} \frac{de_F}{dx}\right), \quad (11.29)$$

donde $\vec{\mathcal{E}}$ es el campo eléctrico y q el valor absoluto de la carga del electrón. A partir de la ecuación 11.29 es posible derivar la componente x de la velocidad de deriva como $v_x = \lambda(P_{+x} - P_{-x})/6$ la cual, en una aproximación de primer orden, queda como la relación de Einstein $v_x = jqD\mathcal{E}_x/(k_B T)$. Esta deriva es importante y tiene que estar adecuadamente representada para poder reproducir el hecho de que, en condiciones de equilibrio, las concentraciones de $[V^0]^*$ y $[I^0]^*$ deben de ser constantes incluso cuando $\nabla e_F \neq 0$.

Estados de carga para pares de defectos puntuales.

Vamos a considerar ahora las situaciones en las cuales los defectos puntuales están constituidos por un par AX , siendo A una impureza y X un I o una V . Sirva para ilustrar el caso el boro intersticial B_i (que, en el caso de nuestro simulador kMC sin red es equivalente al par BI). Se sabe que el B migra principalmente debido a un mecanismo intersticial, vía B_i , en vez de hacerlo a través de la vacante V . Los estados de carga del B_i son B_i^- , B_i^0 y B_i^+ . Las reacciones de emparejado y ruptura, así como las transiciones de carga relacionadas con el B_i pueden ser representadas por



donde se está suponiendo que el boro es inmóvil y siempre está ionizado (B^-). En las ecuaciones anteriores las reacciones horizontales (emparejado y ruptura) conservan la carga, mientras que las verticales establecen el equilibrio electrónico. La ruptura directa del B_i^+ no está considerada porque, como se ha dicho más arriba, no se toma el estado I^{++} . Los valores para los niveles de carga tomados para B_i se encuentran listados en la tabla 11.1. Estos valores han sido medidos por caracterización eléctrica. (Harris et al., 1982; Harris et al., 1987; Watkins, 2000).

Desarrollando la primera reacción de la ecuación 11.30 en condiciones de equilibrio local:

$$\frac{[B_i^-]}{[B^-]} = v_{\text{capt}}[I^0] \frac{\nu_m(I^0)}{\nu_{bk}(B_i^-)}, \quad (11.31)$$

siendo v_{capt} el volumen efectivo de captura para la reacción de emparejamiento y ν_{bk} la frecuencia de ruptura. El cociente de la ecuación 11.31 es independiente de e_F , puesto que las cargas de B_i^- y B^- son iguales. En contraste, $[B_i^0]/[B^-]$ y $[B_i^+]/[B^-]$ sí que dependen del nivel de Fermi. La energía de activación para la ruptura de B_i^- será $E_b(B_i^-) + E_m(I^0)$, donde E_b denota la energía de enlace.

Considerando la conservación de la energía en las reacciones expuestas en la ecuación 11.30, podemos obtener que:

$$E_b(B_i^0) = E_b(B_i^-) + e_{B_i^-} - e_{I^+}, \quad (11.32)$$

y la energía de activación para la ruptura de B_i será $E_b(B_i^0) + E_m(I^+)$. De las condiciones de equilibrio se puede también derivar la relación que han de satisfacer los prefactores

$$\nu_{bk,0}(B_i^0)/\nu_{bk,0}(B_i^-) = \nu_{m,0}(I^+)/\nu_{m,0}(I^0). \quad (11.33)$$

En la figura 11.12 se han representado las energías de formación de B_i^j a partir de un B^- ($E_f(B_i^j) - E_f(B^-)$) en función de e_F (línea discontinua), que coincide con la energía de activación del cociente $[B_i^j]/[B^-]$. Dicha figura presenta un acuerdo notable con la imagen que se extrae de cálculos ab-initio. (Windl et al., 1999; Sadigh et al., 1999).

La difusividad efectiva del boro, $D(B)$ está dada por la suma de la contribución de todas las especies móviles

$$D(B) \approx D(B_i^-) \frac{[B_i^-]}{[B^-]} + D(B_i^0) \frac{[B_i^0]}{[B^-]} + D(B_i^+) \frac{[B_i^+]}{[B^-]}, \quad (11.34)$$

que los modelos de continuo, utilizando la aproximación de Maxwell-Boltzmann, escriben generalmente como

$$D(B) = D_X(B) + D_P(B) \frac{p}{n_i} + D_{PP}(B) \left(\frac{p}{n_i} \right)^2, \quad (11.35)$$

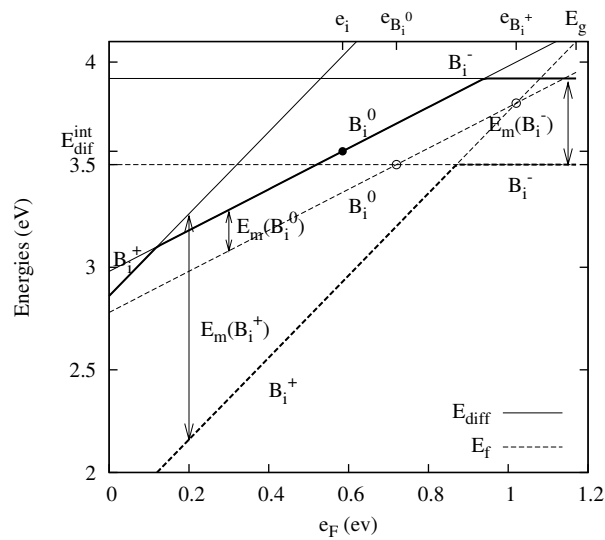


Figura 11.12: Líneas discontinuas: energía de formación de B_i^j a partir de B^- ($= E_f(B_i^j) + E_f(B^-)$) en función del nivel de Fermi y usando los valores de la tabla 11.1. Línea continua: energía de activación de la contribución de B_i^j a la difusividad efectiva del boro en condiciones de equilibrio, E_{diff} (ver ecuación. 11.34). Las energías de migración ($E_m(B_i^j)$) son las diferencias entre las líneas continuas y discontinuas. El punto indica el valor de E_{diff}^{int} y el resto de notación es como la de la figura 11.11.

donde p y n_i son la concentración de huecos y la concentración intrínseca de electrones, respectivamente. De esta manera las relaciones entre las componentes mencionadas de la difusividad, $D_X(B)$, $D_P(B)$, $D_{PP}(B)$ y los parámetros microscópicos son:

$$\begin{aligned} D_X(B) &= v_{\text{capt}} \nu_{m,I^0} [I^0] (\nu_m(B_i^-) / \nu_{bk}(B_i^-)) \\ D_P(B) &= v_{\text{capt}} \nu_{m,I^0} [I^0] (\nu_m(B_i^0) / \nu_{bk}(B_i^-)) \exp((e_{B_i^-} - e_i) / k_B T) \\ D_{PP}(B) &= v_{\text{capt}} \nu_{m,I^0} [I^0] (\nu_m(B_i^+) / \nu_{bk}(B_i^-)) \exp((e_{B_i^-} + e_{B_i^+} - 2e_i) / k_B T) \end{aligned} \quad (11.36)$$

donde e_i es el nivel intrínseco. Estas expresiones puede usarse como un puente entre los parámetros de los simuladores de continuo y los necesitados por simuladores tipo kMC.

Las energías de activación de los términos de la ecuación 11.34 (a las que se suele conocer como “energías de difusión asistida por defectos” o, simplemente, “energías de difusión”, $E_{dif}(B_i^j)$) están representadas en la figura 11.12: se puede ver que B_i^0 es un estado meta-estable (ya que $e_{B_i^+} > e_{B_i^-}$), pero en cualquier caso resulta ser el responsable principal de la difusión del boro para un rango grande de e_{FS} y, en consecuencia, $D(B) \propto p/n_i$ (Fair, 1981b). De acuerdo con la figura 11.12 la energía de migración efectiva de B_i para $e_F \approx e_i$ es $E_{dif}(B_i^0) - E_f(B_i^+) \approx 0,6$ eV, que coincide bien con resultados experimentales (Watkins, 1975).

Los parámetros que se listan en la tabla 11.1 han sido elegidos para ajustar bien los *plot* de Arrhenius” de la difusividad del boro en equilibrio, $D(B)^*$, citada en (Mathiot and Pfister, 1984). A su vez la figura 11.13 compara estos valores con otros ajustes expuestos en (Fair, 1981b; Plummer et al., 2000).

Hay que tener en cuenta que en la descripción anterior no se ha hecho ninguna presunción de equilibrio local para las partículas (hemos considerado equilibrio para derivar las relaciones, pero los resultados que se obtienen son generales).

Se puede hacer un análisis similar para el arsénico, pero teniendo en cuenta la contribución doble, tanto del intersticial como de la vacante, relacionada con As_V y As_i , respectivamente. El conjunto de parámetros que utilizamos para el arsénico está expuesto en la referencia (Pinacho et al., 2004), que se corresponde con la sección 11.6.

Saltos largos en defectos con varios estados de carga

La distancia que viaja un defecto puntual compuesto antes de romperse (llamada generalmente “distancia de salto largo” Λ , está dada por (Cowern et al., 1990; Pinacho et al., 2002)

$$\Lambda = \lambda \sqrt{\frac{1}{6} \frac{\nu_m}{\nu_{bk}}}. \quad (11.37)$$

Así, para el ejemplo anterior del B_i con varios estados de carga, la ecua-

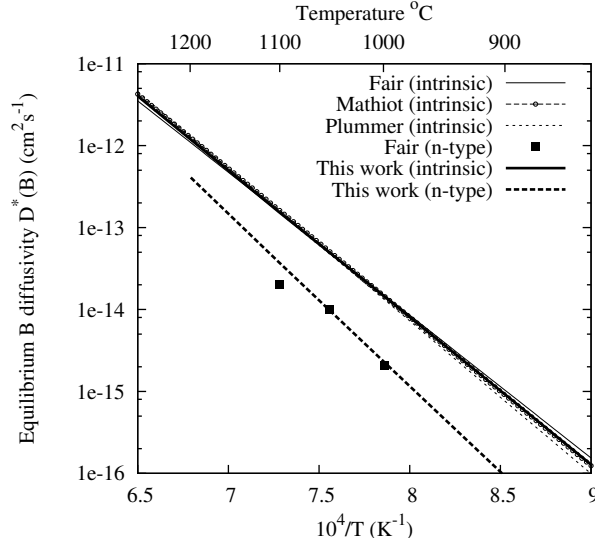


Figura 11.13: Difusividad efectiva del boro en condiciones de equilibrio, D_B^* . Los valores obtenidos con nuestro conjunto de parámetros están comparados con ajustes experimentales para materiales intrínsecos ((Refs. (Fair, 1981b; Mathiot and Pfister, 1984; Plummer et al., 2000) y con valores experimentales para materiales dopados tipo n (Fair, 1981b) ($N_D = 1,5 \times 10^{20} \text{ cm}^{-3}$).

ción 11.37 se transforma en

$$\Lambda = \lambda \sqrt{\frac{1}{6} \frac{\nu_m(B_i^-) \frac{[B_i^-]}{[B^-]} + \nu_m(B_i^0) \frac{[B_i^0]}{[B^-]} + \nu_m(B_i^+) \frac{[B_i^+]}{[B^-]}}{v_{capt}(\nu_m(I^0)[I^0] + \nu_m(I^+)[I^+] + \nu_m(I^-)[I^-])}}. \quad (11.38)$$

La figura 11.14 muestra la dependencia de Λ con la temperatura, usando los parámetros de la tabla 11.1. También se compara con valores experimentales tomados en materiales intrínsecos (Cowern et al., 1991; Cowern et al., 1992). Con los valores que tomamos resulta $\Lambda \propto \sqrt{p/n_i}$ cuando $e_F \simeq e_i$.

Es interesante destacar que las concentraciones de los diferentes estados de carga de los intersticiales (I^+ e I^-) no afectan de manera directa a la difusividad del boro (ecuaciones 11.34 y 11.35). Esto podría sugerir que, a modo de aproximación de simplificación, se podría despreciar la presencia de I^- e I^+ . Ahora bien, la ecuación 11.38 nos muestra que, en ese caso, se obtendrían unos valores de Λ incorrectos para materiales extrínsecos. De hecho, la magnitud de D_B está regida por el término $\nu_m(B_i^j)[B_i^j]$, pero su forma, su apariencia “gaussiana” en el perfil de difusión, depende también de los términos $\nu_m(I^j)[I^j]$.

Otros efectos

La solubilidad de las especies cargadas depende de e_F (Northrup and Zhang, 1993; de Walle et al., 1993). Así, para un aceptor como el boro, el límite de la

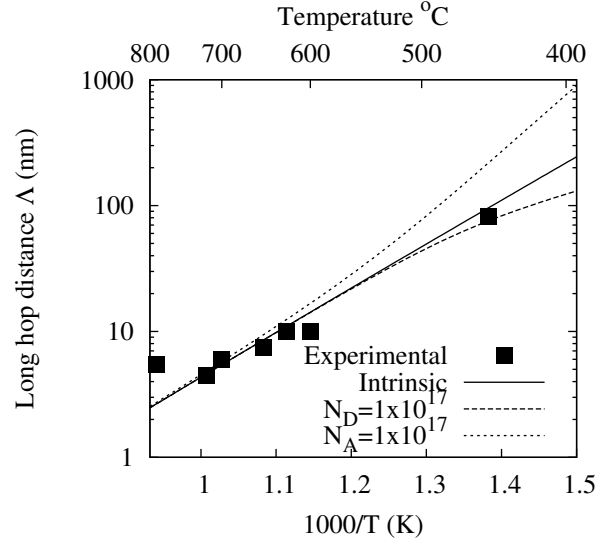


Figura 11.14: *Plot* de Arrhenius para los saltos largos del B_i (Λ). Líneas: valores calculados usando nuestro conjunto de parámetros para materiales intrínsecos y silicio dopado n y p . Símbolos: valores experimentales en condiciones casi-intrínsecas. (Cowern et al., 1991; Cowern et al., 1992).

solubilidad estará dado por:

$$[B^-]_{\max}(T) = [B^-]_{\max}^{\text{intrinsic}}(T) \times \exp\left(\frac{e_F - e_i}{k_B T}\right), \quad (11.39)$$

donde

$$[B^-]_{\max}^{\text{intrinsic}} = \frac{([B^-]_{\max}^{\text{extrinsic}})^2}{n_i} \quad (11.40)$$

es la solubilidad en condiciones intrínsecas ($e_F = e_i$), y $[B^-]_{\max}^{\text{extrinsic}}$ es la solubilidad cuando $[B^-]_{\max} \gg n_i$. La ecuación 11.39 es una consecuencia del hecho de que la energía de formación para las especies cargadas depende de e_F .

Respecto a la presencia de defectos extendidos, tales como clusters, estamos suponiendo que éstos son eléctricamente inactivos y son tomados en cuenta para la desactivación de dopantes. Se podrían incluir efectos de carga en los clusters, ya que en kMC no es un problema tener abundantes mecanismos funcionando a la vez, pero de momento no se dispone de valores para los parámetros físicos involucrados.

► 11.4.3

Implementación

En el método kMC atómico se representa cada defecto individual por una o más partículas, y los mecanismos físicos son descritos como “interacciones”

(reacciones entre partículas dentro del radio de captura, como el emparejamiento) y “eventos” (realizados por los defectos con cierta frecuencia, como una ruptura). La implementación que se va a describir aquí corresponde al simulador DADOS (Jaraiz et al., 2001; Moroz et al., 2002; Synopsys, 2003).

Como ya ha sido mencionado, los defectos extendidos van a ser considerados neutros en una primera aproximación, por lo que las dependencias con el nivel de Fermi estarán constreñidas e los defectos puntuales, y afectarán sólo de manera indirecta a los mencionados defectos extendidos. La tarea principal que debe acometer el simulador en su relación con los defectos puntuales cargados consiste en mantener en todo momento las proporciones locales adecuadas entre diversos estados de carga (ecuación 11.26), así como incluir los arrastres eléctricos (ecuación 11.29). Las interacciones electrostáticas entre las partículas están tomadas en cuenta a través del arrastre, y además no se permiten reacciones de emparejamiento entre partículas con el mismo signo, para reflejar las repulsiones eléctricas. Respecto a la interacción de los defectos puntuales con los extendidos y la superficie, ésta sólo se permite para los estados neutros, de tal manera que de una forma bastante natural se consigue la conducta descrita por la ecuación 11.39.

Para el cálculo de e_F (y por tanto, de las concentraciones de carga), usamos la aproximación de neutralidad de carga. El paso de los valores discretos de carga a las densidades está hecho a través de un radio de suavizado $r_D = (4\pi q^2 N/T)^{-1/2}$, que es igual al radio de Debye (Sze, 1981) y depende de la concentración de dopantes N . De esta manera, es posible que haya regiones cargadas, incluso con la aproximación de neutralidad de carga pero, cuando no se usan las distribuciones suavizadas, se observa una difusividad alta que es incorrecta, debido a la estimación excesiva de densidades de carga en las cercanías de los átomos de dopante.

Los defectos puntuales sustitucionales se consideran siempre activos (ionizados), lo cual es una aproximación buena para dopantes superficiales (boro y arsénico) incluso aunque su concentración sea muy alta (Synopsys, 2003), pero que debe ser replanteada para otro tipo de dopantes más profundos como el indio u otros. Además, para considerar la alta resistividad que hay en regiones de alto daño, se usan los intersticiales y vacantes como centros profundos de compensación, disminuyendo así el valor efectivo de la concentración de dopantes.

Por último, decir que se ha incluido la estadística de Fermi–Dirac, aunque también se puede usar la de Maxwell–Boltzmann (que se corresponde a la aproximación no degenerada), para comparar con resultados analíticos que usen esta última, y que el estrechamiento del ancho del gap se tiene en consideración a partir del modelo de Jain y Roulston (Jain and Roulston, 1991; Persson et al., 1999).

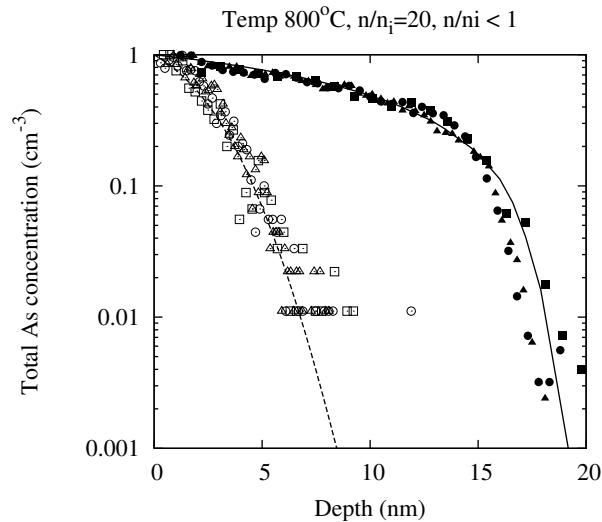


Figura 11.15: Perfiles de concentración de arsénico normalizados en condiciones intrínsecas y extrínsecas para distintos tiempos a 800 °C. Línea discontinua: difusión intrínseca calculada teóricamente. Símbolos huecos: perfiles simulados mediante kMC y normalizados con el tiempo, en condiciones intrínsecas. Línea continua: difusión teórica extrínseca, para $n = 20n_i$. Símbolos rellenos: perfiles atomísticos kMC normalizados con el tiempo para $n = 20n_i$. Para lograr una comparación adecuada hubo que usar estadística de Maxwell-Boltzmann en las simulaciones.

▷ 11.4.4

Ejemplos de simulación

Los esquemas de simulación que han sido presentados han demostrado su eficiencia en la simulación realista de procesos del tamaño de un dispositivo completo. La inclusión de todos los mecanismos que hemos ido mencionando únicamente incrementa el tiempo de simulación en un factor algo menor que 2, respecto a simulaciones sin efectos de carga, con tamaños típicos de 10^6 nm^3 .

La figura 11.15 muestra el buen acuerdo que hay entre un modelo basado en la resolución de ecuaciones diferenciales (continuo) y nuestro simulador, para el caso de la *in-diffusion* del arsénico tanto en condiciones intrínsecas como extrínsecas. Queda evidente en dicha comparación el efecto del nivel de Fermi en los perfiles de difusión y la profundidad a la que llegan. Esta comparación permite ver que la traducción de los parámetros de continuo a la formulación kMC atomística es correcta, (ecuación 11.36 y su equivalente para el arsénico).

Por último, la figura 11.16 comprueba la redistribución que sufre el boro tras ser recocido después de una implantación cerca de una unión N^+P (Hane et al., 2000). La concentración de boro disminuye en la parte n de la unión debido al arrastre eléctrico, que empuja al B^- hacia la parte P , aunque es interesante hacer notar que este aparente desplazamiento eléctrico está asociado realmente

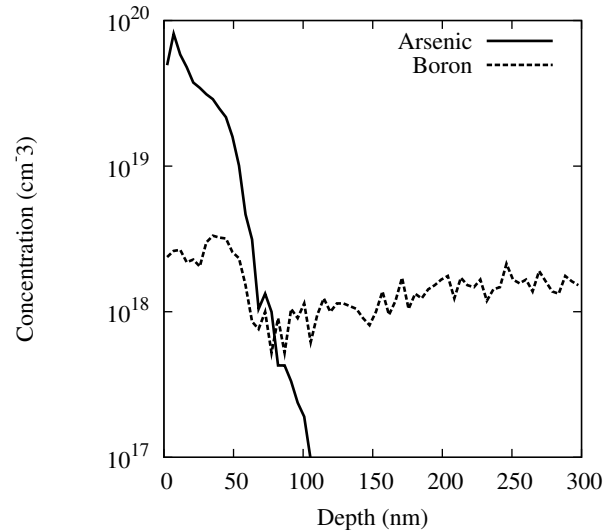


Figura 11.16: Redistribución de los dopantes en una unión PN . El boro se encontraba inicialmente con una concentración cercana a $2 \times 10^{18} \text{ cm}^{-3}$ (región P). En la región N^+ se ha implantado arsénico, 10 KeV, $5 \times 10^{14} \text{ cm}^{-2}$. Posteriormente se ha recocido toda la muestra durante 10 minutos a 750°C .

a un flujo neto de B_i^0 , que es neutro. Concretamente, el gradiente de e_F produce un gradiente de $[B_i^0]/[B^-]$ y, en consecuencia, un flujo de difusión de B_i^0 hacia la región N que tiende a homogeneizar $[B_i^0]$. Respecto al apilamiento de boro en la región rica en arsénico, está relacionado con el valor más bajo de $D(B)$ en un material de tipo N (ecuación 11.35).

▷ 11.4.5

Conclusiones

Se ha presentado el formalismo para la inclusión de los efectos del nivel de Fermi en un modelo atomístico kMC, y también se han expuestos las líneas maestras para su implementación. Este modelo ha sido ilustrado con el análisis de algunos casos para el boro y el arsénico. El método aquí presentado permite una descripción detallada de los fenómenos presentes en casos de alto dopado, y es aplicable a la simulación precisa de procesos presente en los dispositivos nanométricos.

▷ 11.5

Modelizado físico del dañado, amorfización y recristalización

Este trabajo se encuentra publicado en (Rubio et al., 2004).

Modelizado físico del dañado, amorfización y recristalización para simuladores predictivos de procesos con tamaño de dispositivo

Resumen

La ingeniería actual de drenador/fuente para CMOS supone el uso de implantaciones amorfizantes con una geometría tridimensional. Para el recocido, la difusión transitoria extra (TED) sólo se predice de forma precisa cuando la región amorfizada está correctamente modelizada, junto con la formación y evolución de los defectos puntuales, concretamente los $\{311\}$'s y los bucles de dislocación. Además de los defectos extendidos, que ya estaban correctamente modelizados en el simulador de Monte Carlo cinético DADOS, se ha desarrollado un modelo físico para el crecimiento del dañado durante la implantación, la amorfización y recristalización, capaz de manejar simulaciones del tamaño de un dispositivo entero. Este modelo está basado en los pockets amorfos (aglomerados tridimensionales de forma irregular con un número arbitrario de intersticiales y vacantes, junto con impurezas atrapadas) a los cuales se ha dotado de una energía de activación de la recombinación dependiente del tamaño del pocket amorfo. Este modelo es capaz de reproducir aspectos experimentales como pueden ser la temperatura de transición amorfo-cristal y el incremento super-lineal del dañado con la dosis. Se va a describir aquí el modelo y presentar algunos ejemplos de simulaciones. Por último, se discutirá también la eficiencia del modelo, en términos de tiempo de CPU y requerimientos de memoria.

▷ 11.5.1

Introducción

El modelizado predictivo de los actuales procesos de fabricación de dispositivos nanométricos se encuentra con el reto de tener que manejar muchos mecanismos muy dispares que tiene lugar a la vez, y que no pueden ser despreciados como una primera aproximación. Un ejemplo representativo es la implantación y posterior recocido del As para formar un drenador/fuente (Pinacho et al., 2004): no es suficiente con tener un buen modelo y buenos parámetros para la difusión y clustering del As, sino que también es necesario, entre otros temas de ciencia de

materiales, predecir la extensión de la región amorfizada para que el número y tamaño de los defectos extendidos que se formarán en la intercara amorfo/cristalina pueda ser estimado de forma correcta. Incluso en este caso, la escala temporal para la TED no estará correctamente simulada a no ser que el crecimiento de los defectos $\{311\}$ y su transformación en bucles de dislocación esté lograda aceptablemente bien. Por otra parte, para dosis bajas, el papel de la superficie para el atrapamiento y la segregación, también puede ser importante.

Como parte de esta tarea tan extensa, se van a describir y presentar en este trabajo algunos ejemplos de un modelo físico que resulta también eficiente (computacionalmente hablando) para el daño producido por implantación iónica, que sea capaz de dar predicciones precisas de algunas propiedades relacionadas con la amorfización (como la dependencia con la temperatura, velocidad de implantación de la dosis y masa del ion) pero que, aún así, sea capaz de manejar simulaciones del tamaño de un dispositivo completo nanométrico, utilizando tan sólo los recursos de un ordenador típico.

▷ 11.5.2

Descripción del modelo

En esta sección se van a describir los modelos que usamos para la producción y recocido del daño en el simulador de Monte Carlo cinético DADOS (Jaraiz et al., 2001). DADOS obtiene las coordenadas de las partículas (intersticiales, vacantes y átomos de impureza implantada) a través de las cascadas de simulación de MARLOWE. Después de leer una cascada, DADOS inserta secuencialmente cada una de las partículas en la caja de simulación. Cuando no hay vecinos dentro de la distancia del radio de captura, (distancia a segundos vecinos en el Si), de la nueva partícula insertada, dicha partícula se queda como un defecto puntual aislado (I , V) o una impureza, a lo que vamos a llamar en lo que queda de sección como IP. Si hubiera otra IP dentro de la distancia de captura se considera que las dos forman una nueva estructura de daño, denominada pocket amorfo o AP. Las partículas implantadas posteriormente se unirán a este AP si están dentro del radio de captura de cualquiera de las partículas que constituyen este AP. Hay que darse cuenta que se conserva a las partículas que están dentro del AP en su sitio original, por lo que estos APs presentan formas tridimensionales irregulares. Además, cuando en la simulación un nuevo AP se solape con otros APs o IPs existentes, se reagrupan todos en un único y más grande AP. Por simplicidad, vamos a considerar ahora los I s y V s y ya comentaremos el tema de las impurezas más adelante.

Se sabe también que los iones pesados tienden a producir cascadas más densas que los ligeros y, por tanto, una morfología de daño diferente. El modelo sencillo que vamos a ir describiendo es capaz de explicar esta propiedad, como puede observarse comparando los distintos histogramas de APs (número de APs frente a su tamaño) expuestos en la figura 11.17., y que se corresponden a iones de Si y Ge

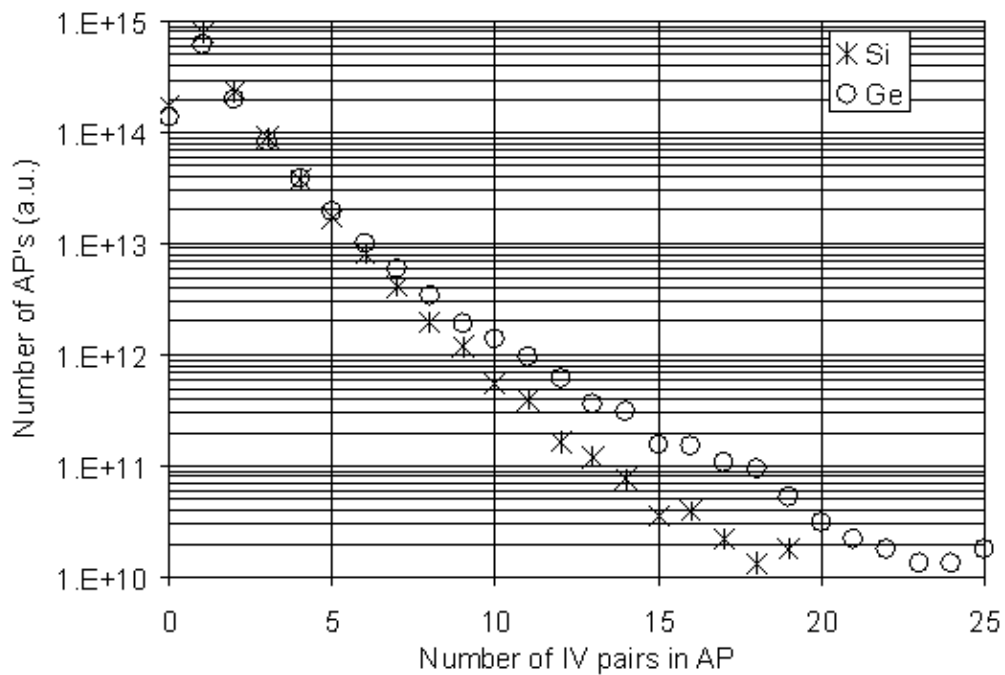


Figura 11.17: Histograma de pockets amorfos para implantaciones de iones de Si y Ge implantados a 80 keV de energía con una dosis de $4 \times 10^{12} \text{ cm}^{-2}$ (por debajo del nivel de amorfización) y a una velocidad de dosis de $5 \times 10^{12} \text{ cm}^{-2}\text{s}^{-1}$ a -200°C .

implantados en Si para las mismas condiciones de 80 keV de energía y velocidad de dosis de $5 \times 10^{12} \text{ cm}^{-2}\text{s}^{-1}$. La temperatura es lo suficientemente baja para evitar un recocido dinámico, y permitir así una acumulación de dañado realmente efectiva. En ambos casos se ha usado la misma dosis de $4 \times 10^{12} \text{ cm}^{-2}$, la cual ha sido elegida por debajo del nivel de amorfización para producir únicamente APs. Como se refleja en el histograma de la figura 11.17, los iones más pesados de Ge producen APs más largos que los de Si, donde se toma el número de APs como el de pares *IV* contenidos. Este número, a su vez, se limita a ser el mínimo número de *Vs* e *Is* que hay en el AP, independientemente de la distancia entre ellos. Si se hubiera tomado el número total de partículas *I+V* como tamaño del AP en vez del número de pares *IV* se habría visto que los histogramas exhibirían la misma tendencia, pero con un factor de dos en la abscisa (no está mostrado en la figura).

El par *IV* ha sido identificado en algunos cálculos teóricos (Tang et al., 1997) como un defecto meta-estable con una barrera para la recombinación. La aglomeración de múltiples pares *IV* (Marques et al., 2003) va formando estructuras de dañado más estables que exhiben una energía de recombinación que está entre la energía de recombinación de un par *IV* aislado (0.43 eV), y la energía de recristalización de la intercara plana amorfo/cristal (2.44 eV). Dicha aglomeración de pares *IV* para formar pockets amorfos ha sido utilizada ya para simular la amorfización y recristalización en silicio (Pelaz et al., 2003).

Nuestro modelo para la acumulación de dañado, amorfización y recristalización (Jaraiz et al., 2001) se basa en el comportamiento estadístico promedio de las regiones dañadas, en vez de en las configuraciones concretas locales de cada partícula *I* o *V* individual. La intención de nuestra aproximación es lograr simular estructuras con tamaños de dispositivos completos, cosa que requeriría un tamaño de memoria prohibitivo si fuera necesario almacenar cada uno de los pares *IV* que hay por debajo del nivel de amorfización. No obstante, hasta ahora sólo habíamos aplicado esta técnica a implantación a temperatura ambiente, pero en este artículo se van a presentar resultados después de recalibrar los parámetros de los APs utilizando datos de implantaciones a distintas temperaturas para poder así expandir el rango de aplicación del simulador. Como se verá, este modelo tan sencillo es suficiente para reproducir adecuadamente las dependencias con la masa del ion, la temperatura y la velocidad de la implantación de una manera eficiente (hablando en términos de tiempo de CPU y uso de memoria) para tamaños de dispositivos por debajo de la micra.

Para continuar con la descripción de nuestro modelo, la energía de activación de los APs para recombinarse se relaciona simplemente con el número de pares *IV* que contienen, sin tener en cuenta su configuración espacial particular. Después de que los APs recombinen todos los pares *IV* que contienen, éstos se disuelven a través de la emisión de defectos puntuales *I* o *V* que, de manera natural, se van convirtiendo en clusters de *Is* o *Vs* con su correspondiente dependencia de energía frente al tamaño. Cuando el AP ha atrapado también alguna impureza, después de recombinar todos sus pares *IV* se fragmenta en pequeños clusters

de impurezas (como As_nV_m , B_nI_m) con energías de formación bajas (Pinacho et al., 2004). Los APs que se solapan durante la implantación pueden dar lugar a regiones locales amorfas y por último, a capas enteras amorfas. Una vez que estas capas (regiones con forma arbitraria) son amorfas, el único evento posible para ellas es la recristalización, que es un proceso que ha sido implementado convirtiendo la acumulación de pequeñas cajas amorfas (del tamaño aproximado de 1nm^3) a cajas cristalinas a una velocidad activada térmicamente con energía 2.4eV , y depositando todos los átomos de impurezas que pudiera haber en la región amorfa como IPs (hasta el límite de la solubilidad eléctrica, para el caso concreto de los dopantes), o bien como pequeños clusters de defectos puntuales con impurezas, y también desplazando una fracción de dopantes junto con el frente de recristalización.

A continuación se va a describir como se han calibrado los parámetros de los APs, y se comentarán los resultados de las simulaciones.

► 11.5.3

Resultados

La figura 11.18a representa los niveles relativos de dañado ($I+V\text{ cm}^{-3}$) alcanzados tras la implantación de silicio en silicio con 80keV , 10^{15} cm^{-2} , frente a la temperatura de la implantación. Las curvas así representadas son similares a las que se obtienen con medidas RBS hechas por (Schultz et al., 1991) con energía de implantación 1MeV . La figura 11.18b es el *plot* de Arrhenius correspondiente y, además de los datos obtenidos en los experimentos de Schultz, incluye también los resultados experimentales obtenidos a 80keV por (Goldberg et al., 1995), quien determinó la temperatura a la que aparecen por primera vez capas amorfas cuando se irradia el silicio con iones con masas entre 12 (C) a 132 amu (Xe). Las energías de activación determinadas con los datos se encuentran entre 0.7 y 1.7eV y se incrementan con la masa del ion. También se encontró que la energía de activación para el silicio era de 0.84eV . En nuestro caso hemos usado los datos de Goldberg para calibrar la energía de recombinación de los APs en función del tamaño (número de IV s), asignando 0.5eV al tamaño de AP 1. Con nuestra asignación se observa un ajuste bueno para el rango completo de temperaturas y dosis, como puede verse en la figura 11.18b.

La figura 11.19 muestra la contribución de los diferentes estructuras de dañado y defectos presentes en el material, frente a la temperatura. Para bajas temperaturas se forma una capa amorfa continua en el punto de máximo dañado, rodeada de regiones muy dañadas, pero aún no amorfas, que contienen APs. Al final de la implantación a altas temperaturas sólo los clusters son lo suficientemente estables para que puedan ser vistos. Un incremento en la velocidad de la implantación da curvas similares a estas, pero desplazadas a temperaturas más altas.

En la figura 11.20 se representa la composición de dañado frente a la dosis de una implantación de (a) silicio y (b) dañado a baja temperatura y a igual velo-

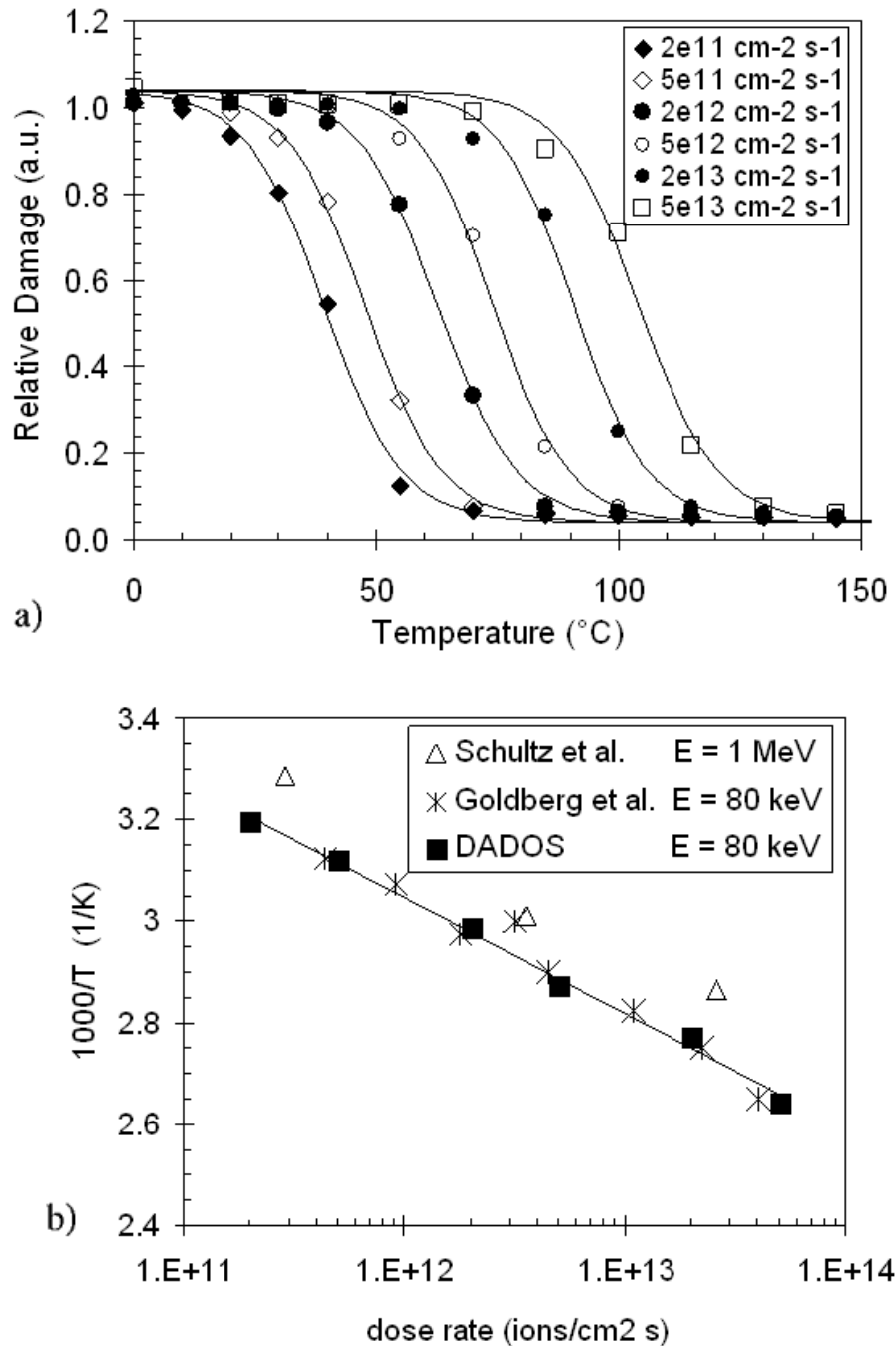


Figura 11.18: (a) Dañado calculado frente a la temperatura del sustrato, para implantaciones de Si en Si de 80 keV, 10^{15} cm^{-2} con velocidades de implantación entre $2 \times 10^{11} \text{ cm}^{-2} \text{ s}^{-1}$ y $5 \times 10^{13} \text{ cm}^{-2} \text{ s}^{-1}$. (b) Temperatura de transición cristal/amorfo en función de la velocidad de implantación. Los cuadrados se corresponden con las simulaciones en (a) y las estrellas y triángulos con los experimentos hechos a 80 keV por (Goldberg et al., 1995) y 1 MeV por (Schultz et al., 1991) respectivamente.

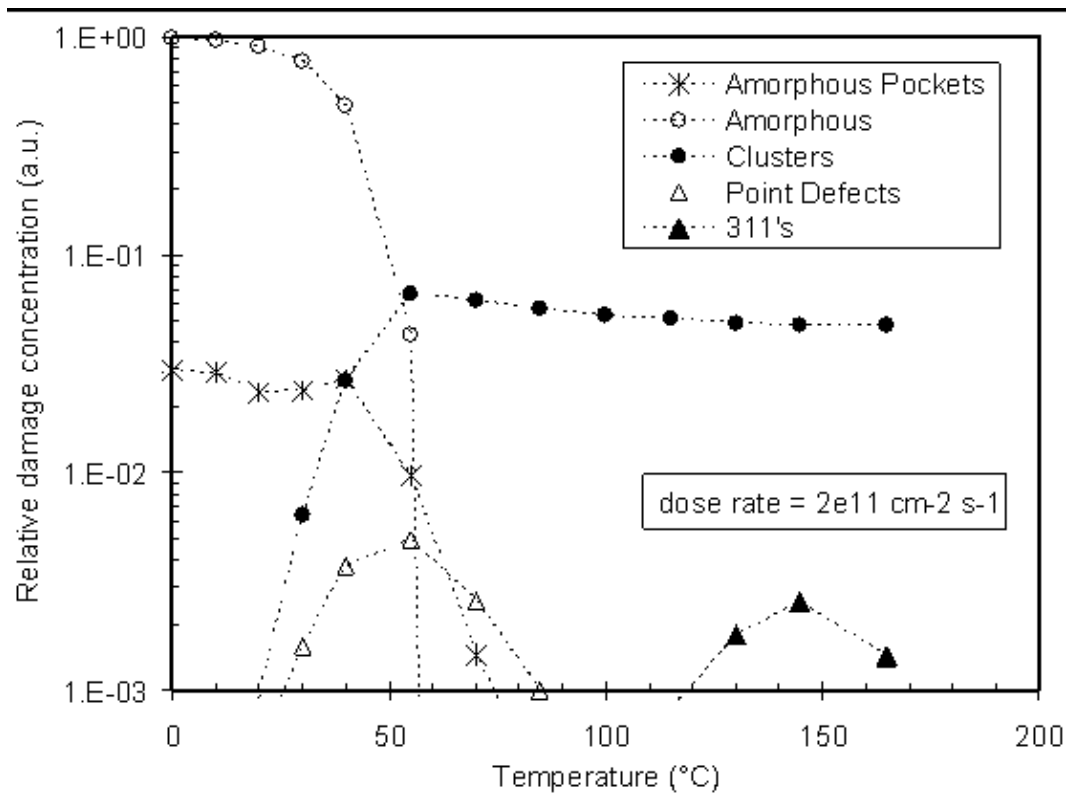


Figura 11.19: Contribución de las diferentes estructuras de defectos al dañado total para una implantación de Si en Si de 80 keV, 10^{15} cm^{-2} con una velocidad de implantación de $2 \times 10^{11} \text{ cm}^{-2} \text{ s}^{-1}$, en función de la temperatura de la muestra.

cidad, hasta lograr una dosis suficiente para amorfizar ambos casos. El comienzo de la amorfización tiene lugar a una dosis más baja para el Ge que para el Si, como era de esperar. Cabe señalar también la conducta super lineal del dañado total acumulado frente a la dosis.

Por último, y como ya se ha comentado, el propósito de nuestro modelo es ser capaz de disminuir drásticamente el uso de tiempo de CPU y memoria por parte del simulador al no tener que ir almacenando cada I o V generado por las cascadas hasta que se llega al nivel de amorfización (10^{22} cm^{-3}). Puesto que la frecuencia de recombinación de nuestros APs no se basa en su configuración local sino en su tamaño (cuenta de IV), se ha implementado la siguiente aproximación para reducir el uso de memoria sin afectar sensiblemente a la simulación: las partículas I s y V s van siendo incorporadas a la celda de simulación de DADOS hasta que se alcanza un cierto nivel de concentración (10^{20} cm^{-3}). Una vez por encima de este nivel se van ocultando las partículas aleatoriamente en el AP, pero éste mantiene una cuenta precisa del número de las partículas que lo componen a medida que nuevas partículas se van incorporando al AP o éste va recombinando algunos pares IV . Esta sencilla tarea de mantenimiento es suficiente para conservar la frecuencia adecuada de recombinación. Al mismo tiempo, la concentración de partículas cercana a 10^{20} cm^{-3} parece proveer a los APs de un número representativo de partículas suficiente para ir capturando los nuevos I s o V s que se van implantando y conseguir resultados de simulación correctos. Todo esto se puede ver en la figura 11.21, donde se comparan una simulación (a) hecha sin limitar la concentración de partículas hasta la amorfización y (b) limitando este número en la celda de simulación a un nivel de 10^{20} cm^{-3} , es decir, más de dos órdenes de magnitud por debajo. Pese a todo, el modelo predice prácticamente la misma composición y evolución del dañado.

▷ 11.5.4

Conclusiones

Se ha mostrado que es posible modelizar la acumulación de dañado hasta la amorfización en un método de Monte Carlo con un número de partículas en la simulación que es al menos dos órdenes de magnitud menor que el nivel real de amorfización. Además de la reducción drástica de requerimientos de CPU y memoria que esto supone, el modelo es capaz de predecir de forma precisa las dependencias con la temperatura y velocidad de implantación que tienen las diferentes estructuras de dañado dependiendo de la masa del ion. Se continúa haciendo más trabajo para calibrar el modelo para iones con otras masas. Este modelo se puede utilizar para simular las condiciones de amorfización de estructuras con el tamaño de dispositivos enteros por debajo de la micra.

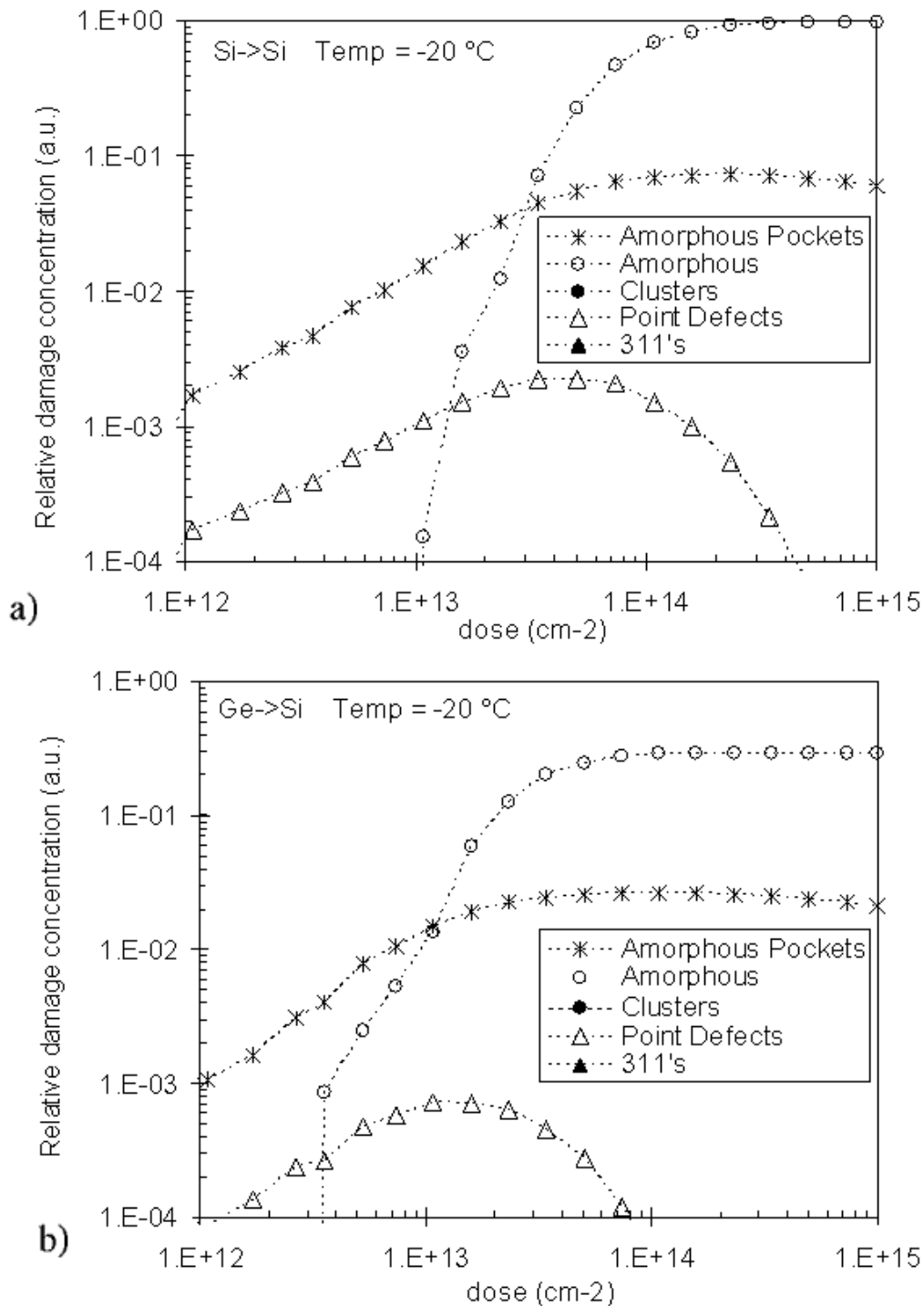


Figura 11.20: Dependencia con la dosis de la fracción de daños debidos a diferentes estructuras de defectos para una implantación de 80 keV de (a) Si y (b) Ge con una velocidad de $5 \times 10^{12} \text{ cm}^{-2} \text{ s}^{-1}$ y a -20 °C .

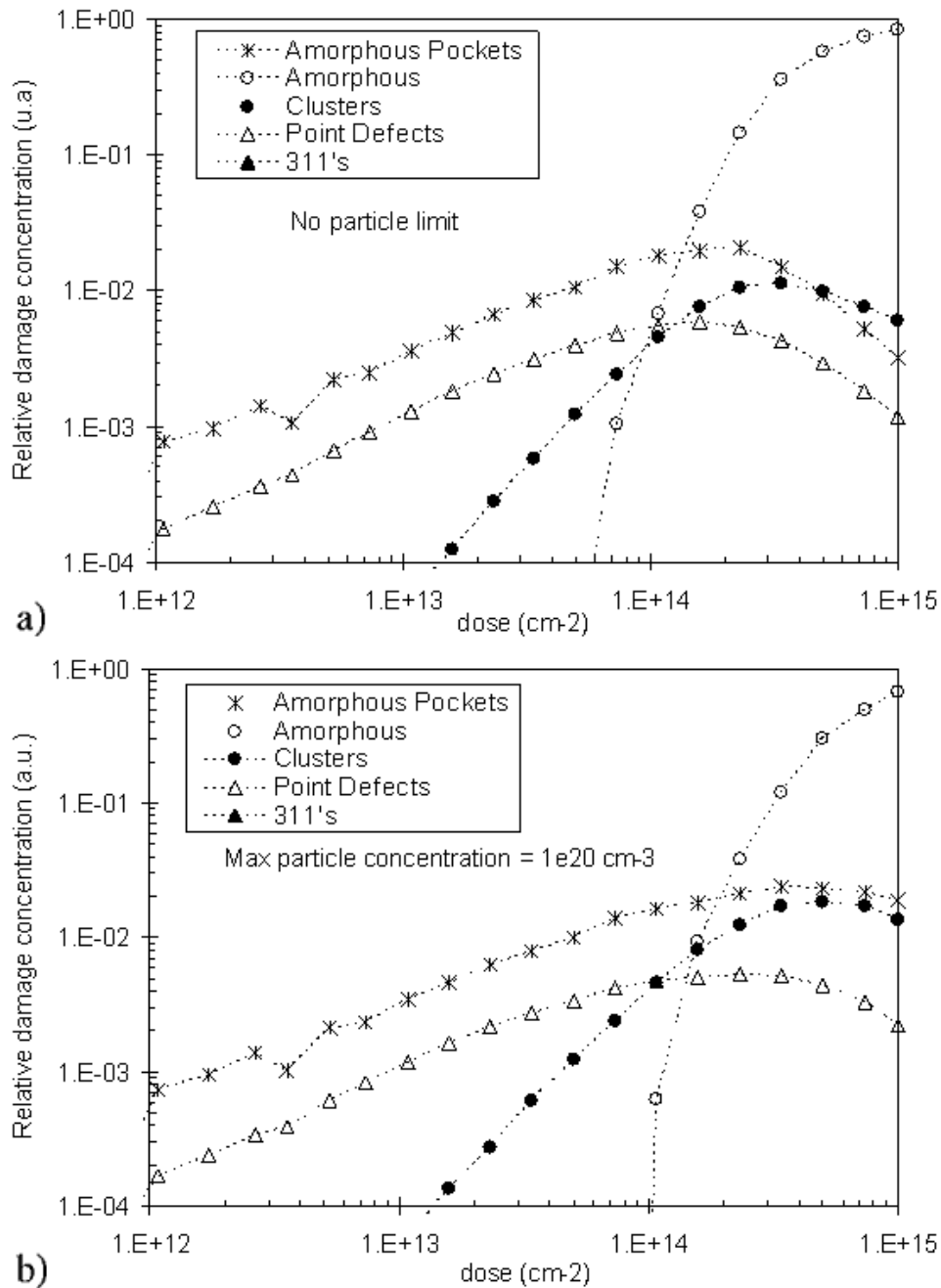


Figura 11.21: (a) y (b) representan la dependencia de la fracción de daño con la dosis para diferentes defectos para una simulación de implantación de Si a 80 keV, una temperatura de 70°C y una velocidad de implantación de $5 \times 10^{12} \text{ cm}^{-2}\text{s}^{-1}$. Para poder establecer una comparación, (a) está realizada sin límite de concentración, pero en (b) la concentración máxima de partículas en una caja está limitada a 10^{20} cm^{-3} con el objeto de ahorrar recursos de cálculo.

▷ 11.6

Un modelo físico y extenso para el arsénico en Silicio

Este artículo se encuentra en la referencia (Pinacho et al., 2004).

Un modelo físico y extenso para el arsénico en Silicio

Resumen

La tecnología actual de procesado de los CMOS exige un conocimiento preciso de la conducta del arsénico, que se encuentra a grandes concentraciones durante el recocido térmico, ya que resulta ser uno de los dopantes más utilizados y extendidos en la fabricación de circuitos de escala de integración ultra-larga (ULSI). Así, pese a la larga sucesión de trabajos de investigación exitosos dedicados a reproducir la cinética del As en Si, el nivel de complejidad que están alcanzando las tecnologías de fabricación en Silicio exigen un modelo físico completo que, estando basado en los parámetros fundamentales de algunas configuraciones básicas del As, pueda a la vez tener en cuenta aspectos diversos como la difusión, desactivación eléctrica y fenómenos de amorfización/recristalización del As durante su implantación y posterior recocido.

De esta manera, se ha utilizado el simulador atomístico de Monte Carlo cinético DADOS para desarrollar un modelo consistente para el As que incluya un conjunto limitado de clusters de AsV con diferentes tamaños y energías. A través de un modelado detallado de los efectos causados por el nivel de Fermi se discutirán las principales propiedades del As en Si, tales como (i) su difusión intrínseca y extrínseca, (ii) su desactivación eléctrica a concentraciones altas (iii) el recocido de perfiles implantados de As y (iv) otras características interesantes como pueden ser la supersaturación de intersticiales inducida por la rápida desactivación eléctrica de concentraciones muy altas de As a temperaturas bajas.

▷ 11.6.1

Introducción

El arsénico es el dopante de tipo n más utilizado en silicio debido, en parte, a su baja difusividad y alta activación eléctrica, propiedades que son cruciales a la hora de maximizar la conductividad de las uniones de los dispositivos, las cuales van siendo cada vez más superficiales. Gracias a los nuevos métodos de procesado en condiciones de no-equilibrio, como el recocido láser y la implantación iónica seguida de un recocido térmico rápido, se pueden fabricar dispositivos muy su-

periciales con muy altas concentraciones de As, a veces incluso por encima de su solubilidad. En este régimen de no-equilibrio, los modelos de simulación que intentan reproducir la conducta del As tienen que incluir varios efectos producidos por las altas concentraciones, que además, sucedan simultáneamente durante el procesado. Entre ellos, además de la difusividad extrínseca (Fahey et al., 1989), que se produce cuando la concentración de As supera la concentración intrínseca de electrones n_i , cabe destacar la rapidísima desactivación eléctrica que sufre el As durante el recocido en aquellas zonas que fueron tratadas con un recocido láser previo (Rousseau et al., 1998) o con una implantación amorfizante de As (Nobili et al., 2001). Así pues se necesita un modelo completo que describa también la amorfización y posterior recristalización del Si, puesto que los átomos de As son lo suficientemente pesados para amorfizar el Si bajo las condiciones de procesamiento típicas.

Ante esta situación los simuladores de Monte Carlo cinético resultan ideales para reproducir estos experimentos, puesto que en ellos es posible incluir todas las reacciones que sean necesarias sin añadir excesivo tiempo de cálculo computacional. Además, los parámetros que usan están directamente relacionados con los modelos físicos calculados para las diferentes configuraciones del As y los distintos defectos extendidos que son intrínsecos al Si. En concreto, el simulador atómico de Monte Carlo cinético DADOS (Jaraiz et al., 2001) está demostrando ser capaz de reproducir y predecir de forma precisa la formación y eventual disolución de defectos que aparecen de forma simultánea en condiciones de no-equilibrio, tales como los $\{311\}$ s, los bucles de dislocación y otros, como los efectos de amorfización y recristalización de capas durante la implantación (Rubio et al., 2004). Así pues, en esta sección se expone cómo se ha incorporado a DADOS un modelo completo para el As en Si que, con un conjunto reducido de parámetros para los clusters As_nV_m , es capaz de explicar las características experimentales mencionadas más arriba, a la vez que mantiene una consistencia con los cálculos de primeros principios y los modelos previamente implementados en DADOS para el resto de los fenómenos que tienen lugar en el silicio durante los pasos de procesado.

► 11.6.2

El modelo

Debido a las propiedades especiales que presenta la difusión del As, nos hemos visto obligados a introducir nuevos procesos en DADOS, los cuales podemos clasificar en tres ramas principales:

1. La difusión extrínseca del As es asistida tanto por mecanismos intersticiales como vacantes.
2. La desactivación eléctrica del As es debida a la formación de clusters de As_nV_m .

3. Dicha desactivación eléctrica resulta ser muy rápida al principio, cuando hay capas con altísimas concentraciones de As, que han podido ser creadas bien por un recocido/fusión láser superficial, o bien por recristalización de las capas amorfas tras la implantación de As.

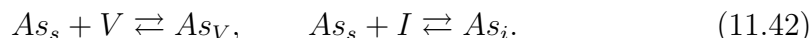
Estos tres puntos serán explicados a continuación.

Difusión extrínseca

Pese a que hay ciertas inconsistencias en los resultados experimentales (Fahey et al., 1989), parece ser que el As difunde tanto por mecanismos de anillo de vacantes, como por *kick-out* intersticial (Matsumoto et al., 1983). De esta manera la difusión del As en Si se puede escribir como:

$$D_{As}^* = D_{As_V}^* + D_{As_i}^*, \quad (11.41)$$

donde D_{As}^* es la difusividad del arsénico en condiciones de equilibrio y $D_{As_V}^*$, $D_{As_i}^*$ son las fracciones debidas a la difusividad por vacantes e intersticiales, respectivamente. Desde un punto de vista atomístico, estas dos contribuciones son consecuencia de la existencia de dos configuraciones móviles para el As, a las que llamamos As_V y As_i , a la vez que consideramos al As sustitucional As_s inmóvil. As_V es así la asociación de un átomo de As con una vacante V , mientras que As_i representa tanto a un As en posición intersticial como al par formado por un As más un intersticial del silicio (I). Las reacciones que controlan la difusión del As serán, por tanto:



Las energías de migración y enlace del As_V y el As_i determinan la difusión del As. Si se altera la concentración de As_V y As_i , por ejemplo, debido a una supersaturación de I que podría acontecer tras la implantación de arsénico, el As es sometido a un proceso de alta difusión temporal (TED). También se puede variar la difusividad efectiva del As cuando hay altas concentraciones de As eléctricamente activo, bastante por encima del nivel de electrones intrínseco n_i para la temperatura de recocido. Para este caso la difusión de As tiene lugar bajo condiciones extrínsecas y, puesto que las configuraciones móviles del As presentan estados de energía en el gap, y pueden atrapar o liberar electrones, dan lugar a la aparición de diferentes estados de carga tales como As_V^0 , As_V^- , As_V^+ , As_i^0 y As_i^+ , cuyas concentraciones van a depender de la posición del nivel de Fermi en el gap, de acuerdo a la concentración de dopantes en cada punto del material. Así pues, en estas condiciones resulta que la difusividad del As (D_{As}) dependerá de la concentración de As eléctricamente activo, como muchos artículos ya han comentado (Fair, 1981b). Las energías de migración y enlace para estas especies involucradas en la difusión del As, junto con sus valores en el gap, se encuentran en la

| Defecto | D_0 (cm ² s ⁻¹) | E_m (eV) | E_b (eV) $T = 0$ K | $e_i - e_V$ (eV) $T = 0$ K |
|----------|--|------------|----------------------|----------------------------|
| AsV^0 | 1×10^{-3} | 1.4 | 1.3 | |
| AsV^+ | 1×10^{-3} | 1.4 | 1 | 0.3 |
| AsV^- | 2×10^{-3} | 1.7 | 1.4 | |
| As_i^+ | 2×10^{-3} | 1.7 | 0.5 | 0.1 |

Tabla 11.2: Parámetros de difusión para las especies móviles de As incluidas en DADOS, donde el super-índice denota el estado de carga del defecto. Las difusividades se relacionan con la expresión general $D_X = D_0 \times \exp(-E_m/k_B T)$ cm²s⁻¹. Se asume que los niveles profundos en el gap varían proporcionalmente al estrechamiento del gap con la temperatura, por lo que también variarán las energías de enlace de las especies con la temperatura (Martin-Bragado et al., 2004c).

tabla 11.2. Los valores de las energías de migración y enlace para el As_V^0 están tomadas de cálculos de primeros principios (Ramamoorthy and Pantelides, 1996). Respecto al As_i , se encuentra en la literatura bastante controversia en la estimación de sus energías (Centoni et al., ; Nichols et al., 1989), razón por la que hemos tomado los valores de energía de As_i de tal manera que ajusten la difusión total intrínseca del As en la ecuación:

$$D_{As}^* = D_{AsV}^* + D_{As_i}^* = \exp(-4,69/k_B T) \text{cm}^2 \text{s}^{-1} \quad (11.43)$$

que utiliza el TSupremIV (Avant!, 2001).

Con los parámetros de la tabla 11.2, se consigue un ajuste bueno entre los modelos de continuo basados en ecuaciones diferenciales y las simulaciones atómicas realizadas con DADOS para diferentes valores de la proporción n/n_i (Martin-Bragado et al., 2004c).

La figura 11.22 muestra una comparación entre la difusividad dependiente de la concentración predicha por DADOS y los resultados experimentales de Fair (Fair, 1981b). La difusividad extrínseca del As (Fahey et al., 1989) es descrita con la siguiente ecuación:

$$D_{As} = D_{AsV} + D_{As_i} = h \left(D_{AsV}^i + D_{AsV^0}^i \frac{n}{n_i} + D_{AsV^-}^i \left(\frac{n}{n_i} \right)^2 + D_{As_i^+}^i + D_{As_i^0}^i \frac{n}{n_i} \right), \quad (11.44)$$

donde D_X^i es la difusividad de la especie X en condiciones intrínsecas, n la concentración de electrones, n_i la concentración intrínseca de electrones y h un factor debido al campo eléctrico, ver (Fahey et al., 1989). Con los parámetros listados en la tabla 11.2 la difusión se debe principalmente al término As_V^0 , que supone a su vez que el segundo término de la ecuación 11.44 va a predominar y predice, por tanto, una dependencia lineal de la difusividad del As con la concentración de electrones. Utilizando la estadística de Maxwell–Boltzmann, podemos escribir

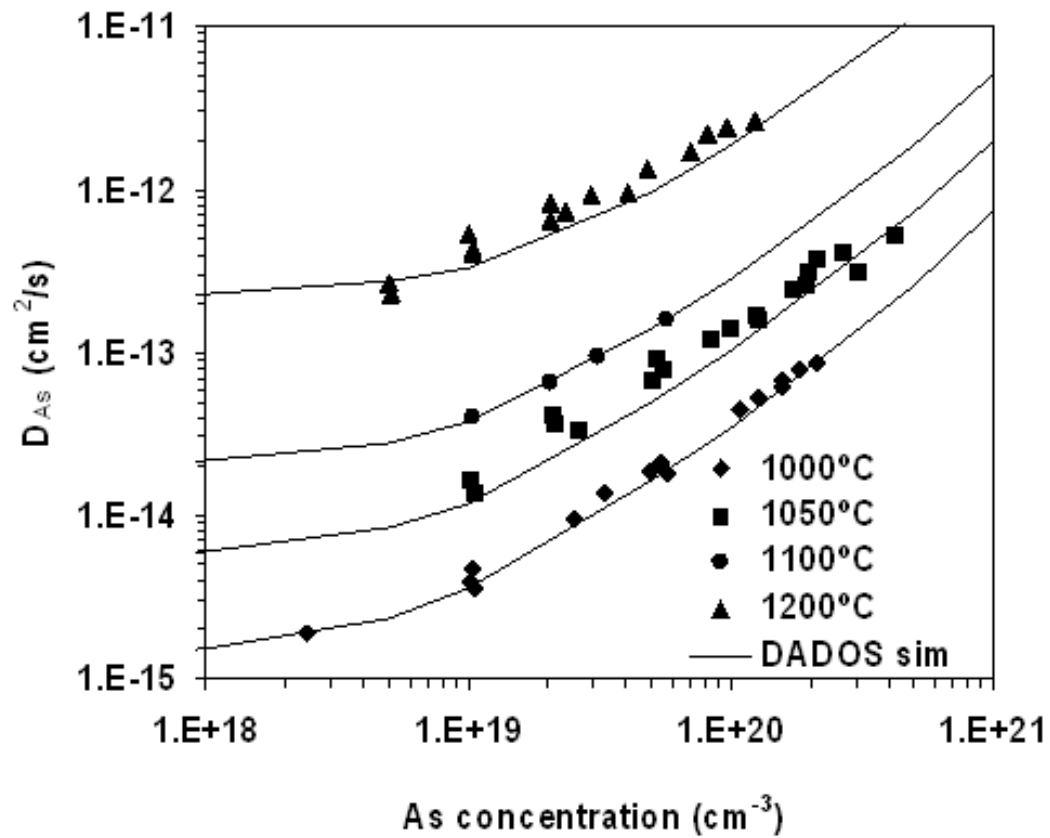


Figura 11.22: Difusividad del As frente a la concentración total de As para temperaturas distintas. Los símbolos son resultados experimentales tomados de la referencia (Fair, 1981b), y las líneas son los valores predichos por DADOS.

n como

$$n = \frac{[As_s]}{2} + \left(\frac{[As_i^2]}{4} + n_i^2 \right)^{1/2}, \quad (11.45)$$

y por tanto, para concentraciones altas de As ($[As_s] \gg n_i$), el modelo de DADOS predice una dependencia de la difusividad del As con la concentración casi lineal. Ese rango es el mostrado en la figura 11.22, en buen acuerdo con los experimentos.

Clusters de As_nV_m

Para concentraciones de As altas, por encima de $\approx 3 \times 10^{20} \text{ cm}^{-3}$, el fenómeno de la desactivación eléctrica del As que acontece de forma simultánea a la difusión del As, se puede observar fácilmente (Nobili et al., 2001). Este proceso tiene una amplitud bastante considerable, y de hecho, el límite máximo de concentración para el As disuelto en equilibrio con sus precipitados, excede en un orden de magnitud el valor límite de la concentración de portadores (Nobili et al., 1994; Solmi and Nobili, 1998).

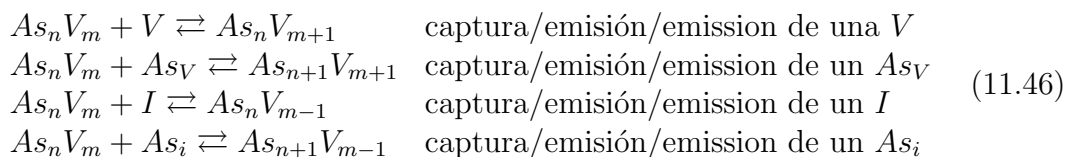
Así pues la clusterización de los átomos de As está aceptada como el fenómeno responsable de los As eléctricamente inactivos que no han precipitado, habiendo sido propuestos distintos modelos para ella. Todos ellos asumen la formación de agregados eléctricamente inactivos a temperatura ambiente, que consisten en un número fijo de átomos dopantes en configuraciones específicas de red, que pueden incluir vacantes o, incluso, intersticiales (Rousseau et al., 1998; Ramamoorthy and Pantelides, 1996; Mueller et al., 2003; Pandey et al., 1988; Berding et al., 1998; Chadi et al., 1997). Los modelos propuestos van desde la agregación de dos átomos de dopantes, sin vacantes añadidas, hasta la formación de clusters de 2 a 4 dopantes que incluyen vacantes o intersticiales. No obstante, algunos experimentos recientes (medidas Hall y de resistividad, así como aniquilación de positrones), están apuntando a los mecanismos de desactivación inducidos por el clustering de As_nV (Ranki et al., 2002; Lawther et al., 1995). Estos clusters deberán ser más energéticamente favorables, en comparación con el As sustitucional eléctricamente activo, debido a sus bajas energías de formación y, por tanto, después de un tratamiento térmico la mayoría de los átomos de As terminan encontrándose en estas configuraciones estables.

En nuestro modelo hemos incluido la formación de clusters As_nV_m eléctricamente neutros, que son conglomerados formados por átomos de As y vacantes. No hemos incluido clusters de As combinados con Is , As_nI_m porque, de acuerdo con la mayor parte de los cálculos, no parece que sean lo suficientemente estables (Ramamoorthy and Pantelides, 1996; Mueller et al., 2003; Pandey et al., 1988). La tabla 11.3 expone los valores de energía para estos clusters. En ella se puede ver que los clusters más estables son As_2V , As_3V y As_4V , lo que está de acuerdo con la mayoría de los cálculos de primeros principios (Ramamoorthy and Pantelides, 1996; Mueller et al., 2003; Pandey et al., 1988; Berding et al., 1998) y con recientes observaciones experimentales (Ranki et al., 2002; Lawther et al., 1995).

| | | | |
|------|---------|---------|-----------|
| As | As_2 | As_3 | As_4 |
| 0 | 0 | 0 | 0 |
| | As_2V | As_3V | As_4V |
| | -5 | -6 | -6 |
| | | | As_4V_2 |
| | | | -10.3 |

Tabla 11.3: Energías de enlace totales para los clusters As_nV_m . Hay que señalar que, debido a su baja energía, es posible formar clusters As_2V , As_3V y As_4V cuando los átomos de As estén lo suficientemente cerca ($As + As + As \rightarrow As_3V + I$), es decir, cuando su concentración sea muy alta.

La cinética de desactivación del As está controlada por el crecimiento y disminución de los clusters de As_nV_m . Puesto que para el caso del As tenemos dos especies móviles (tanto el As_V como el As_i), dichos mecanismos de crecimiento y disminución estarán descritos por las siguientes ecuaciones:



Por otra parte, la reacción inversa de las dos últimas expresiones supone la formación de un par IV desde el volumen, donde la vacante queda inmovilizada en el cluster mientras que se emite el intersticial por separado (ecuación 11.46c) o junto con un As (ecuación 11.46d). Consecuentemente, en ambos casos, se puede incrementar de manera anómala la difusión del As.

Fusión láser y recristalización

Hasta ahora se ha explicado como difunde el As y cómo los clusters de As_nV_m pueden crecer y disolverse. No obstante, aún no hemos explicado cómo surgen los clusters de As al principio. Para concentraciones muy altas de As ($2,3 \times 10^{20} - 1,9 \times 10^{21}$) creadas a través de la recristalización de capas dopadas con As por fusión láser, se observa experimentalmente que la desactivación electrónica inicial es muy rápida (unos 15 segundos) incluso a temperaturas bajas (hasta 500°C) (Rousseau et al., 1998; Dunham et al., 2003) a la vez que se insertan intersticiales hacia el sustrato. A estas temperaturas tan bajas, la concentración de I s y V generada térmicamente por la superficie es bastante baja y por tanto, la formación de clusters asistida por estas especies debería ser mucho más baja de lo que se ve experimentalmente. Así pues, para explicar esta desactivación tan rápida algunos artículos (Rousseau et al., 1998) proponen la formación de As_nV asociada con la emisión de un I , de acuerdo con la ecuación 11.46c.

Por otra parte, un efecto similar se puede observar en las implantaciones y recristalizaciones rápidas de capas de As con concentraciones muy altas. En este caso, aunque se supone que no se dejan defectos en las partes recristalizadas, si que queda una concentración de As eléctricamente activo, cuyo límite superior se encuentra cerca de $1 \times 10^{21} \text{ cm}^{-3}$ (Nobili et al., 2001).

En el simulador DADOS, ambos fenómenos están adecuadamente modelizados por la reacción inversa de la ecuación 11.46c. De hecho, con las energías que hay en la tabla 11.3, la formación de As_2V , As_3V , y As_4V es bastante rápida incluso a temperaturas bajas, gracias a reacciones como $As + As + As \rightarrow As_3V + I$. Para que sea posible la realización de tales procesos es necesario que la concentración de As sea lo suficientemente alta como para que existe una probabilidad elevada de que dos, tres o cuatro átomos de As estén cerca unos de otros, y puedan por tanto, transformarse en un cluster de As_nV mediante la emisión de un I . Así pues este proceso es muy sensible a la concentración de As, como se ha observado experimentalmente (Rousseau et al., 1998).

▷ 11.6.3

Discusión de los resultados

Hemos simulado el experimento realizado por (Rousseau et al., 1998) para poder comprobar nuestro modelo. La estructura experimental consiste en una capa enterrada de boro, que funciona como detector de intersticiales, y una capa superficial de As recocido por láser completamente activo con una alta concentración de As, concretamente $4,5 \times 10^{20} \text{ cm}^{-3}$, como se muestra en la figura 11.23. Posteriormente se hace un recocido a 750°C que desactiva el As a la vez que el boro sufre un proceso de difusión transitoria incrementada, por la generación de intersticiales durante la desactivación del As.

La figura 11.23, muestra una comparación entre los experimentos de Rousseau y las simulaciones hechas con DADOS. Como se puede ver, se logra un acuerdo muy razonable. Las simulaciones con DADOS también permiten ver que en los primeros segundos de recocido el cluster que domina la composición es el As_2V , quizás debido a que la probabilidad de que haya sólo dos átomos de As cercanos es mayor a la de que haya tres. A su vez, los I s generados durante la formación de As_2V , As_3V o As_4V difunden y pueden combinarse con átomos aislados de As, que forman así As_i s. Estos As_i s incrementan la difusión del arsénico y permiten, por tanto, que los clusters de As crezcan también de acuerdo a la reacción 11.46d y a la reacción opuesta a la 11.46c. Al final del recocido la mayor parte del As inactivo permanece como As_3V , tal como expone (Lawther et al., 1995), quien encuentra por medio de experimentos de aniquilación de positrones, que el número medio de As que rodean a una vacante es de 2.5 y 3.2 para tiempos de 15 s y 2 h a 750°C , respectivamente. Una fracción de As se encuentra también en As_4V_2 , que se forma a partir del As_3V por la ecuación 11.46b o bien desde el As_4V mediante la ecuación 11.46c. También queda claro mediante la figura 11.23 que la difusividad del

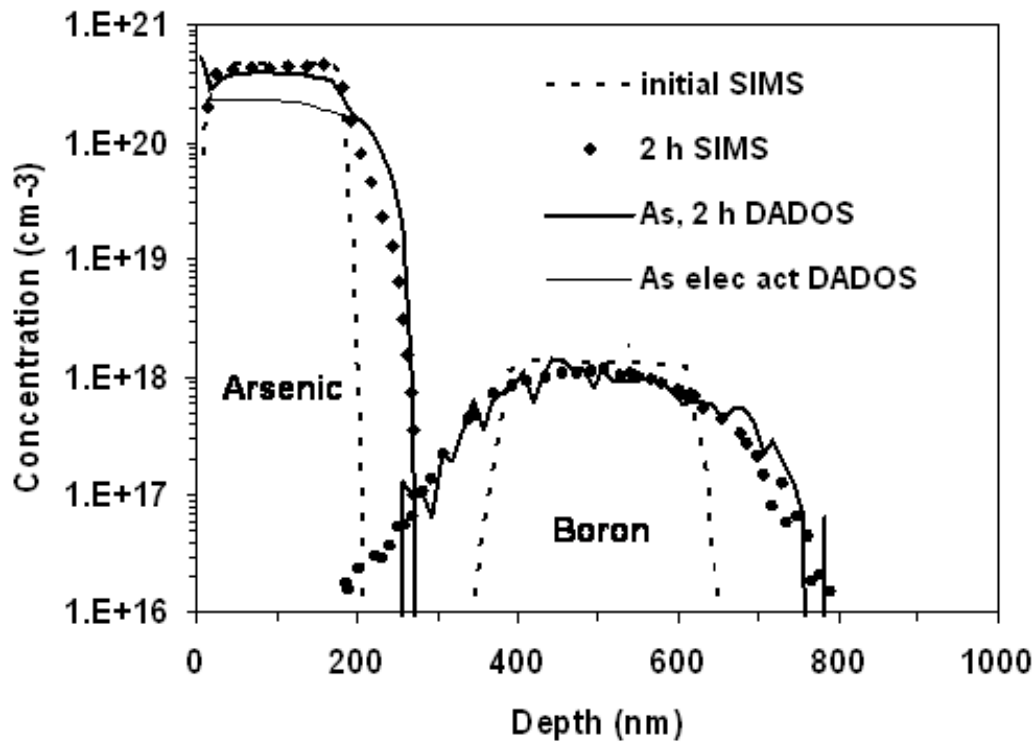


Figura 11.23: Resultados SIMS para las estructuras usadas en el experimento de (Rousseau et al., 1998), que consisten en una capa enterrada de B y una capa superficial de As implantada y posteriormente recocida con fusión láser. Después de 2 h. a 700°C , los datos SIMS muestran la difusión incrementada para la capa enterrada de B. La simulación hecha con DADOS (líneas) ajusta bien los perfiles de As y B, y también hay que destacar que la simulación de DADOS predice una concentración de As eléctricamente activa cercana a $2 \times 10^{20} \text{ cm}^{-3}$, en un buen acuerdo con el experimento.

B, y por tanto la supersaturación de I_s generada, está adecuadamente predicha en nuestro modelo. Por último, la simulación también predice la formación de bucles de dislocación desde el principio del tratamiento térmico, como afirman los experimentos (Dokumaci et al., 1995).

Se han simulado también algunos experimentos con implantaciones de As amorfizantes y un posterior recocido, con la intención de chequear el comportamiento del modelo de amorfización y la conducta de la difusión y clusterización del As durante y tras la recristalización (Rubio et al., 2004).

Por último, en la figura 11.24 a) y b) se muestran los experimentos de (Kim et al., 2000) para implantaciones amorfizantes de As y posterior recocido a 720°C y 820°C respectivamente, junto con las simulaciones realizadas con DADOS. Se observa un buen acuerdo entre los experimentos y las simulaciones. Concretamente, las simulaciones de DADOS muestran que la capa amorfa creada durante la implantación, que se extiende hasta 50 nm de profundidad, recristaliza muy rápidamente en los primeros segundos del tratamiento térmico y, después de esto, en la región recristalizada queda una pequeña porción de As en forma de clusters de As_nV energéticamente favorables. Esta fracción aumenta ligeramente durante el recocido, por el crecimiento de los clusters. A ambas temperaturas se observa que el As sufre TED debido a que la difusión por medio de As_i s es incrementada por la supersaturación de I_s que se generó durante la implantación. Por último los $\{311\}$ s presentes van evolucionando hacia bucles de dislocación que aparecen más allá de la interfaz amorfo/cristal, atrapando el exceso de I_s y rebajando de esta manera la difusión anómala del As para tiempos largos.

▷ 11.6.4

Conclusiones

Se ha desarrollado un modelo completo para explicar el comportamiento del As en Si. Hemos puesto énfasis en incluir los efectos que se producen a altas concentraciones, tales como la difusión extrínseca y la desactivación eléctrica. Con este modelo, los experimentos de desactivación rápida de concentraciones muy altas de As logradas con capas de fusión láser, junto con los experimentos de implantación amorfizante de As y posterior recristalización, se pueden reproducir de forma precisa y adecuada.

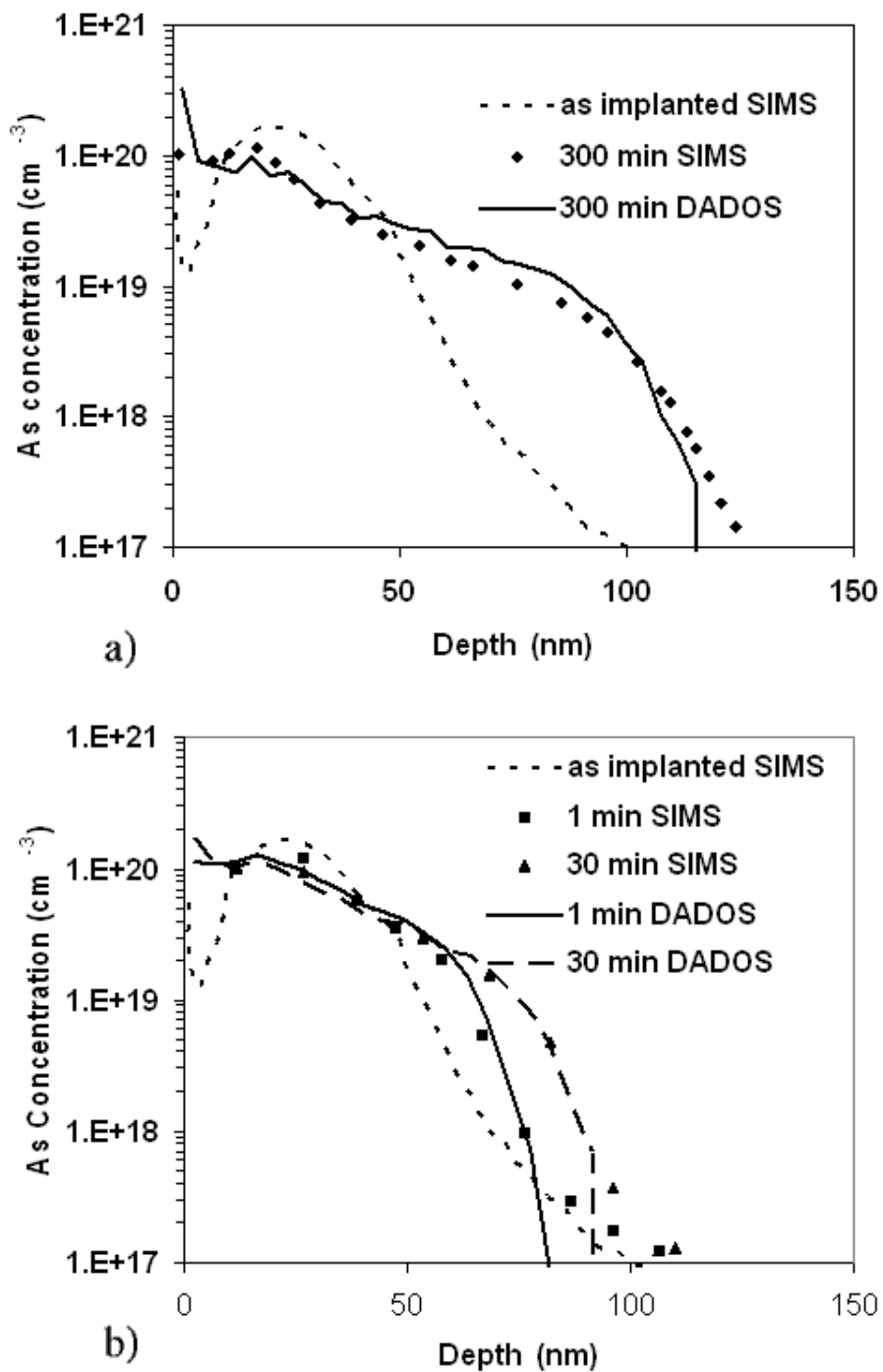
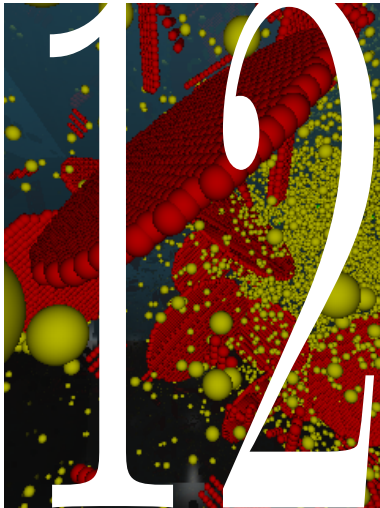


Figura 11.24: Perfiles de As tomados con SIMS por (Kim et al., 2000) para muestras de As implantado a $5 \times 10^{14} \text{ cm}^{-2}$ y recocido posteriormente a dos temperaturas diferentes: a) 720°C durante 300 min y b) 820°C durante 1 y 30 min. Se incluyen también las simulaciones hechas con DADOS para estas condiciones.

Parte IV
Conclusiones



Conclusiones

| | |
|---|-----|
| 12.1. Construcción de un entorno de simulación | 317 |
| 12.2. Desarrollo de diversos simuladores atomísticos . . | 317 |
| 12.3. Desarrollo y puesta a punto del simulador KMC DADOS . . . | 317 |
| 12.4. Estudios con resultados originales | 319 |
| 12.5. Conclusión | 320 |

A lo largo de los capítulos de esta tesis se han ido presentando diferentes técnicas de simulación atomística, incluyendo la dinámica molecular, Monte Carlo de red y Monte Carlo cinético, junto con un método de optimización de clusters que usa algoritmos genéticos. De estos métodos se ha hecho un hincapié especial en el modelo de Monte Carlo cinético, aplicado a la difusión de impurezas y formación de defectos extendidos en Silicio durante los procesos de fabricación de dispositivos. Así pues, después de todo este estudio, es tiempo de ir concluyendo.

Para concluir la presente tesis doctoral debemos preguntarnos qué se ha logrado con el presente estudio y qué aporta este trabajo al campo de la simulación atomística de procesos para microelectrónica.

▷ 12.1

Construcción de un entorno de simulación

Se ha construido el entorno completo para la simulación de procesos atomísticos UVAS, con el trabajo de formación que esto supone, empezando por el dominio de las técnicas de programación en C++, Visual C++ y Unix, y terminando con la resolución de los problemas técnicos involucrados en el desarrollo de un entorno de visualización en dos y tres dimensiones, es decir, la construcción de un interfaz intuitivo y genérico para manipular las simulaciones atomísticas: las instrucciones a simular, su ejecución, y el análisis posterior de los resultados generados.

▷ 12.2

Desarrollo de diversos simuladores atomísticos

Se han desarrollado íntegramente, e incluido en el entorno UVAS, simuladores de Dinámica Molecular, Monte Carlo con red y un optimizador de estructuras por algoritmos genéricos. Estas técnicas han supuesto un esfuerzo de formación, programación y ajuste, y han logrado aportar una visión general del tema de la simulación atomística. Durante el desarrollo se cuidó el aspecto físico del problema tratado, pero también aspectos más técnicos relacionados con la eficiencia de los modelos desarrollados, para lograr que fueran computacionalmente aceptables.

▷ 12.3

Desarrollo y puesta a punto del simulador KMC

DADOS

En este trabajo se ha mostrado la evolución del simulador DADOS hasta convertirlo en una herramienta usada profesionalmente para la simula-

ción de los procesos de difusión y activación/desactivación de dopantes durante los pasos de recocido. Se han añadido modelos y mejorado otros que ya existían, a la par que se ha hecho un gran esfuerzo en mejorar su funcionamiento global y eficacia. **Todo esto ha permitido su disponibilidad comercial a través del simulador TaurusTM Process Atomistic**, que es distribuido por la empresa Synopsys^(R) como herramienta de simulación de procesos en dispositivos de tamaño inferior a la micra que incorpora casi todos los mecanismos conocidos hasta la fecha, prestando especial atención al reto que supone el escalamiento de los dispositivos y, en concreto, a los problemas de difusión y activación de dopantes, evolución del dañado de implantación y amorfización y al uso que se puede dar al simulador para extraer parámetros para modelos de continuo e investigar las variaciones estadísticas de los dispositivos (Moroz et al., 2003).

Más en concreto, se ha trabajado activamente en los siguientes aspectos:

- El modelo de carga estaba presente originariamente en una única dimensión y era poco eficiente. El modelo actual está en tres dimensiones y ha sido completamente reescrito para solucionar muchos problemas tanto técnicos como físicos. Estos problemas estaban relacionados con la difusividad irrealmente alta que se obtenía al discretizar la carga atomística. Ha habido que desarrollar un método de suavizado y promediado de la carga para evitar dicho problema. Así mismo ha sido necesario un estudio intenso para lograr un modelo rápido y eficiente. Esto ha supuesto evolucionar desde un modelo primitivo que tomaba en cuenta todas las contribuciones a las cargas, por pequeñas que fueran, y que resultó ser muy poco eficiente, a otro un poco más aproximado (suponiendo que los dopantes sustitucionales están siempre ionizados, entre otras aproximaciones) pero igualmente preciso.
- Inclusión de un modelo de acumulación de dañado, amorfización y recristalización. Este modelo supone un reto técnico, puesto que el gigantesco número de partículas involucradas en el proceso de amorfización hace que una implementación atomística directa sea poco eficiente e impracticable para tamaños medianos. Ha sido necesario, por tanto, adaptar el modelo a tamaños grandes, despreciando las coordenadas de muchas partículas y guardando únicamente su número. La posterior recristalización presentó también abundantes problemas técnicos y físicos. Entre los técnicos cabe destacar el problema de recristalizar correctamente todas las regiones en el orden correcto, sin que queden islas cristalinas en medio de regiones amorfas (o al revés), y entre los físicos modelizar el depósito de clusters de impurezas durante la recristalización cuando la concentración de éstas se encuentre por encima de la máxima permitida.
- Inclusión del modelo de Oh-Ward para la interacción de los dopantes con la superficie. Ha habido que adaptar este modelo desde una formulación del

continuo hasta una atomística, y después encontrar métodos de modelizado atomístico que permitan implementarlo de modo eficiente.

▷ 12.4

Estudios con resultados originales

Además de generar, implementar y ajustar herramientas de simulación, se han utilizado éstas para conseguir resultados de investigación útiles, valiosos y originales, entre los que vamos a destacar los siguientes:

- Se han estudiado las implicaciones que supondría la inclusión de la especie móvil I_2 en los simuladores de procesos, llegando a la conclusión de que no parece imprescindible, de momento, la inclusión de dicha especie, ya que sus efectos pueden tenerse en consideración a través del I .
- Se ha trabajado en la importancia de la correlación de los pares IV en la simulaciones de recocido tras una implantación iónica, tema que aporta información valiosa y justifica prácticas comunes de simulación en simuladores de continuo. Se ha llegado a la conclusión de que para simulaciones con implantaciones medias dicha correlación no es importante, y que la diferencia $I - V$ es suficiente para tener en cuenta efectos posteriores, concretamente la TED. Así mismo se ha observado como el número total de partículas si que resulta necesario para la formación de clusters de impurezas que inmovilicen y desactiven dichas impurezas.
- Se ha ofrecido un estudio completo, novedoso y original que implementa la dependencia de la difusividad de dopantes y de energías de formación con el nivel de Fermi (modelo de carga). Este estudio no sólo desarrolla a fondo los aspectos físicos del modelo, sino también las implicaciones que surgen al intentar crear un modelo preciso, pero también computacionalmente efectivo. Se han ofrecido resultados que muestran como el modelo simula exitosamente diferentes difusividades y distancias de salto largo según la concentración electrónica. También se ha mostrado como el modelo permite simular el apilamiento de boro en la zona P de una unión PN, y su ausencia parcial en el inicio de la zona N . Este fenómeno no se podría explicar sin un modelo de carga.
- Se ha mostrado un modelo de acumulación de daño, amorfización y recristalización integrado con el resto de modelos atomísticos. Con las condiciones de procesado actuales es práctica común el uso de implantaciones que amorficen el silicio, y la inclusión de modelos de amorfización y posterior recristalización precisos se vuelve crítica para la correcta simulación

de este tipo de procesos. Así mismo el trabajo no sólo muestra los aspectos físicos del modelo, sino detalles técnicos muy valiosos para la generación de un modelo funcional y eficiente que pueda ser incorporado en simuladores de dispositivos.

- Por último se han mostrado los últimos resultados relacionados con la activación y desactivación del arsénico, uno de los dopantes que más se está utilizando en los procesos microelectrónicos actuales. Se ha prestado especial atención a la difusión y formación de clusters de este dopantes en condiciones extremas de muy alta concentración y bajo implantaciones amorfizantes.

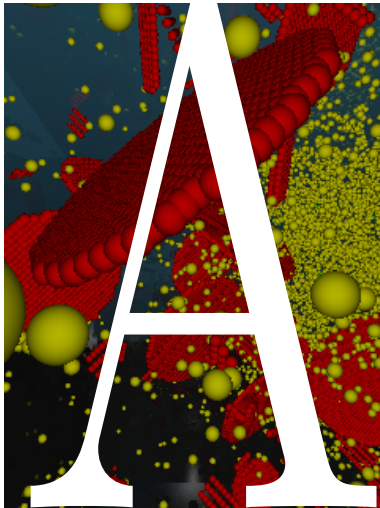
▷ 12.5

Conclusión

Otros simuladores atomísticos de los que tenemos noticia (Ezaki et al., 2002; Yu et al., 2003; Theiss et al., 1999), se han basado en las primeras versiones de DADOS (distribuidas a algunas universidades y centros de investigación), y no incluyen aún muchos de los mecanismos que ya hemos incorporado.

En resumen, adelantando las expectativas y previsiones del ITRS Roadmap del año 2003 (Several, 2003), hemos desarrollado lo que creemos es el primer simulador atomístico utilizable en los simuladores de proceso usados en la industria.

Parte V
Apéndices



Deducción de expresiones utilizadas en DADOS

- A.1. Ecuaciones de difusividad 325
- A.2. Impurezas 326
- A.3. La superficie 331
- A.4. Equilibrio en el volumen interno 334
- A.5. Cálculos del modelo de carga 335
- A.6. Implementación del modelo de carga 336
- A.7. Un ejemplo: Difusión del boro y estados de carga 339
- A.8. Energías y prefactores de activación efectivos para gráficas de Arrhenius 342
- A.9. Notación 343

En este capítulo se muestran algunas relaciones matemáticas y demostraciones que han sido utilizadas en capítulos anteriores y, en general, durante la implementación de modelos en DADOS.

▷ A.1

Ecuaciones de difusividad

Vamos a introducir las ecuaciones de difusividad, según la aproximación del continuo, para así poder definir los conceptos de flujo y difusividad.

▷ A.1.1

Definiciones de flujo y difusividad en una dimensión

Vamos a definir el flujo (Φ) como el número de partículas que atraviesan una determinada superficie por unidad de superficie y tiempo. Sea así $n(x)$ el número de partículas por unidad de volumen (la concentración de partículas). Suponiendo que las partículas migran al azar, la mitad de ellas saltarán a la izquierda y la mitad a la derecha, por lo que tendremos que

$$\Phi(x) = \left(\frac{1}{2}n(x)\Delta x - \frac{1}{2}n(x + \Delta x)\Delta x \right) \tau^{-1},$$

donde τ es el tiempo consumido por una partícula al saltar, que se puede expresar en función de su frecuencia media de salto como $\tau = \frac{1}{\nu}$. Si dividimos y multiplicamos por la magnitud Δx :

$$\Phi(x) = \frac{1}{2} \frac{(\Delta x)^2}{\tau} \frac{n(x) - n(x + \Delta x)}{\Delta x}$$

e identificamos esta última fracción con una derivada parcial, y la distancia incremental Δx con λ , la distancia de salto, podremos reescribir la anterior ecuación como:

$$\vec{\Phi}(\vec{r}) = -D\vec{\nabla}n \quad (\text{A.1})$$

donde la difusividad queda definida como:

$$D = \frac{1}{6} \frac{\lambda^2}{\tau} \quad (\text{A.2})$$

▷ A.1.2

Ley de Fick en una dimensión

Se va a calcular a continuación la variación con el tiempo del número de partículas. Para ello la variación en el número de partículas será $\Delta n \Delta x$, es decir, la variación de la concentración en una determinada profundidad multiplicada por la pequeña distancia considerada. Esta variación es creada por las nuevas partículas que cruzan una separación en x imaginaria y por las que se salen de ella en $x + \Delta x$, es decir:

$$\Delta n(x)\Delta x = \Phi(x)\Delta t - \Phi(x + \Delta x)\Delta t$$

Modelo en la
sec. 8.1.1,
pág. 104.

de tal manera que

$$\frac{\Delta n(x)}{\Delta t} = -\frac{\Delta \Phi(x)}{\Delta x}$$

y, por último, si identificamos términos con una derivada parcial:

$$\frac{\partial n}{\partial t} = \frac{\partial}{\partial x} \left(D \frac{\partial n}{\partial x} \right).$$

Suponiendo que D no presenta ninguna dependencia con x ¹:

$$\frac{\partial n}{\partial t} = D \frac{\partial^2 n}{\partial x^2}. \quad (\text{A.3})$$

▷ A.2

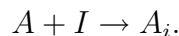
Impurezas

Supongamos que A representa una impureza en el silicio, que podría difundir por intersticiales (A_i) o vacantes (A_V). En esta sección se pretende, en primer lugar, deducir las ecuaciones atómicas que determinarán la concentración de A_i (suponiendo que difunde por intersticiales) en condiciones de equilibrio y casi-equilibrio, y en segundo lugar deducir la difusión de A_i y la difusión efectiva de la especie A a través de A_i . Se acabará la sección con un ejemplo para el caso particular del boro.

▷ A.2.1

Concentración de impurezas en equilibrio

Vamos a suponer defectos puntuales independientes, es decir, la distancia entre ellos va a ser lo suficientemente grande como para que no haya interacciones entre ellos, y los podemos suponer aislados, y llamemos A a la impureza (inmóvil) e I al intersticial propio del silicio. Tendremos entonces la siguiente reacción



En equilibrio y casi-equilibrio los dos términos deberán contar el mismo número de aportaciones y, por tanto,

$$\nu_{\text{cap}}^{A \rightarrow I} = \frac{\text{frecuencia de captura por unidad de volumen}}{\text{frecuencia de salto de } I \text{ por unidad de volumen}} \times \frac{\text{Probabilidad de ser capturado por } A_s}{\text{Probabilidad de ser capturado por } A_s}$$

¹Cosa que no tiene por qué se cierta si existe, por ejemplo, una dependencia con el nivel de Fermi, que a su vez puede cambiar con x .

▷En nuestro modelo atomístico vamos a representar A_i como un átomo de la especie A y un intersticial de Si. Esto quiere decir que estamos denominando de la misma manera a las configuraciones diferentes correspondientes a un I más un A en posición substitucional (A_s), y a A en posición intersticial, dado que despreciamos las configuraciones microscópicas a ese nivel, y tomamos valores efectivos de A_i .

La frecuencia de salto de I por unidad de volumen es:

$$\frac{6D(I)}{\lambda^2}[I], \quad (\text{A.4})$$

y la probabilidad de ser capturado por un A_s es

$$V_{\text{cap}}(1)/(V_{\text{total}}) \times p_{\text{cap}},$$

siendo $V_{\text{cap}}(1)$ el volumen de captura de la especie A_s , V_{total} el volumen total y p_{cap} la probabilidad de esta captura.

▷La probabilidad de captura será $p_{\text{cap}} = 1$ cuando se junten siempre A e I , o bien un valor fijo $p_{\text{cap}} = K$. También puede ser una barrera energética

$$p_{\text{cap}} = \exp(-E/k_B T).$$

En general vamos a suponer siempre que esta probabilidad es $p_{\text{cap}} = 1$.

De las anteriores ecuaciones vemos que

$$\frac{V_{\text{cap}}(A_s)}{V_{\text{total}}} p_{\text{cap}} = [A] v_{\text{cap}}(1),$$

donde $v_{\text{cap}}(1)$ depende de la geometría considerada:

$$\begin{array}{ll} \text{Esférica} & v_{\text{cap}}(1) = \frac{1}{2} \frac{4}{3} \pi \lambda^3 \\ \text{Ortogonal} & v_{\text{cap}}(1) = \underbrace{\frac{1}{2}}_{\text{Average}} (2\lambda)^3, \end{array} \quad (\text{A.5})$$

y por último, usando la definición de $\nu_{\text{cap}}^{A \rightarrow I}$ junto con la ecuación A.4 podemos poner la frecuencia de captura por unidad de volumen como:

$$\nu_{\text{cap}}^{A \rightarrow I} = \frac{6D(I)}{\lambda^2}[I] \times \frac{1}{2}(2\lambda)^3 \times [A_s] = 24\lambda D_I [I][A_s].$$

En condiciones de equilibrio, el número de I s capturados será igual al de A_i rotos, luego

$$\underbrace{[A_i]^* \nu_{\text{break}}(A_i)}_{\text{frecuencia de emisión}} = \underbrace{24\lambda D(I)[I]^*[A_s]^*}_{\text{frecuencia de captura}}$$

y por tanto

$$[A_i]^* = v_{cap}(1) \frac{6D(I)}{\lambda^2 \nu_{break}(A_i)} [I]^* [A]^*. \quad (A.6)$$

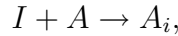
▷ En DADOS todas las frecuencias están definidas en unidades de difusividad, con lo que para convertir unidades habrá que tener en cuenta que

$$\nu_{break}(A_i) = \frac{6D_{break}(A_i)}{\lambda^2}.$$

▷ A.2.2

Cálculo de $[A_i]$ en condiciones de casi-equilibrio

Vamos a volver a escribir la reacción original



en donde hay una especie fija (A) que se mueve a través de A_i .

En casi-equilibrio el número de A_i s rotos será bastante similar al número de A_i s que se forman, y de esta manera podemos reescribir la ecuación A.6 como

$$\frac{[A_i]}{[A]} = v_{cap}(1) \frac{6D(I)}{\lambda^2 \nu_{break}(A_i)} [I], \quad (A.7)$$

con lo que, si dividimos A.6 entre A.7 tendremos

$$\frac{[A_i]}{[A][A_i]^*} = \frac{[I]}{[A]^*[I]^*},$$

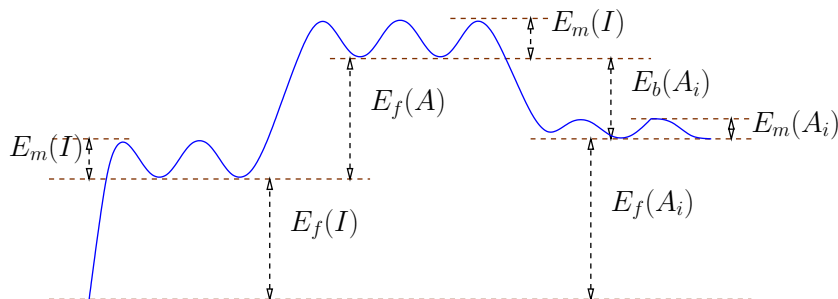
y ordenando esta última ecuación

$$\frac{[A_i]}{[A]} = \frac{[I]}{[I]^*} \frac{[A_i]^*}{[A]^*}.$$

Ya que hemos supuesto que A está fijo, está claro que $[A] = [A]^*$, es decir, la concentración substitutional que tengamos en la simulación, y por tanto al usar la ecuación A.22 se puede escribir:

$$[A_i] = S(I)[A_i]^*. \quad (A.8)$$

Siendo $S(I)$ la supersaturación (cociente entre la concentración de intersticiales y la concentración de éstos en equilibrio).

Figura A.1: Energía de formación de A_i .

▷ A.2.3

Prefactor y energía de activación de $[A_i]^*$

Modelo en la
sec. 8.6.1,
pág. 139.

Vamos a definir

$$[A_i]^* = \text{prefactor} \times \exp(-E_f(A_i)/k_B T)$$

e intentar extraer el valor del “prefactor” y de $E_f(A_i)$ usando A.6, donde $\nu_{break}(A_i)$

Modelo en la
sec. 8.6.1,
pág. 140.

en nuestro caso es

$$\nu_{break}(A_i) = \frac{6}{\lambda^2} D_{0,b}(A_i) \exp[-(E_b(A_i) + E_m(I))/k_B T],$$

y, por tanto

$$\begin{aligned} E_f(A_i) &= E_m(I) + E_f(I) + E_f(A) - E_b(A_i) - E_m(I), \\ &= E_f(A) + E_f(I) - E_b(A_i) \end{aligned}, \quad (\text{A.9})$$

valores que pueden observarse de manera más intuitiva en la figura A.1.

▷ A.2.4

Difusión de A

Hasta ahora hemos visto como la especie A difunde usando el “par” móvil A_i , y el número de saltos de A_i por unidad de tiempo y volumen es

$$D(A_i) = \nu_m(A_i)[A_i],$$

que, relacionándolo con magnitudes del equilibrio, resulta ser (ecuación A.8)

$$D(A_i) = \nu_m(A_i)S(I)[A_i]^*.$$

No obstante, en esta sección pretendemos calcular la difusividad de A , no la de A_i . De esta forma, la aparente difusividad de A (una difusividad efectiva),

será proporcional al número de saltos de A_i por unidad de tiempo y volumen, divididos por la concentración de A , es decir

$$\frac{A_i \text{ jumps}}{[A]} = \nu_m(A_i)S(I) \frac{[A_i]^*}{[A]},$$

con lo que finalmente se podrá escribir

$$D(A) = D(A_i)S(I) \frac{[A_i]^*}{[A]} = D(A)^*S(I). \quad (\text{A.10})$$

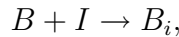
▷La presencia de $S(I)$ en estas ecuaciones anteriores se debe a que la especie A difunde a través de la especie A_i , que introduce la dependencia con I .

▷ A.2.5

Un ejemplo: energías de difusión del boro

En este apartado se presenta un ejemplo que muestra como relacionar el valor de la difusividad macroscópica (efectiva) del boro, a partir de su difusividad atomística. Para ello se va a establecer el vínculo entre la difusividad del B con la migración del B_i , si bien esta demostración no va a tener los estados de carga en consideración.

Suponiendo que hay equilibrio en la reacción



podemos escribir que

$$[B]^*[I]^*v_{\text{cap}}(1)\frac{6}{\lambda^2}D_m(I) = [B_i]^*, \frac{6}{\lambda^2}D_{\text{break}}(B_i) \quad (\text{A.11})$$

y notando que el boro difunde a través del B_i podemos definir la difusividad del boro,

$$D_m(B) = D_{0,m}(B) \times \exp(-E_b(B)/k_B T).$$

Para ello primero hacemos constar que

$$D_m(B)[B]^* = D_{B_i}[B_i]^*, \quad (\text{A.12})$$

y después tendremos que

$$D_m(B) = \frac{D_m(B_i)[B_i]^*}{[B]^*}.$$

Al usar las ecuaciones A.11 y A.12 se puede escribir que

$$D_m(B) = \frac{D_{0,m}(B_i) \exp(-E_m/k_B T)}{D_{0,\text{break}}(B_i) \exp(-(E_b(B_i) + E_m(I))/k_B T)} \times \\ \times [I]^*v_{\text{cap}}(1)D_{0,m} \exp(-E_m(I)/k_B T)$$

con lo que el uso de la concentración de intersticiales en equilibrio, ecuación A.20, lleva a la conclusión final de que:

$$E_m(B) = E_m(B_i) + E_f(I) - E_b(B_i) \quad (\text{A.13})$$

$$D_{0,m}(B) = \frac{D_{0,m}(B_i)}{D_{0,\text{break}}(B_i)} \frac{6}{\lambda^3} D_{0,\text{surface}} D_{0,m}(I) v_{\text{cap}}(1). \quad (\text{A.14})$$

▷ A.3

La superficie

La superficie juega un papel fundamental a la hora de establecer la concentración de equilibrio de los I s y V s, así como para regular la concentración de las impurezas y dopantes. Actualmente, hay dos modelos de comportamiento superficial en DADOS, uno dedicado a los I s y V s y el otro para las impurezas, de los cuales vamos a detallar algunos aspectos extra que no fueron tratados en capítulos precedentes. Estos aspectos son la longitud de recombinación, la forma en que la superficie se encarga de establecer la concentración de I y cómo se calcula la supersaturación de éstos. A partir de aquí sería sencillo hacer un estudio análogo para V .

▷ A.3.1

La distancia de recombinación

Aquellos átomos que estén situados a una distancia a la superficie menor o igual a λ son los que tienen posibilidad de recombinarse en el siguiente salto. Tomemos por tanto $[I]$ la concentración de intersticiales y ν la frecuencia de salto para dichos I s. Como los I s pueden saltar en tres dimensiones, únicamente 1/6 de ellos lo harán hacia la superficie, por lo que el número de I s recombinados en unidades de $\text{cm}^{-2}\text{s}^{-1}$, n será

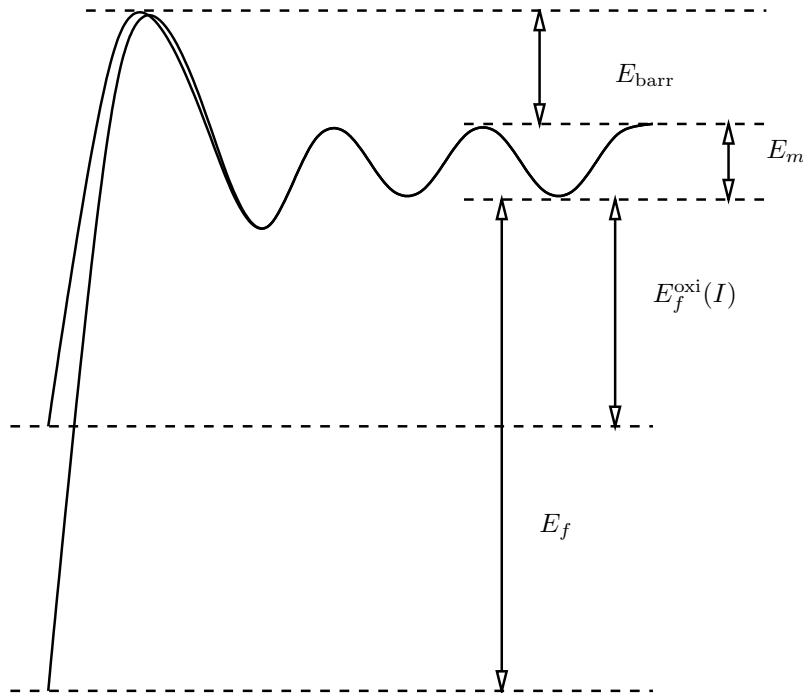
$$n = \frac{1}{6}[I]\lambda\nu,$$

donde sustituyendo $\nu = (6D)/\lambda^2$ tendremos para n que

$$n = \frac{D}{\lambda}[I]. \quad (\text{A.15})$$

Sólo nos falta suponer que no todos los átomos que inciden en la superficie son recombinados por ésta, sino que hay una probabilidad p de que sean absorbidos (es decir, suponemos que la superficie no tiene por qué ser un sumidero perfecto).

Modelo en la
sec. 8.7.1,
pág. 149.

Figura A.2: Significado de L_r

Añadiendo este factor tendremos que n pasa a ser

$$n = \frac{D}{\lambda} Cp$$

y, reescribiendo esta ecuación con otra notación

$$n = \frac{D}{L_r} C, \quad (\text{A.16})$$

donde hemos definido la longitud de recombinación L_r como:

$$L_r = \frac{\lambda}{p}. \quad (\text{A.17})$$

La figura A.2 intenta dar una idea intuitiva del significado físico de L_r como una barrera para los procesos de emisión y captura. La analogía no es perfecta porque la barrera debería cambiar las probabilidades con una dependencia con la temperatura, cosa que p no hace, ya que la suponemos constante para toda T .

► A.3.2

Coefficiente de captura superficial

Modelo en la
sec. 8.7.1,
pág. 151.

El coeficiente de captura en superficie será:

Departamento de Electricidad y Electrónica. Tesis de Ignacio Martín Bragado.

$$\nu_{cap}(\text{Surface}) = \underbrace{\frac{1}{6}}_{\text{Probabilidad}} \underbrace{\frac{6D_I}{\lambda^2}[I]}_{\text{número de Is}} \underbrace{\lambda Y \times Z}_{\text{volumen de captura}} \quad (\text{A.18})$$

▷ A.3.3

Concentración y capacidad de transporte de I

El producto concentración por difusividad para el intersticial en equilibrio es Modelo en la
sec. 8.7,
pág. 149.

$$[I]^* D(I) = \text{Prefactor}_0(I) \times \exp(-E_f(I)/k_B T) D_0(I) \times \exp(-E_m(I)/k_B T).$$

Podemos calcular la concentración de intersticiales $[I]^*$ en equilibrio si suponemos que bajo dicho régimen el número de I s emitidos desde la superficie debe igualar el de aquellos I s recombinados por ésta, de tal manera que el número de I s recombinados coincidirá con el coeficiente de captura que ha sido calculado en la ecuación A.18 multiplicado por la probabilidad de recombinación

$$[I]^* \underbrace{ZY\lambda}_{\text{Volumen}} \frac{1}{6} \underbrace{\frac{6}{\lambda^2} D_{0,m}(I) \exp(-E_m(I)/k_B T)}_{\nu \text{ de migración}} \underbrace{P_{\text{rec}}}_{\lambda/L}$$

y el número de intersticiales emitidos es:

$$\underbrace{\frac{2}{a^2} YZ}_{\text{sitios en superficie}} \times P_{\text{rec}} \times D_{\text{emission},0}^{\text{interface}}(I) \times \exp(-E_f(I)/k_B T) \times \frac{6}{\lambda^2} \times D_m(I) \times \exp(-E_m(I)/k_B T),$$

Modelo en la
sec. 8.7.1,
pág. 153.

Concentración de intersticiales en equilibrio

De la anterior ecuación se obtiene $[I]^*$

$$[I]^* = \frac{2}{a^2} \frac{6}{\lambda} D_{\text{emission},0}^{\text{interface}}(I) \times \exp(-E_f(I)/k_B T). \quad (\text{A.19})$$

Con lo que la concentración de intersticiales relativa, es decir, la concentración en equilibrio respecto a la densidad del silicio, será:

$$[I]_{\text{rel}}^* = \frac{3}{\sqrt{2}} D_{\text{emission},0}^{\text{interface}}(I) \times \exp(-E_f(I)/k_B T). \quad (\text{A.20})$$

Donde hemos tenido en consideración que $a = \sqrt{2}\lambda$.

▷ Utilizamos el término $6/\lambda^2$ para pasar de difusividades a frecuencias. En DADOS las constantes D_m están expresadas en unidades de difusividad, por lo que deben ser multiplicadas por esta constante para dar frecuencias (ver ecuación A.1).

El producto $D(I)[I]^*$

Partiendo de la ecuación A.20 y considerando que

$$D(I) = D_{0,m}(I) \times \exp(-E_m(I)/k_B T),$$

podemos escribir el producto $D(I)[I]^*$ como

$$D_i[I]_{rel}^* = \frac{3}{\sqrt{2}} D_{\text{emission},0}^{\text{interface}}(I) D_{0,m}(I) \times \exp[-(E_m(I) + E_f(I))/k_B T]. \quad (\text{A.21})$$

La medida experimental de este producto, concretamente su energía de activación, no se ve afectada por la presencia de trampas de intersticiales, por lo que su magnitud se obtiene de forma bastante precisa. No sucede lo mismo para el desglose en $E_m(I)$ y $E_f(I)$, para el que hay varios valores.

▶ **A.3.4****Supersaturación**

Modelo en la
sec. 8.9.5,
pág. 181.

La definición de la supersaturación es

$$S(I) = \frac{[I]}{[I]^*}, \quad (\text{A.22})$$

que equivale al número de intersticiales que emite la superficie (relacionado con $[I]^*$), dividido por el número de intersticiales capturados en la superficie (relacionado con $[I]$).

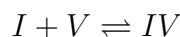
La frecuencia de emisión desde la superficie ya ha sido calculada anteriormente, Para la captura en la superficie tenemos que

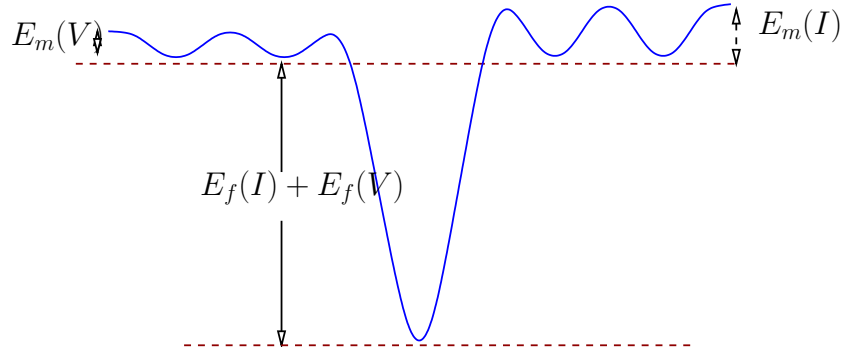
$$\underbrace{\frac{J(t_2) - J(t_1)}{6}}_{\text{saltos}} \quad \underbrace{\frac{\lambda}{X}}_{\text{proporción de captura}} \quad \underbrace{p_{\text{rec}}(I)}_{\text{probabilidad}},$$

porque el número de intersticiales que se captura en la superficie será el número de los que saltan hacia ella ($J(t)$ para un tiempo t), por su probabilidad de recombinación.

▶ **A.4****Equilibrio en el volumen interno**

La reacción de recombinación IV para el volumen interno es



Figura A.3: Energía de formación de un par IV .

con lo que, en equilibrio, tendríamos que el número de Is y Vs recombinados sería:

$$[I]^*[V]^* \frac{6}{\lambda^2} (D_0(I) + D_0(V)) \times \exp[-(E_m(I) + E_m(V))/k_B T]$$

y el número de pares emitidos:

$$D_{0,em}(I) \times \exp[-(E_f(I) + E_f(V) + E_m(I))/k_B T] + D_{0,em}(V) \times \exp[-(E_f(I) + E_f(V) + E_m(V))/k_B T]$$

Por lo que:

$$[I]^*[V]^* \propto \exp[-(E_f(I) + E_f(V))/k_B T] \quad (\text{A.23})$$

▷ La recombinación de los pares IV está implementada en DA-DOS utilizando el modelo de los pocket amorfos, pero su generación no ha sido implementada, dada su pequeñísima contribución, que es despreciable en los casos prácticos, que complicaría algo el modelo.

Modelo en la
sec. 8.4.1,
pág. 127.

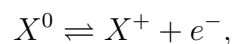
▷ A.5

Cálculos del modelo de carga

▷ A.5.1

Energías de formación

Suponemos que $E_f(X^0)$ es independiente del nivel de Fermi (e_F), con ello, en la reacción



podemos tomar

$$E_f(X^+) = E_f(x^0) + e_F - e(+, 0),$$

porque

$$E_f(X^0) + e_C - e(+, 0) = E_f(x^+) + e_C - e_F.$$

Para un estado de carga negativo, como por ejemplo X^- , escribimos

$$X^- \rightleftharpoons X^0 + e^- \Rightarrow E_f(X^-) = E_f(X^0) - (e_F - e(0, -)).$$

Estamos suponiendo que las reacciones de carga son más rápidas que aquéllas de estructura.

Las concentraciones están relacionadas como

$$\frac{[X^0]}{[X^+]} = \exp\left(\frac{e_F - e(+, 0)}{k_B T}\right)$$

$$\frac{[X^-]}{[X^0]} = \exp\left(\frac{e_F - e(0, -)}{k_B T}\right).$$

Teniendo en cuenta el factor de degeneración g :

$$\frac{[X^0]}{[X^+]} = \frac{g^0}{g^+} \times \exp\left(\frac{e_F - e(+, 0)}{k_B T}\right) = \exp\left(\frac{e_F - e'(+, 0)}{k_B T}\right),$$

donde

$$e'(+, 0) = e(+, 0) - k_B T \ln(g^-/g^0).$$

Debido a que las precedentes ecuaciones son generales, sin relación a un posible estado de equilibrio, es factible escribir que:

$$\frac{[X^0]}{[X^+]} = \frac{[X^0]^*}{[X^+]^*}$$

y por tanto

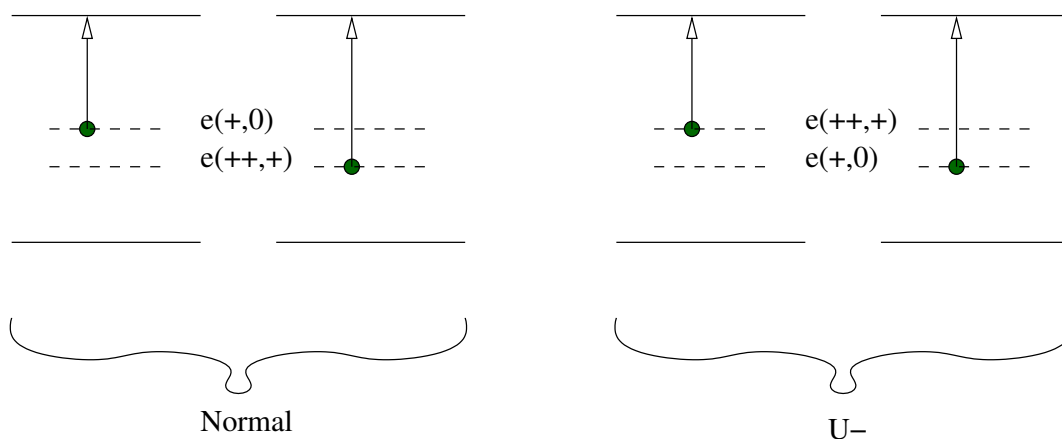
$$\frac{[X^0]}{[X^0]^*} = \frac{[X^+]}{[X^+]^*} = \frac{[X^-]}{[X^-]^*} = S(x),$$

lo que permite definir de forma sencilla la supersaturación $S(X)$.

▷ A.6

Implementación del modelo de carga

El modelo de carga ha sido implementado a partir de dos enfoques distintos:

Figura A.4: Configuraciones normal y U^- .

Estático Una actualización estática consiste en hacer un bucle a todas las partículas de la actualización y actualizar los estados de carga para todas y cada una de ellas. Debido a este carácter masivo, estas actualizaciones son lentas y deben ser realizadas de la forma más efectiva posible, intentando minimizar su número.

Dinámico Cada vez que una partícula móvil salta se hace una actualización dinámica sobre ella, es decir, se recalcula su estado de carga final. Con ello se consigue que la carga esté adecuadamente calculada cuando las partículas móviles van atravesando regiones con distinta concentración de dopantes.

▷ A.6.1

Actualización estática de carga

Consideremos un estado de carga denominado X^j . La probabilidad total para dicho estado será $P^j = [X^j]/[X]$ donde $[X]$ es la concentración total de la especie P .

Modelo en la
sec. 8.8.5,
pág. 162.

Escribamos ahora la probabilidad neutra relativa como:

$$P_0^j = \frac{[X^j]}{[X^0]}, \quad j = 0, -, --, +, ++$$

lo que implica que:

$$P_0^0 = 1$$

$$P_0^- = \frac{[X^-]}{[X^0]} = \frac{n}{n'(0, -)}$$

$$P_0^{--} = \frac{[X^{--}]}{[X^0]} = \frac{[X^{--}][X^-]}{[X^-][X^0]} = P_0^- \frac{n'(++ , +)}{n}$$

y, en general, $P_0^j \propto n^{-j}$.

De esta manera,

$$P^j = \frac{[X^j]}{[X]} = \frac{P_0^j}{\sum_j P_0^j}.$$

► **A.6.2**

Actualización dinámica de carga

Modelo en la
sec. 8.8.5,
pág. 163.

Tomemos, por ejemplo, los estados hipotéticos x^0, x^- con una probabilidad fija entre ellos. En equilibrio está claro que los cambios entre x^0 y x^- serán iguales que aquellos entre x^- y x^0 , por lo que

$$[X^0]\nu_n(X^0) \underbrace{P_d(0, -)}_{\text{Probabilidad decambio de 0 a - tras un salto}} = [X^-]\nu_m(X^-)P_d(-, 0)$$

Escojamos $P_d(i, j)$ de tal forma que se independiente de i :

$$P_d(i, j) = P_d(i', j) = P_d^j,$$

y con ello

$$[X^0]\nu_m(X^0)P_d^- = [X^-]\nu_m(X^-)P_d^0,$$

y

$$\frac{P_d^j}{[X^-]\nu_m(X^-)} = \frac{P_d^0}{[X^0]\nu_m(X^0)},$$

de tal forma que $P_d^j/([X^j] \times \nu_m(j))$ resulta ser constante, y en consecuencia

$$\frac{P_d^j}{P^j \nu_m(j)} = K.$$

Tenemos cierta libertad para elegir esta K , de manera que vamos a seleccionar

$$\frac{P_d^j}{P^j \nu_m(j)} = \frac{1}{\sum_i P^i \nu_m(i)},$$

y de esta manera llegamos al resultado final para la probabilidad de actualización dinámica de la carga:

$$P_d^j = \frac{P^j \nu_m(j)}{\sum_i P^i \nu_m(i)}. \quad (\text{A.24})$$

▷ A.7

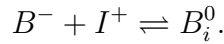
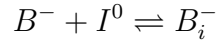
Un ejemplo: Difusión del boro y estados de carga

Modelo en la
sec. 8.8.8,
pág. 170.

En las siguientes reacciones estamos suponiendo que:

- Los estados de carga se encuentran en equilibrio.
- No hay reacciones entre especies que se repelan.
- Hay reversibilidad microscópica.
- Los eventos de ruptura conservan la carga.

DADOS no incluye los electrones en las reacciones, de tal manera que para conservar la carga sólo se permiten ciertas reacciones, y dichas reacciones no dependen del nivel de Fermi en cada punto. Así pues, si suponemos que la gran mayoría del boro se encuentra en el estado B^- tendremos:



Para obtener los parámetros adecuados hay que plantear que



De las reacciones anteriores, y teniendo en cuenta que la diferencia de energías entre $B^- + I^0$ y B_i^0 debe ser la misma siguiendo el término $B^- + I^+$

$$E_b(B_i^0) + e_{I^+} - e_F$$

o el término B_i^-

$$e_{B_i^-} - e_F + E_b(B_i^-)$$

se podrá escribir que:

$$E_b(B_i^0) + e_{I^+} - e_F = e_{B_i^-} - e_F + E_b(B_i^-) \Rightarrow E_b(B_i^0) = E_b(B_i^-) + e_{B_i^-} - e_{I^+}.$$

En cuanto a los prefactores, las reacciones de equilibrio relacionadas con las ecuaciones (frecuencia de formación igual que la frecuencia de ruptura) A.25 son:

$$D_{break}(B_i^0)[B_i^0] = v_{cap} D_m(I^+)[B^-][I^+]$$

$$D_{break}(B_i^-)[B_i^-] = v_{cap} D_m(I^0)[B^-][I^0].$$

Si dividimos miembro a miembro las anteriores ecuaciones:

$$\frac{D_{break}(B_i^0)}{D_{break}(B_i^-)} \times \exp\left(\frac{e_{B_i^-} - e_F}{k_B T}\right) = \frac{D_m(I^+)}{D_m(I^0)} \times \exp\left(\frac{e_{I^+} - e_F}{k_B T}\right).$$

Y dichas reacciones darán las siguientes ecuaciones para el prefactor:

$$\frac{D_{0,break}(B_i^-)}{D_{0,m}(I_0)} = \frac{D_{0,break}(B^0)}{D_{0,m}(I^+)} = \frac{D_{0,break}(B^+)}{D_{0,m}(I^{++})}.$$

Por último, la difusividad del boro estará regida por:

$$D_B[B] = D_{B_i^-}[B_i^-] + D_{B_i^0}[B_i^0] + D_{B_i^+}[B_i^+].$$

Suponemos que $[B] \approx [B^-]$ de tal manera que $D_B \equiv D_B^-$ y así

$$D_B = D_{B_i^-} \frac{[B_i^-]}{[B^-]} + D_{B_i^0} \frac{[B_i^0]}{[B^-]} + D_{B_i^+} \frac{[B_i^+]}{[B^-]}.$$

Es decir:

$$D_B = D_{B_i^-} \frac{[B_i^-]}{[B^-]} + D_{B_i^0} \frac{[B_i^-]}{[B^-]} \underbrace{\frac{[B_i^0]}{[B_i^-]}}_{\left(\frac{[B_i^0]}{[B_i^-]}\right)_{int} \frac{p}{n}} + D_{B_i^+} \frac{[B_i^-]}{[B^-]} \underbrace{\frac{[B_i^+]}{[B_i^-]}}_{\left(\frac{[B_i^+]}{[B_i^-]}\right)_{int} \left(\frac{p}{n}\right)^2}$$

Que, comparando con la notación del SUPREM

$$D_B = D_{xB} + D_{PB} \frac{p}{n_i} + D_{PPB} \left(\frac{p}{n_i}\right)^2$$

permite identificar cada miembro como:

$$D_{xB} = D_{B_i} \frac{[B_i^-]}{[B^-]}$$

$$D_{PB} = D_{B_i^0} \frac{[B_i^-]}{[B^-]} \left(\frac{[B_i^0]}{[B_i^-]}\right)_{int}$$

$$D_{PPB} = D_{B_i^+} \frac{[B_i^-]}{[B^-]} \left(\frac{[B_i^+]}{[B_i^-]}\right)_{int}.$$

La figura A.5 muestra las energías de migración para cada estado de carga del boro, donde se ha tenido en cuenta que $E_{dif} = \min(E_f + E_m)$.

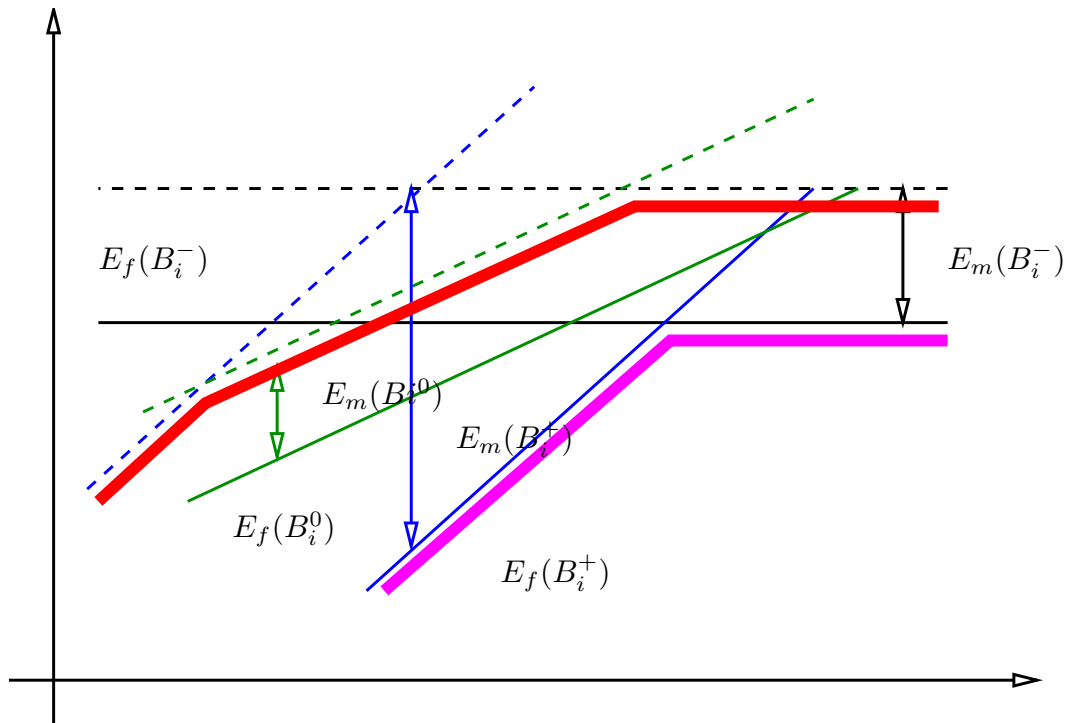


Figura A.5: Energías de difusión y estados de carga del boro.

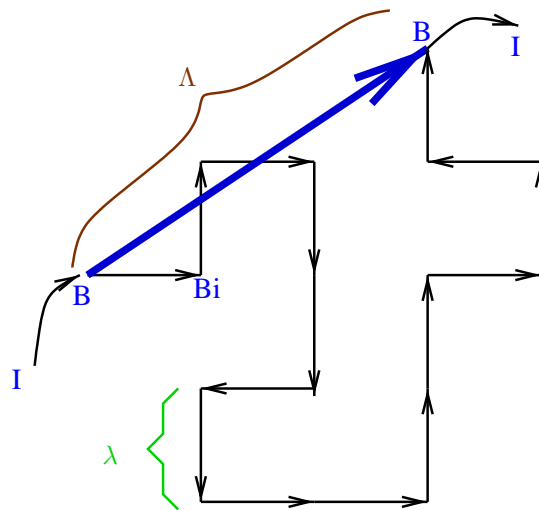


Figura A.6: Distancia de salto (λ) y salto largo (Λ).

▷ A.7.1

Difusividad del boro y saltos largos

El propósito de este apartado es estudiar los saltos largos (Λ) del boro al difundir. Si λ corresponde con la longitud de salto cuando una partícula realiza un único evento de difusión, la distancia recorrida por una partícula B_i hasta que se disocia es llamada Λ (ver figura A.6).

Modelo en la sec. 8.8.8, pág. 171.

Se define Λ como

$$\Lambda = \lambda \sqrt{\frac{1}{6} \frac{\nu_m}{\nu_{break}}} = \lambda \sqrt{\frac{1}{6} \frac{D_m(B_i^-)[B_i^-] + D_m(B_i^0)[B_i^0] + D_m(B_i^+)[B_i^+]}{D_{break}(B_i^-)[B_i^-] + D_{break}(B_i^0)[B_i^0] + D_{break}(B_i^+)[B_i^+]}}$$

de donde sabemos que:

$$D_{break}[B_i^-] = v_{cap} D_m(I^0)[I^0][B^-].$$

$$D_{break}[B_i^0] = v_{cap} D_m(I^+)[I^+][B^-].$$

$$v_{cap}[Si] = 3\lambda^3 \frac{8}{a^3} = \frac{3a^3}{2\sqrt{2}} \frac{8}{a^3} = 36\sqrt{2}.$$

y por tanto una expresión simplificada para Λ resulta ser:

$$\Lambda = \lambda \sqrt{\frac{1}{6} \frac{D_m(B_i^-) \frac{[B_i^-]}{[B^-]} + D_m(B_i^0) \frac{[B_i^0]}{[B^-]} + D_m(B_i^+) \frac{[B_i^+]}{[B^-]}}{v_{cap}(D_m(I^0)[I^0] + D_m(I^+)[I^+] + D_m(I^-)[I^-])}} = \lambda \sqrt{\frac{1}{36\sqrt{2}} \frac{D_B}{D_I}}. \quad (A.26)$$

▷ A.8

Energías y prefactores de activación efectivos para gráficas de Arrhenius

El objetivo de esta sección es calcular unos prefactores y energías de activación efectivos para la difusión de las especies suponiendo un comportamiento de *plot* de Arrhenius que tenga en cuenta los efectos de carga.

Cuando se calcula la magnitud definida como

$$\delta_x = \exp\left(\frac{e_i - e_X}{k_B T}\right) \quad (A.27)$$

hay que tener en cuenta los efectos de estrechamiento del ancho del gap semiconductor con la temperatura, que hacen que

$$e_C^T - e_X^T = \frac{E_g(T)}{E_g(0)} (e_C - e_X),$$

y

$$e_C^T - e_i^T = \frac{1}{2} \left(E_g(T) - k_B T \ln \frac{N_V}{N_C} \right),$$

implicando, por tanto

$$e_i^T - E_X^T = (e_i - e_X) \frac{E_g(T)}{E_g(0)} + \frac{k_B T}{2} \ln \frac{N_V}{N_C}. \quad (\text{A.28})$$

Con la combinación de las ecuaciones A.27 y A.28 podemos escribir

$$\delta_X^T = \sqrt{\frac{N_V}{N_C}} \times \exp \left(-\frac{E_g(T)}{E_g(0)} \times \frac{e_X - e_i}{k_B T} \right), \quad (\text{A.29})$$

y por tanto

$$E_{\text{act}} = -\frac{d \ln \delta_x^T}{d(1/(k_B T))},$$

y

$$\ln(\text{prefac}) = \ln \delta_x^T + \frac{E_{\text{act}}}{k_B T}.$$

De las anteriores expresiones, y utilizando las relaciones entre $E_g(T)$ y $E_g(0)$ los resultados finales quedan como (Mathiot and Pfister, 1984):

| | E_{act} | $E_{\text{act}}(1000)$ |
|---------------|--|------------------------------------|
| δ_{X-} | $(e_{X-} - e_i) \left[1 + \frac{ABT^2}{E_g(0)(B+T)^2} \right]$ | $1,096(e_{X-} - e_i)$ |
| δ_{X+} | $(e_i - e_{X+}) \left[1 + \frac{ABT^2}{E_g(0)(B+T)^2} \right]$ | $1,096(e_i e_{X+})$ |
| | <i>pref</i> | <i>pref</i> (1000) |
| δ_{X-} | $\sqrt{\frac{N_V}{N_C}} \exp \left[\frac{AT(T+2B)}{K(B+T)^2} \frac{e_{X-} - e_i}{E_g(0)} \right]$ | $0,753 \exp[3,9824(e_{X-} - e_i)]$ |
| δ_{X+} | $\sqrt{\frac{N_C}{N_V}} \exp \left[\frac{AT(T+2B)}{K(B+T)^2} \frac{e_i - e_{X+}}{E_g(0)} \right]$ | $1,328 \exp[3,9824(e_i - e_{X+})]$ |

(A.30)

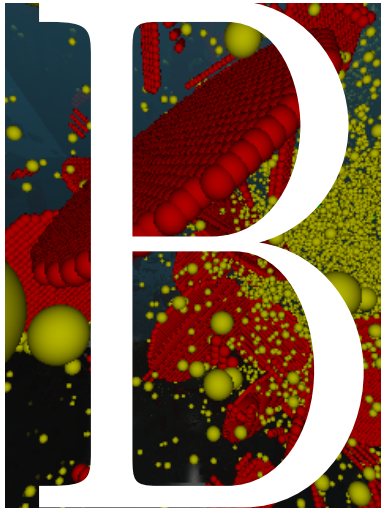
Donde se ha tomado $E_g(0) = 1,17 \text{ eV}$, $A = 4,72e - 4 \text{ eV/K}$, $B = 636 \text{ K}$ y $N_V/N_C = 0,567$.

▷ A.9

Notación

La notación seguida en este apéndice, para una hipotética especie A , ha sido:

| <i>Símbolo</i> | <i>Descripción</i> |
|-------------------------------|--|
| $D_{0,m}(A)$ | Prefactor de migración para una especie A . |
| $D_{0,em}^B(A)$ | prefactor de A para emitir B s. |
| $D_{0,break}(A)$ | prefactor de ruptura de A . |
| $D_{0,surface}(A)$ | prefactor de emisión desde la superficie de la especie A . |
| $D(A)$ | Difusividad de A . |
| $E_b(S)$ | Energía de enlace (ruptura) de A . |
| $E_b^B(A)$ | Energía de enlace de A cuando captura una especie B . |
| $E_m(A)$ | Energía de migración de A . |
| $E_f(A)$ | Energía de formación de A . |
| $\nu_{em}^B(A)$ | Frecuencia de emisión de B s por parte de A . |
| $\nu_m(A)$ | Frecuencia de migración de A . |
| $\nu_{break}(A)$ | Frecuencia de ruptura de A . |
| $[A]^*$ | concentración de A en el equilibrio. |
| $[A]$ | concentración de A . |
| $V_{cap}(n)$ | volumen de captura por parte de un cluster de tamaño n . |
| $V_{cap}(A)$ | volumen de captura de una especie A . |
| $\nu_{cap}^{A \rightarrow B}$ | frecuencia de captura de B por parte de A por unidad de volumen. |



Arquitectura del proyecto UVAS

| | |
|---|-----|
| B.1. Introducción . . . | 347 |
| B.2. Propósitos | 347 |
| B.3. Implementación . | 349 |
| B.4. Opciones globales | 353 |
| B.5. Gestionando los comandos: El "Script Wizard" . | 357 |
| B.6. Los datos de salida: DataBrowser | 371 |
| B.7. Gestionando las gráficas | 385 |
| B.8. Extrayendo los datos en un modo no gráfico . | 396 |

El programa UVAS (University of Valladolid Atomistic Simulator) pretende crear un entorno de trabajo que integre todos las herramientas necesarias para analizar los resultados de las simulaciones junto con los propios simuladores. Ahora bien, hacer este tipo de entorno supone construir un programa que no sólo sea capaz de visualizar los datos de la manera apropiada, sino que a la vez gestione y ejecute distintos tipos de simulaciones atomísticas. En este capítulo se expone el funcionamiento del entorno UVAS, las ideas directrices para su implementación y algunos fragmentos de código para mostrar cómo ha sido desarrollado.

▷ B.1

Introducción

Configurar los parámetros de entrada de una simulación, ejecutar, visualizar e interpretar los datos provenientes de ésta son tareas comunes que pueden requerir una considerable cantidad de tiempo y esfuerzo. Por ello, nos gustaría resolver estas situaciones de una forma genérica para cualquier tipo de simulación atomística.

- Una simulación atomística necesita una lista inicial (un *script*), que son los diferentes comandos e instrucciones que damos al simulador. Por ejemplo, en MD se podría hacer un recocido a 800 °C, o en BCA una implantación a una cierta temperatura y energía, o bien en kMC podemos realizar distintos recocidos a temperaturas diferentes.
- Una vez que se conocen los comandos (las instrucciones que la simulación va a procesar), la simulación en concreto empieza a ejecutarse y a producir datos de salida.
- Cuando la simulación termina, o bien mientras está funcionando, produce una serie de resultados. Cada tipo de simulación tendrá un diferente tipo de datos, por lo que es importante saber qué tipo de salida produce la simulación.

Estas ideas son muy importantes para determinar la estructura del programa UVAS. De esta manera, las simulaciones son tratadas como objetos, los cuales necesitan una cierta entrada y producen unos datos de salida (ver figura B.1).

▷ B.2

Propósitos

▷ B.2.1

Propósitos del entorno

El entorno UVAS ha sido programado para:

- Ser lo más general posible. No damos por supuesto el tipo de simulación que va a ser ejecutado en UVAS, de manera que UVAS debe ser capaz de manejar y dibujar cualquier tipo de datos provenientes de simulaciones atomísticas.
- Ser *portable*. Actualmente UVAS se ejecuta tanto en plataformas Windows como en Unix.

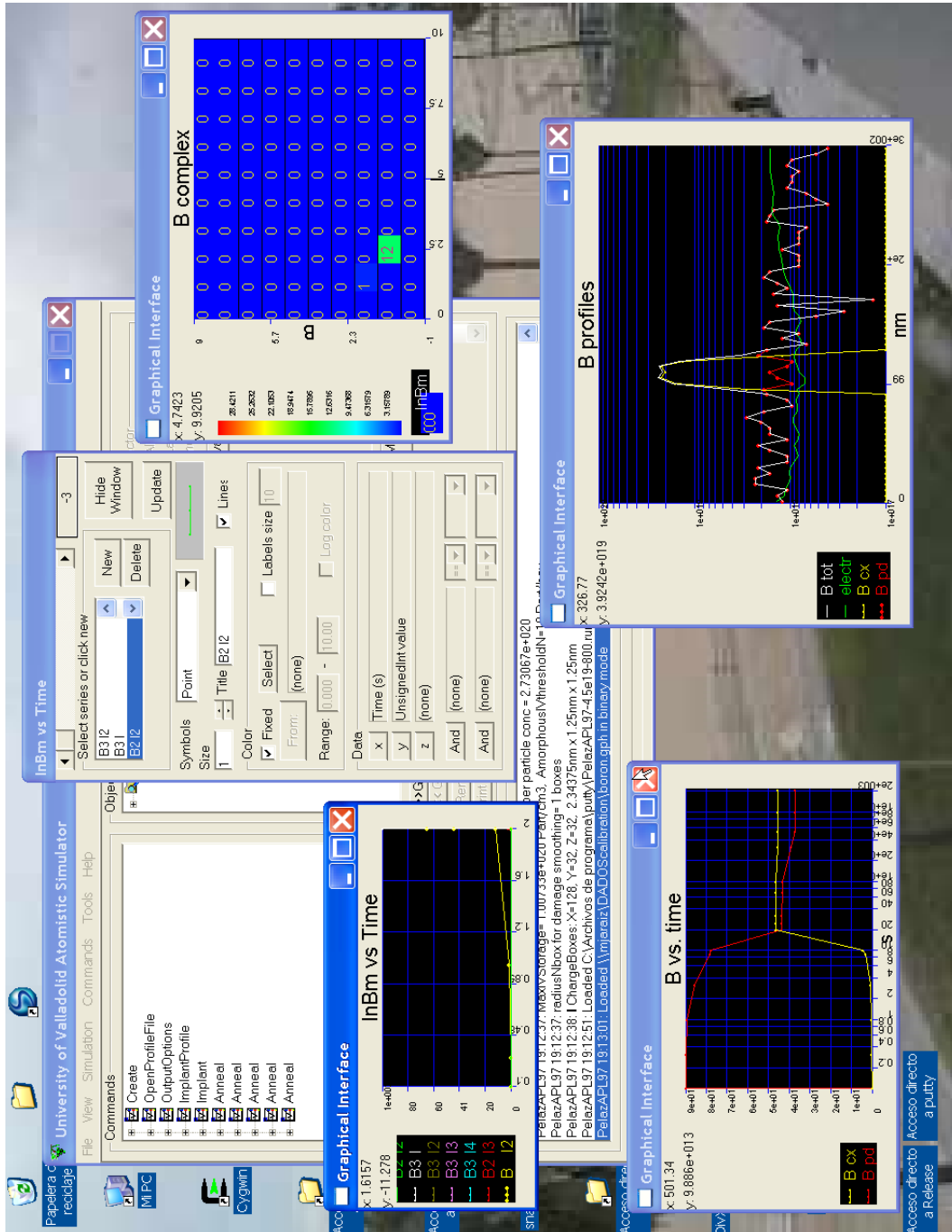


Figura B.1: Vista del Simulador Atomístico de la Universidad de Valladolid, UVAS.

- Ser sencillo de usar para los usuarios. El programa UVAS es un programa visual, en el cual las operaciones pueden ser fácilmente realizadas con el ratón. Los gráficos expuestos pueden ser rotados y desplazados, y sus datos asignados con un simple “click”.
- Ser eficiente en el uso de los recursos del sistema.
- Dar la adecuada información sobre qué está sucediendo en la simulación (ver imagen B.2).

▷ B.2.2

Propósitos de los simuladores

Los propósitos para los diferentes simuladores son:

- Representar adecuadamente los modelos físicos.
- Ser lo más rápidos posibles. Se ha hecho un esfuerzo especial en las partes críticas de los simuladores para lograr operaciones rápidas.
- Ser flexibles y extensibles. En un futuro habrá nuevos modelos y fenómenos físicos que simular, y los simuladores deberían estar preparados para ello.

▷ B.3

Implementación

En vista de la idea, expuesta anteriormente, según la cual las simulaciones deberían ser como objetos con unas ciertas entradas y salidas desde, y hacia, el mundo exterior, se ha escogido el lenguaje de programación C++ para desarrollar el código. C++ es un lenguaje extraordinariamente potente, moderno, extendido, portable y orientado a objetos. Aunque FORTRAN es un lenguaje bastante extendido en la comunidad científica, se ha preferido C++ por las anteriores razones, así como por la gran cantidad de aplicaciones profesionales que están escritas en C y C++, que avalan las capacidades de este lenguaje.

Las ideas anteriores se han implementado en el código de la siguiente forma. Cada tipo de simulación es un gran objeto en UVAS, y cada uno de estos objetos permite ciertos comandos, que generan el “script”, y producen ciertos resultados de salida. Todos los objetos UVAS presentan la misma interfaz estándar, que se dibuja en la figura B.3, y que permite tener diferentes simulaciones atomísticas en un mismo programa, así como dibujar y visualizar los datos gestionados sin necesidad de programas extra. Desde este punto de vista, el programa UVAS tiene dos grandes partes:

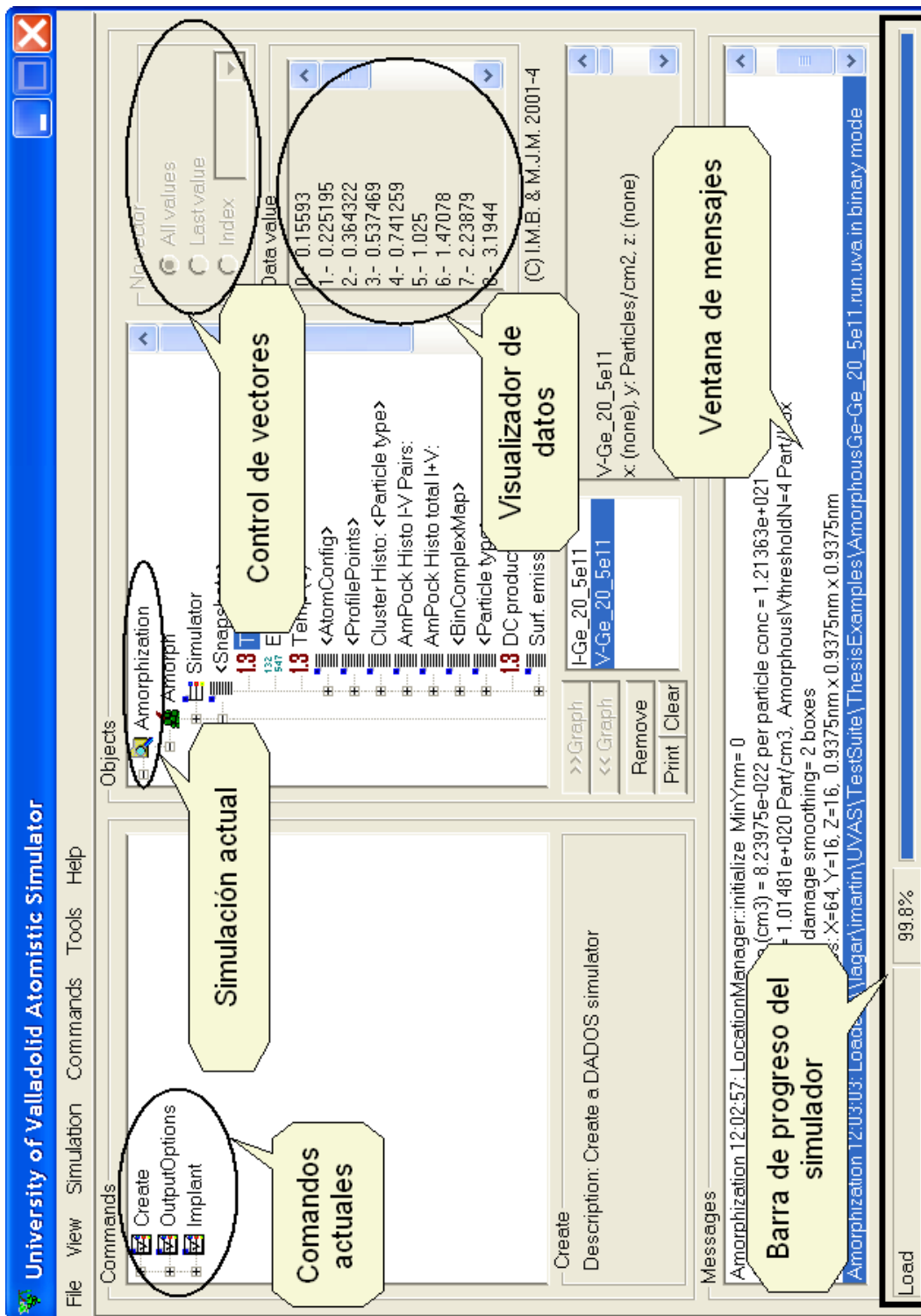


Figura B.2: La ventana de mensajes muestra el cuaderno de bitácora de las simulaciones, y una barra de progreso va mostrando al usuario el porcentaje de simulación que se ha realizado.

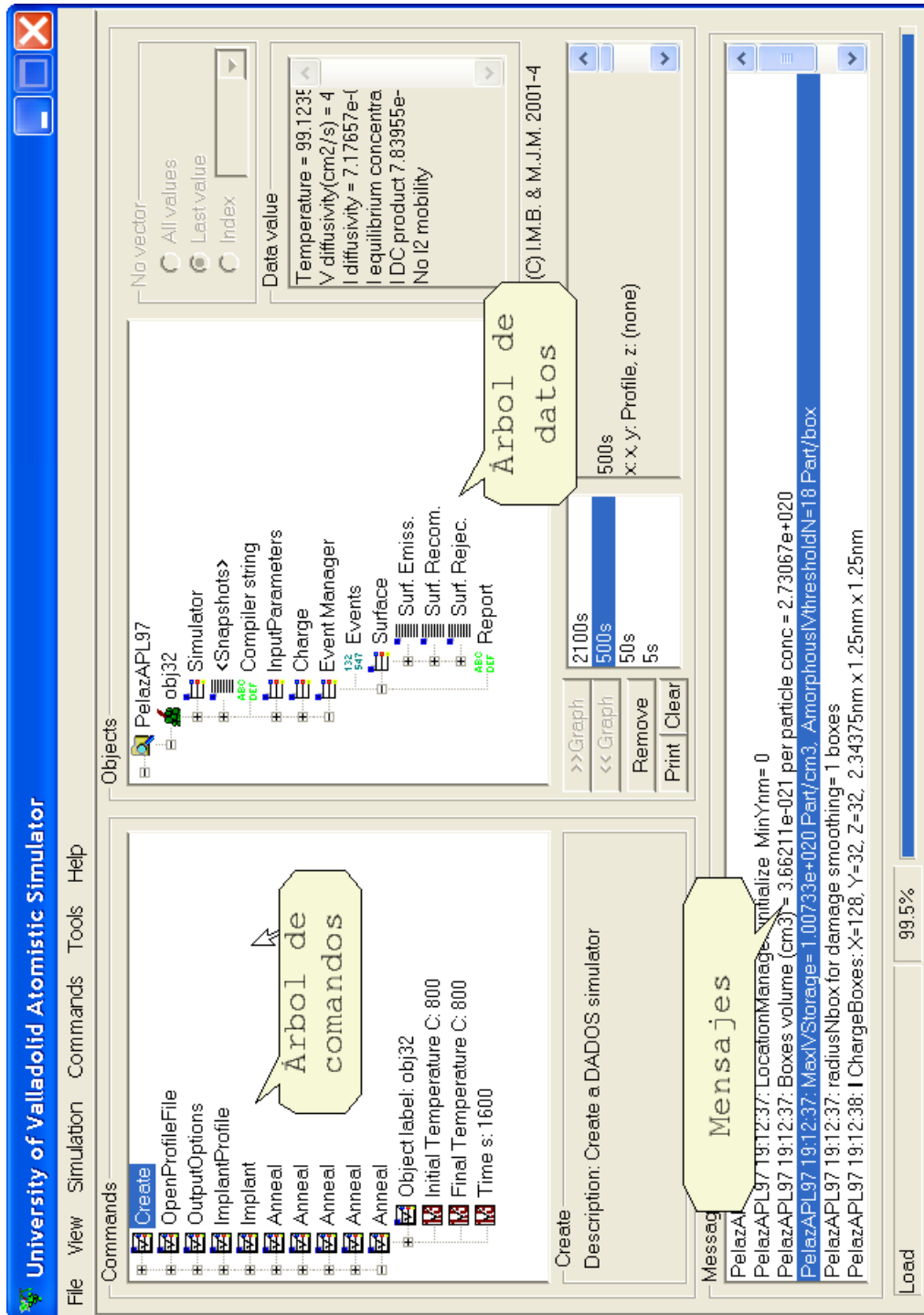


Figura B.3: Entorno común para realizar simulaciones atomísticas. La ventana de la izquierda (con el árbol de comandos) muestra los comandos de entrada, la de la derecha (un árbol de datos), permite navegar por los datos de salida.

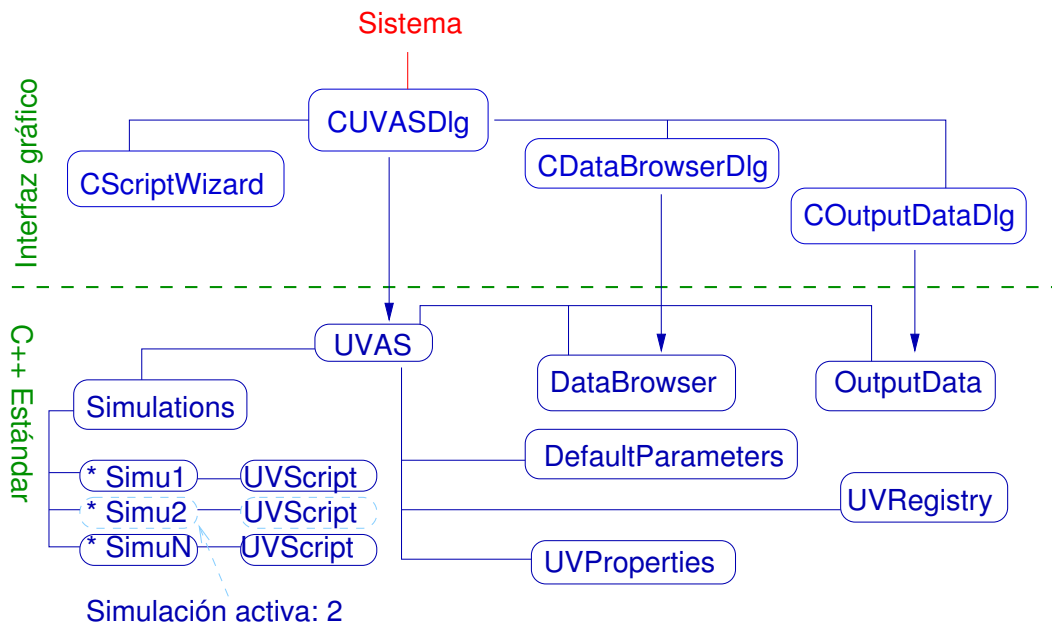


Figura B.4: Estructura del UVAS desde un punto de vista del programador. Grandes partes del código son C++ estándar, y por tanto se pueden compilar en todas las plataformas existentes. Otras porciones, en cambio, dependen del sistema operativo (Unix, Windows) y deben ser “portadas”. En la medida en que ha sido posible se han separado y aislado las clases estándar de aquellas que dependían del sistema operativo.

- Por una parte el código que permite dibujar, configurar, manejar, grabar y leer las simulaciones, que sería el entorno UVAS propiamente dicho.
- Por otra parte el código que realiza las simulaciones (los módulos de simulación).

Una representación del código que maneja las simulaciones está en la figura B.4. Este código se puede dividir en tres partes:

- Código que gestiona los gráficos. Este código es responsable de generar las representaciones en dos y tres dimensiones.
- El “Script wizard”. Este código se preocupa de las instrucciones de entrada (comandos) de los simuladores.
- El “Data Browser”. Este código maneja la salida de los simuladores, y además permite una conexión con el código gráfico, para producir resultados visibles.

Para hacer lo más fácil posible la portabilidad del código las clases han sido aisladas, siempre que ha sido posible, en clases que compilan con las librerías C++ estándares, y aquellas que dependen del sistema operativo concreto utilizado.

A su vez, el código que se encarga de los simuladores se puede dividir en:

- Código de dinámica molecular (MD). Este código realiza simulaciones de dinámica molecular. Dispone de las siguientes partes:
 - Integradores. Los distintos algoritmos para integrar las ecuaciones diferenciales del movimiento newtoniano. Se han implementado dos integradores concretos, el de Verlet, y el de Verlet isobárico.
 - Potenciales. Las distintas formas de calcular la energía de un sistema físico. Se han implementado dos potenciales concretos, el de Stillinger–Weber, para átomos de silicio, y el de Lennard–Jones, para gases.
- Código de Monte Carlo cinético. Se ha sido añadido el simulador de Monte Carlo cinético DADOS al UVAS como módulo de simulación.
- Optimizadores que usan algoritmos genéticos. Este código busca entre distintas configuraciones atómicas para encontrar aquella de mínima energía.
- Conjunto Atómico Cinético. Un experimento (inconcluso) para establecer un puente entre la dinámica molecular y el Monte Carlo cinético.
- Monte Carlo cinético de red. Un simulador LMC para el depósito de capas delgadas de aluminio.

▷ B.4

Opciones globales

Las opciones globales son manejadas por el `uvas` con la clase `UVGlobalProperties`. Esta clase se limita a ser un contenedor para las distintas opciones pasadas por el usuario:

```
struct UVGlobalProperties
{
    UVGlobalProperties() { bBinarySave = true; bAutoexpand = true;
    bDBListLimit = true; bShowPointers = false; uiMaxNumberLastValues = 15;
    bAskSimulationAndObject = true; bDBEnumeration = true;
    bDBWriteOutput = false; bChangingTreeWarning = false; uiDBListLimit = 1500;
    uiVerbosity = 1; }

    mutable bool bBinarySave;
    bool bAutoexpand;
    bool bDBListLimit;
    bool bDBEnumeration;
    unsigned int uiDBListLimit;
    bool bDBWriteOutput; //Write UVOutputData instead of print it.
    bool bShowPointers;
    bool bAskSimulationAndObject; //Ask when loading a UVSeries
```

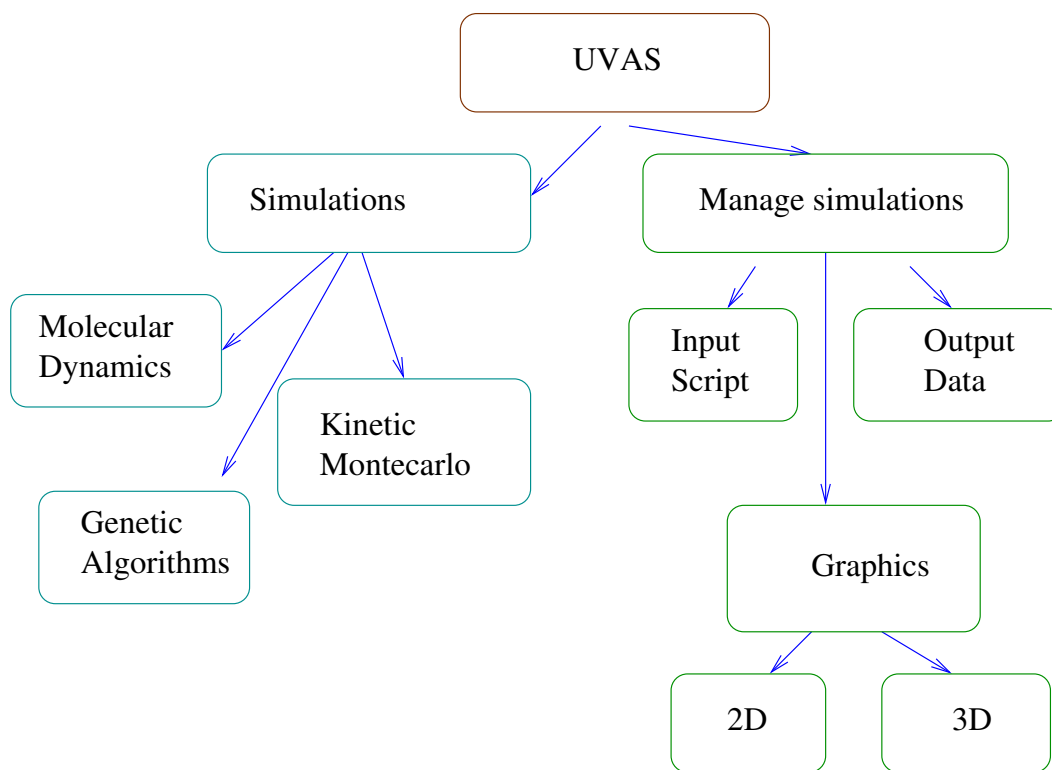


Figura B.5: Diferentes partes del código del UVAS desde el punto de vista del usuario.

```
    //that do not contain the correct simulation and/or object
    bool bChangingTreeWarning;
    unsigned int uiMaxNumberLastValues;
    unsigned int uiVerbosity;    // low, med, high, max.
};
```

El significado de estas opciones es:

Binary Save. Graba las simulaciones (extensión .uva) como textos ASCII o binario.

Auto Expand. Mientras crea el árbol del “Script Wizard”, va expandiendo los diferentes ítems que cuelgan de él.

Data Browser List Limit. Limita o no el número de elementos mostrados al valor indicado en `uiDBListLimit`.

Data Browser enumeration. Enumera los elementos de un vector al ser listados o no.

Write Output. Opción que permite que ciertos datos de salida (gestionados con la clase `OutputData`), sean grabados en ficheros o listados en una ventana.

Show Pointers. Opción para mostrar o esconder los punteros, (un tipo de dato especial), en el “Data Browser”.

Ask Simulation and Object. Opción para especificar si el programa preguntará al usuario sobre el origen de los datos leídos de un fichero, si no los encontrase en la memoria.

Changing Tree Warning. Muestra o no un mensaje cuando el programa cambie los índices de los vectores de manera automática como respuestas a ciertos eventos del usuario.

Verbosity. Nivel de información que queremos que se emita por la ventana de mensajes.

Las anteriores opciones se especificarán utilizando el cuadro de diálogo que se muestra en la figura B.6.

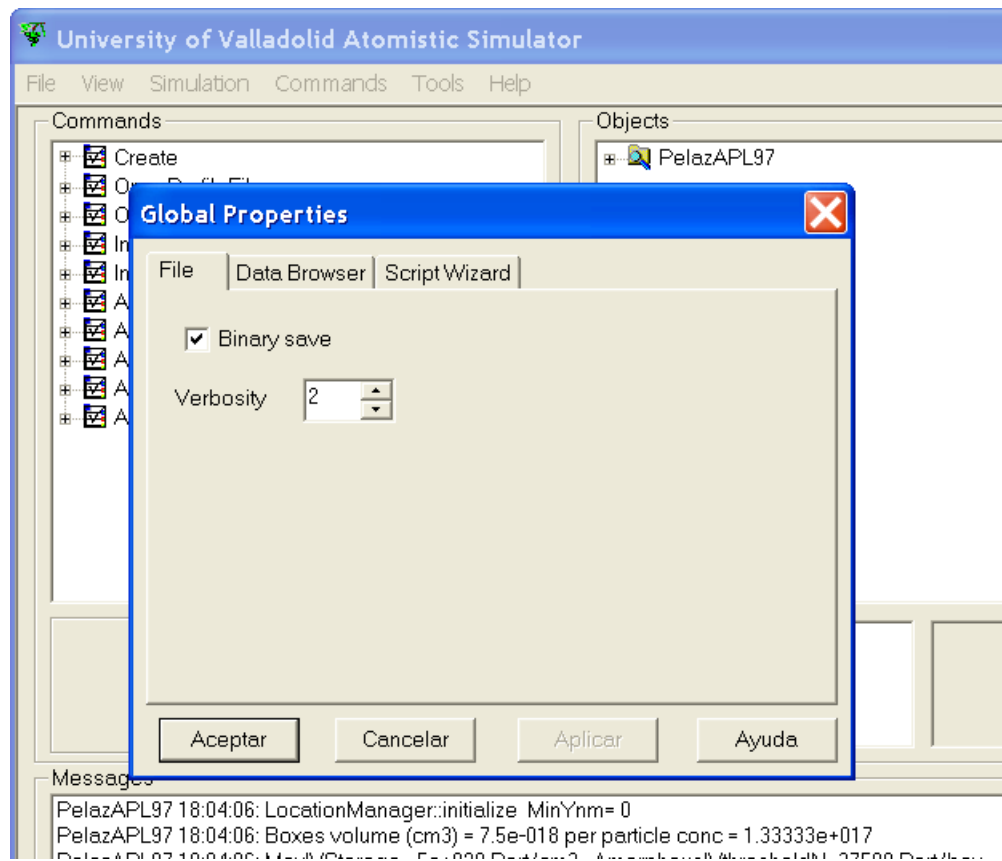


Figura B.6: Cuadro de diálogo que permite al usuario cambiar las opciones globales del programa UVAS.

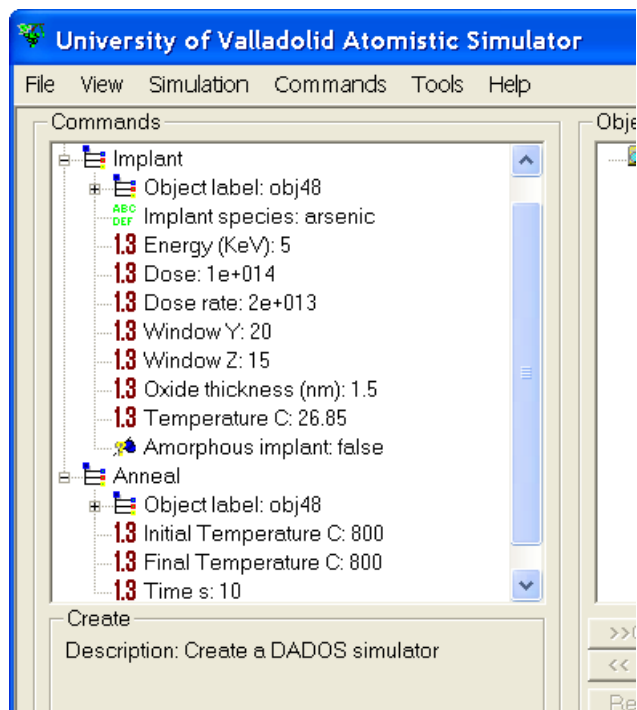


Figura B.7: Vista parcial del “Script Wizard”. Se utiliza una estructura de árbol para mostrar los comandos y sus opciones.

▷ B.5

Gestionando los comandos: El “Script Wizard”

▷ B.5.1

Introducción

“Script Wizard” es el nombre general que hemos dado a todo el conjunto de algoritmos y clases que se encargan de organizar las instrucciones y comandos. Básicamente este “Script Wizard” establece nexos de unión entre las instrucciones del usuario y los comandos internos, preguntando los parámetros que se necesitan al usuario para pasárselos a las funciones. El “Script Wizard” muestra toda esta información en una estructura en forma de árbol, como se puede ver en la figura B.7, en la que se intentan clarificar las diferentes dependencias entre los objetos.

Todos los objetos principales (módulos de simulación) del UVAS, a los que llamamos `UVObjects`, presentan una interfaz que es utilizada por el “Script Wizard”. Dicha interfaz define los comandos y parámetros que cada función requiere.

No obstante, ciertos comandos pueden requerir el uso de otros objetos adicionales o externos (por ejemplo, un módulo de dinámica molecular requiere su potencial e integrador). El “Script Wizard” es capaz de entender cuándo un objeto

requiere la existencia de otro, y permite al usuario que lo cree sobre la marcha, antes incluso de la creación del otro, para así poder luego referenciarlo. En el mismo ejemplo de la dinámica molecular, el objeto principal `DynAtomSet` necesita una configuración atómica (`AtomicConfiguration`) para saber la posición de todos los átomos que habrá en la simulación. De esta manera, cuando el usuario selecciona un `DynAtomSet`, el “Script Wizard” da al usuario la posibilidad de elegir algún `AtomicConfiguration` ya creado, o bien de crear uno nuevo. Cuando el usuario seleccione la creación de uno nuevo, el “Script Wizard” insertará la instrucción adecuada para crear el nuevo objeto, e investigará recursivamente si este nuevo objeto necesita a su vez de otros. El proceso acabará cuando todas las dependencias entre objetos queden satisfechas.

Para poder cumplir con los requerimientos anteriores se han creado dos tipos de comandos, llamados de “trabajo” y de “construcción”. El objetivo de un comando constructor es crear un “UObject” partiendo de la nada, esto es, construir la estructura básica necesaria, pedir al sistema la memoria necesaria para el objeto y poblarlo de valores iniciales. A continuación, los comandos de “trabajo” ejecutarán las tareas requeridas en el objeto ya creado.

Esta división no es visible para el usuario. El usuario puede ejecutar cualquier comando que tenga a su disposición pero, cuando elige un comando complejo, es decir, uno que requiera a su vez la creación de otros objetos, se guiará al usuario a través de una serie de comandos constructores hasta que se hayan generado todos los objetos requeridos (ver figura B.8). Finalmente el usuario puede cambiar los valores por defecto que se van dando a los comandos y, por último, comenzar la simulación.

Internamente, gracias al planteamiento recursivo del problema, hasta los parámetros son considerados como comandos, siendo el proceso total completamente transparente para el usuario.

A continuación, se van a explicar las diferentes clases creadas para gestionar todo el “Script Wizard”, ayudados por la figura descriptiva B.9. Finalmente, la estructura del “Script Wizard” se dibuja en la figura B.10.

▷ B.5.2

La clase de comandos básica: `UVCmd`

`UVCmd` es la clase base para todos los comandos del UVAS. Implementa la estructura esencial para leer y guardar comandos, los tipos y descripciones de cada uno de ellos, así como los algoritmos para manipular adecuadamente los sub-comandos y sus diferentes relaciones, junto con si han sido ejecutados aún o no.

Una muestra del fichero de cabecera de su código fuente es:

```
// Command.h: interface for the Command class.
//
#ifdef !defined(_UVCMD_H_)
```

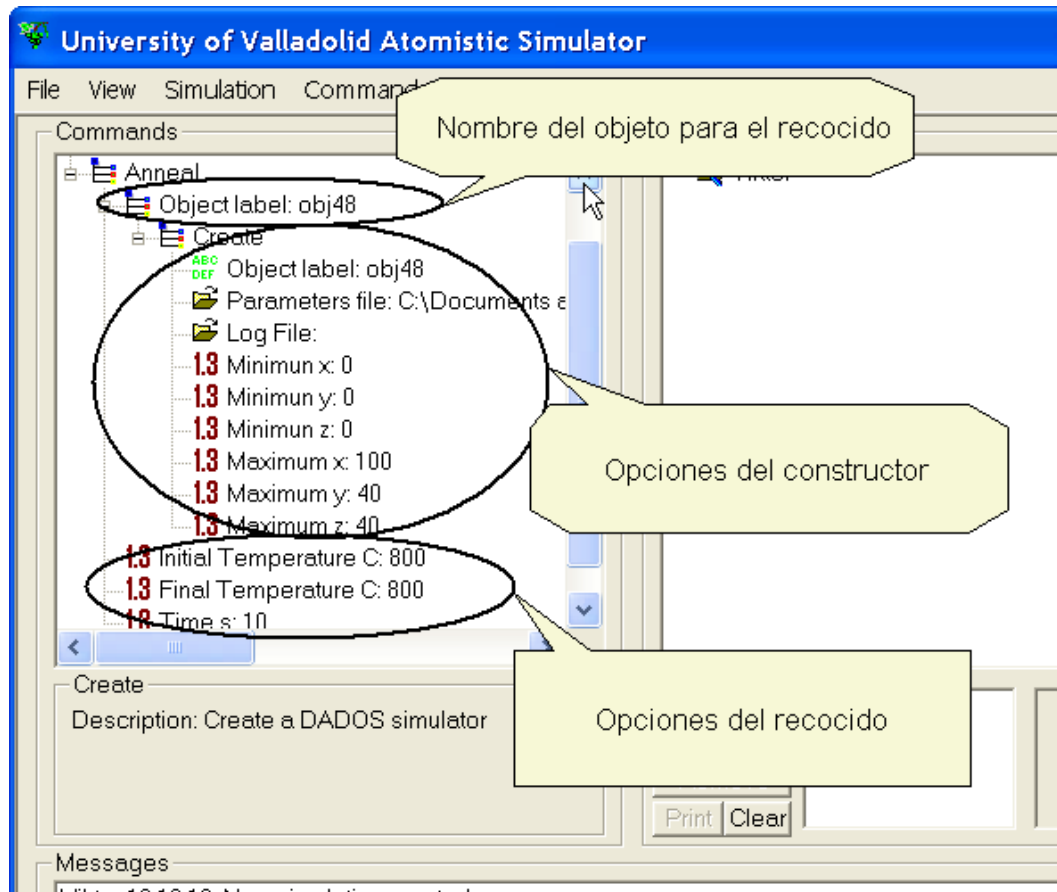



Figura B.8: Vista del comando de recocido. Este comando requiere un objeto sobre el que operar, el cual se asigna a través del parámetro “Object label”, que a su vez crea o ajusta (“Create”) los constructores adecuados. Después de ello se muestran los parámetros del recocido, para que el usuario los cambie si ésta es su elección: el tiempo y las temperaturas inicial y final.

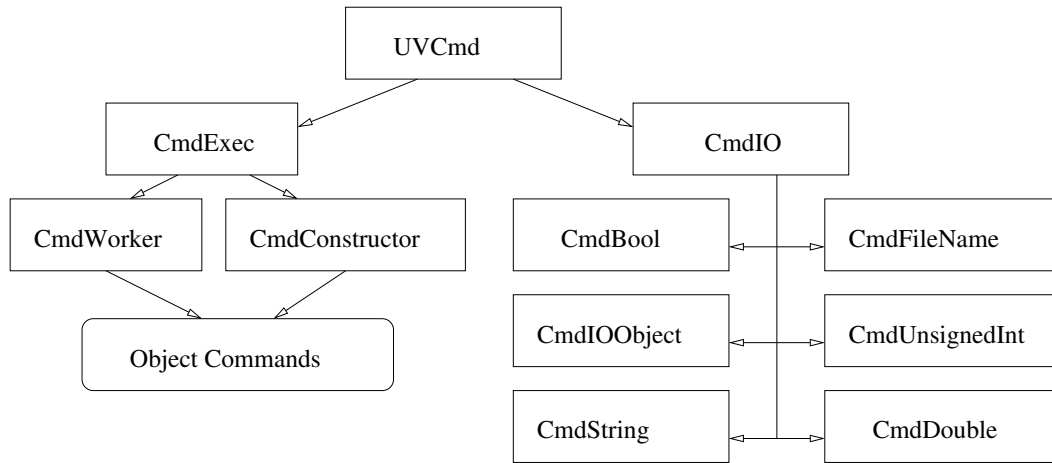


Figura B.9: Jerarquía de clases de comandos en UVAS.

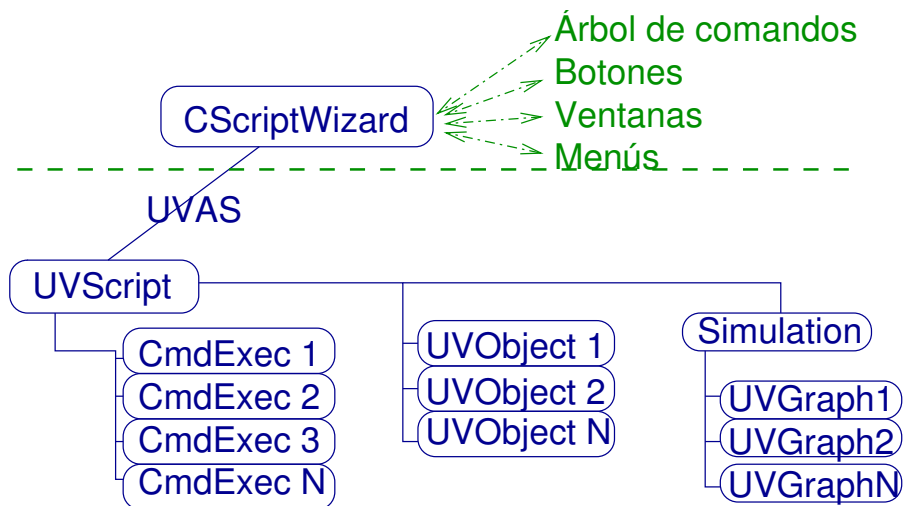


Figura B.10: Estructura interna del "Script Wizard". La clase CScriptWizard se encarga de manejar todo el interfaz gráfico (botones, ventanas...). Esta clase, mediante la utilización del entorno UVAS a modo de pasarela, consigue el UVScript actual, que contiene los comandos que deben ejecutarse, así como sus objetos y gráficos asociados.

```

#define _UVCMD_H_
#include <string>
#include <vector>
#include <set>
class CmdExec;
class UVCmd
{
public:
    UVCmd() { bReadOnly = false; }
    virtual ~UVCmd();
    virtual UVCmd * Clone() const = 0;
    virtual std::string Type() const = 0;
    virtual std::string Description() const = 0;
    unsigned int NCmd() const { return stlCmd.size(); }
    const std::vector<UVCmd *> * GetCmds() const { return &stlCmd; }
    UVCmd * GetCmd(unsigned int) const;
    void SetReadOnly();
    bool IsReadOnly() const { return bReadOnly; }
    static void StaticSaveTo(const CmdExec *, std::ostream &, CmdInfoSave &);
    static CmdExec * StaticLoadFrom(std::istream &);
    static const char * OBJECTLABEL;

protected:
    virtual void SaveTo(std::ostream &, CmdInfoSave &) const;
    virtual UVCmd * LoadFrom(std::istream &);

    void RegSubCommand(const UVCmd &);
    std::vector<UVCmd *> stlCmd;
    std::vector<UVCmd *> stlMine; //For correct deleting.
    bool bReadOnly;
};
#endif

```

En el anterior código se pueden ver las funciones que se encargan de realizar las operaciones anteriormente reseñadas.

► B.5.3

Comandos de entrada de parámetros: La clase CmdIO

La clase CmdIO (ver figura B.11) está derivada de la clase UVCmd y su objetivo es obtener los parámetros del usuario y pasárselos a las distintas funciones. Esta clase ha sido creada para satisfacer la demanda de recursividad que todo el algoritmo requiere a través de UVCmd en la forma en que ha sido planteado.

Lo que la clase CmdIO añade a UVCmd son, básicamente, unas funciones para poner (set) y obtener (get) los parámetros, que se pasan como cadenas de texto. Un extracto del código CmdIO sería:

```

// CmdIO.h: interface for the CmdIO class.
//
#if !defined(_CMDIO_H_)
#define _CMDIO_H_
#include "UVCmd.h"
#include <string>
#include <deque>
class UVScript;

```

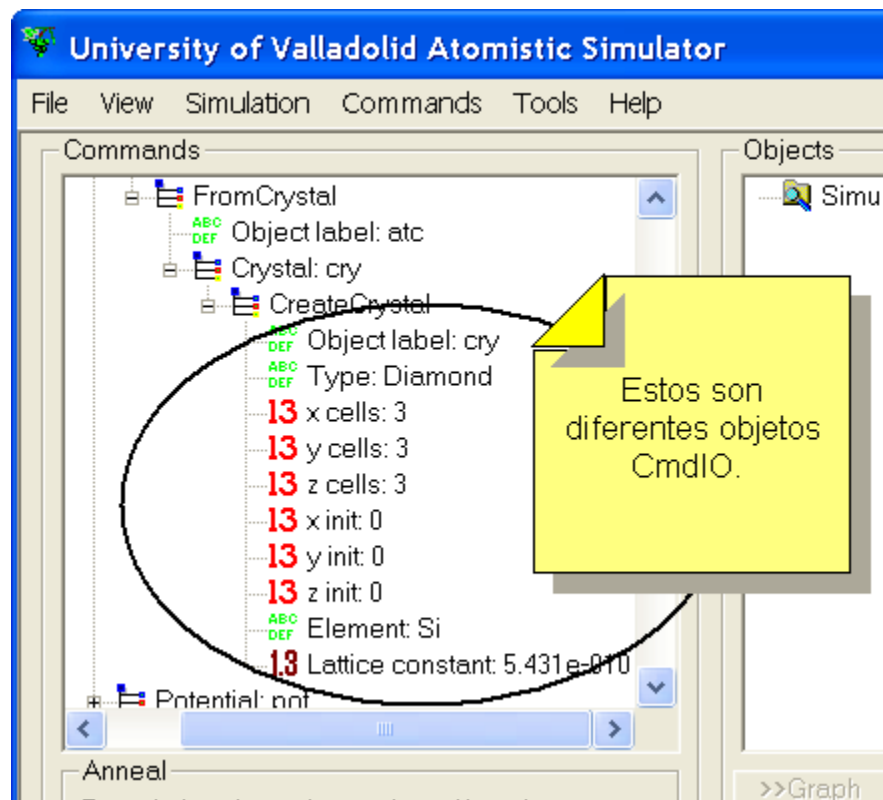


Figura B.11: Algunas clases de tipo CmdIO funcionando.

```

class CmdIO : public UVCmd
{
public:
    typedef std::deque<std::string> stringQue;
    CmdIO(const std::string &szT) : szParName(szT) {}
    virtual ~CmdIO() {};
    virtual std::string DataOut() const = 0;
    virtual void DataIn(const std::string &) = 0;
    std::string GetParName() const { return szParName; }
    virtual void SaveTo(std::ostream &, CmdInfoSave &) const;
    virtual UVCmd * LoadFrom(std::istream &);
    virtual stringQue GetLastValues() const=0;
    virtual std::string ObtainData() const = 0;
protected:
    const std::string szParName;
    void UpdateLastValues(const std::string &, stringQue &stl) const;
};

#endif // !defined(_CMDIO_H_)

```

Parámetros concretos derivados de CmdIO

Una vez que se tienen estructuras generales para establecer y obtener parámetros, es el momento de hablar de las clases concretas que han sido implementadas para cada tipo concreto de parámetro. Éstas son:

CmdIOBool para argumentos booleanos (verdadero o falso). Figura B.12.

CmdIODouble para argumentos numéricos con punto decimal. Figura B.13.

CmdIOFileName para nombres de ficheros. Figura B.15.

CmdIOObject para argumentos que a su vez son otros objetos o módulos de simulación (UVOjects). Por ejemplo, para funciones de conversión entre objetos se necesita el objeto del cual extraer la información. Figura B.16.

CmdIOString para cadenas de caracteres. Figura B.17.

CmdIOUnsignedInt para números enteros positivos. Figura B.14.

A modo de ejemplo para clarificar la implementación, se lista el código de CmdIOBool:

```

// CmdSetBool.h: interface for the ParBool class.
//
#if !defined(_CMDIOBOOL_H_)
#define _CMDIOBOOL_H_

#include "CmdIO.h"

class CmdIOBool : public CmdIO
{
public:
    CmdIOBool(const std::string &szT, bool b) : CmdIO(szT) { bValue = b; }
    virtual ~CmdIOBool() {};
    bool Get() const { return bValue; }
};

```

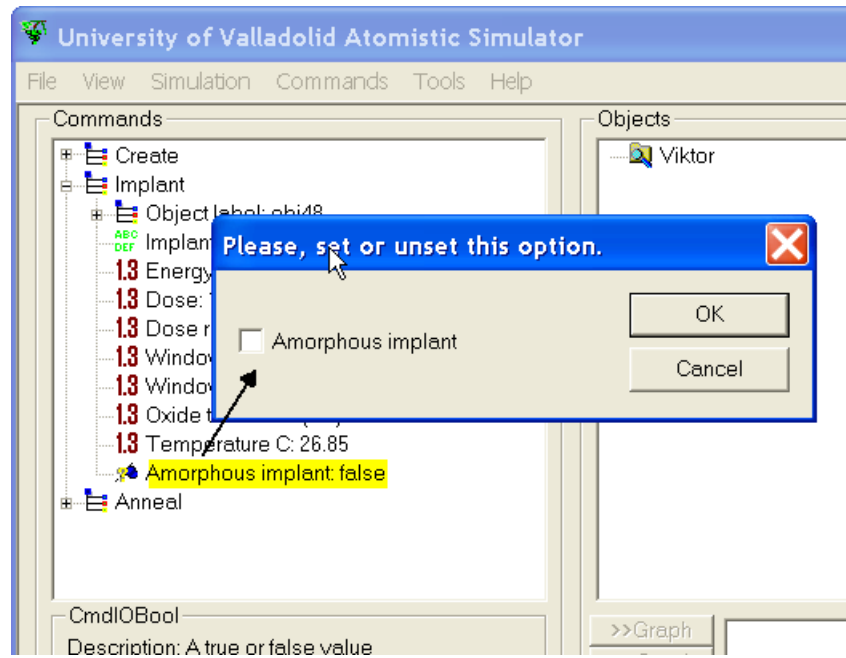


Figura B.12: Diálogo asociado y CmdIOBool.

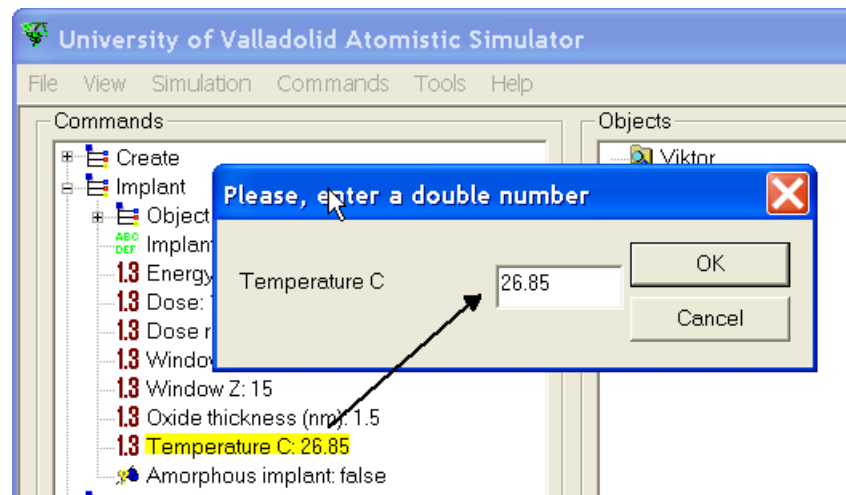


Figura B.13: Diálogo asociado y CmdIODouble.

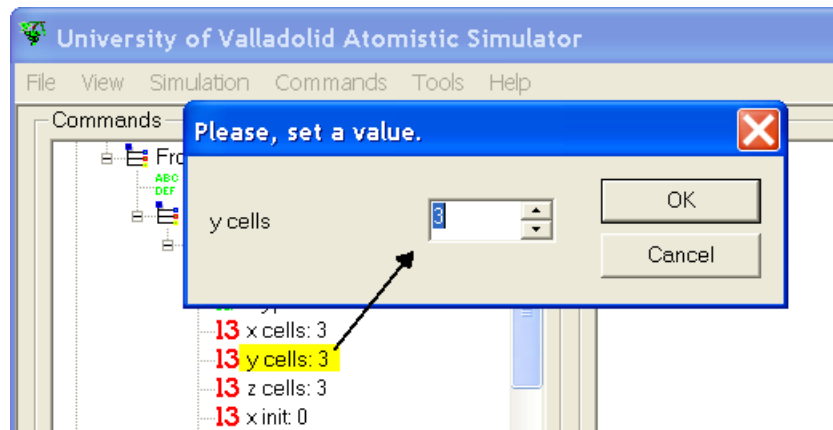


Figura B.14: Diálogo asociado y CmdIOUnsignedInt.

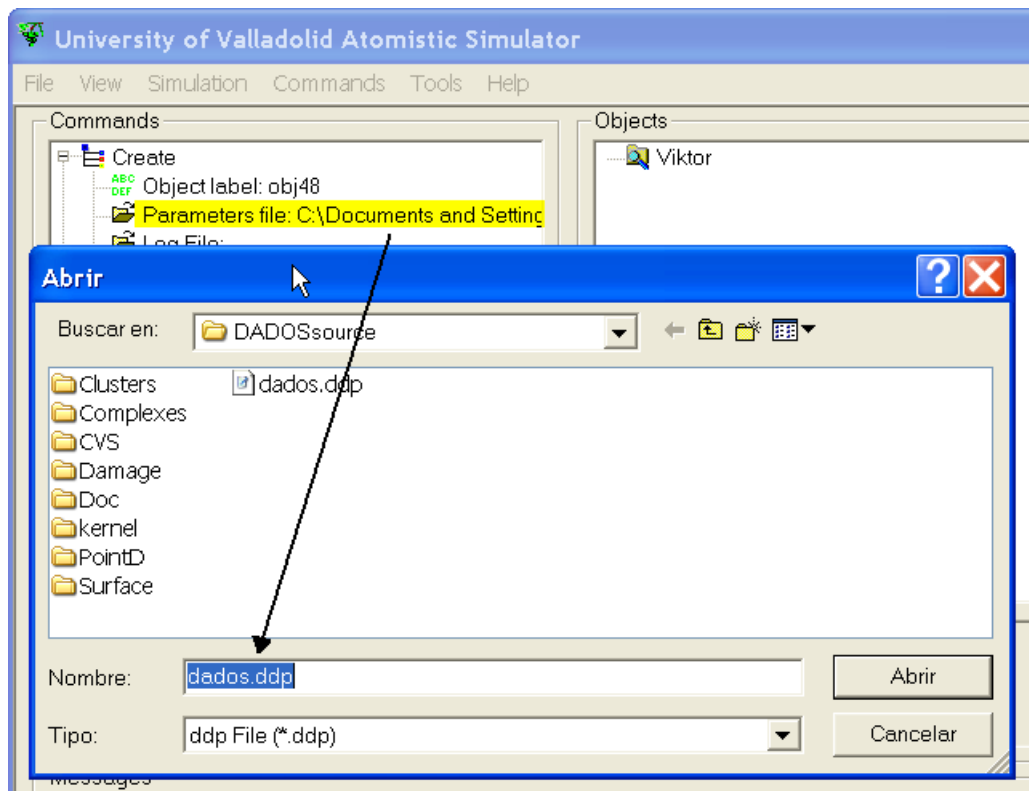


Figura B.15: Diálogo asociado y CmdIOFileName.

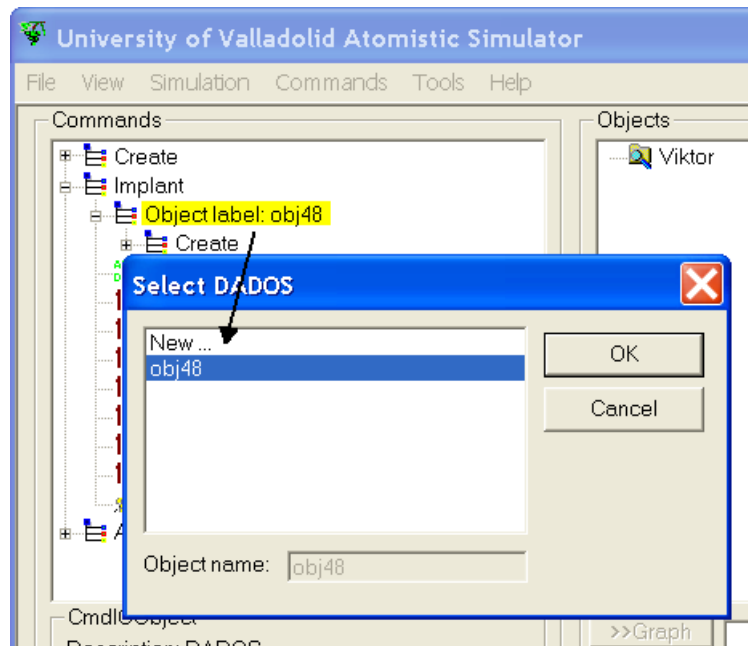


Figura B.16: Diálogo asociado y CmdIOObject.

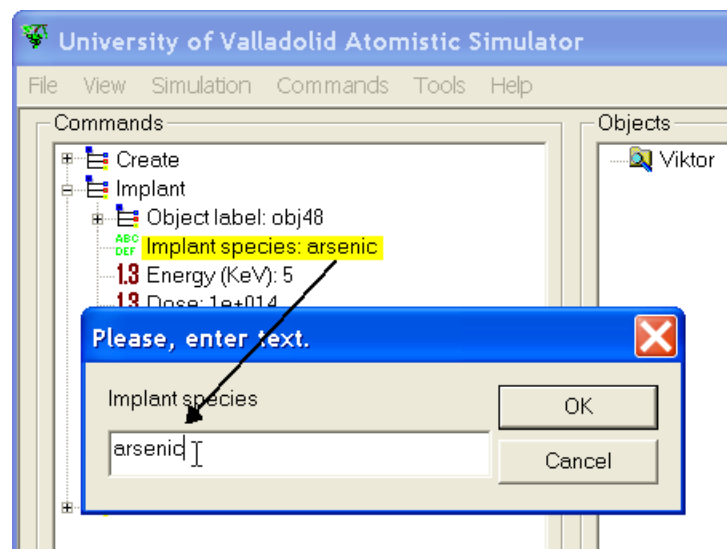


Figura B.17: Diálogo asociado y CmdIOString.

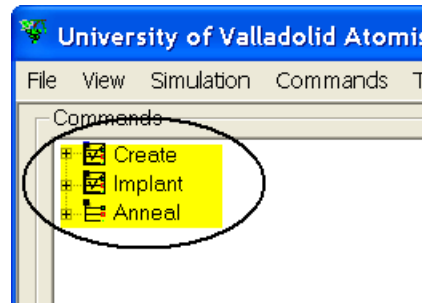


Figura B.18: Un “script” de ejecución con tres comandos.

```

void Set(bool bV) { bValue = bV; }

UVCmd * Clone() const { return new CmdIOBool(szParName, bValue); }

stringQue GetLastValues() const
{ stringQue a;a.push_front("true");a.push_back("false"); return a; }
void DataIn(const std::string &);
std::string DataOut() const;
std::string Description() const { return "A_true_or_false_value"; }
std::string Type() const { return "CmdIOBool"; }
virtual std::string ObtainData() const;

private:
    bool bValue;
};
#endif

```

▷ B.5.4

Comandos de ejecución: CmdExec

Los comandos `CmdExec` serían los comandos propiamente dichos, es decir, una interfaz para las funciones que hay que ejecutar y no sólo la forma de pasarles los parámetros, como resultaban los anteriores. En la figura B.18 se puede ver un ejemplo de alguno de estos comandos.

`CmdExec` añade a `UVCmd` las estructuras necesarias para almacenar los parámetros requeridos y acceder a ellos fácilmente, así como información sobre el objeto o módulo de simulación en el cual van a operar.

Un extracto del código de `CmdExec` sería:

```

// CmdExec.h: interface for the CmdExec class.
//
#if !defined(_CMDEXEC_H_)
#define _CMDEXEC_H_

#include "UVCmd.h"
#include "CmdIOBool.h"
#include "CmdIOString.h"
#include "CmdIOFileName.h"

```

```

#include "CmdIOUnsignedInt.h"
#include "CmdIODouble.h"
#include "CmdIOObject.h"
#include <cassert>

class UVObject;

class CmdExec : public UVCmd
{
public:
    CmdExec(const UVObject *p) : pProto(p) { assert(p); }
    virtual ~CmdExec() {}
    const UVObject * GetProto() const { return pProto; }
    std::string GetObjLabel() const
        { return GetCmdIO(OBJECTLABEL)->DataOut(); }
    void SetObjLabel(const std::string &szT)
        { GetCmdIO(OBJECTLABEL)->DataIn(szT); }
    virtual void SaveTo(std::ostream &, CmdInfoSave &) const;
    virtual UVCmd * LoadFrom(std::istream &);

protected:
    const UVObject *pProto;
    //To easily obtain SubCommands
    bool GetBool(const std::string &) const;
    unsigned int GetUnsignedInt(const std::string &) const;
    double GetDouble(const std::string &) const;
    std::string GetString(const std::string &) const;
    std::string GetFileName(const std::string &) const;
    UVObject * GetUVObject(const std::string &) const;
    //etc...
private:
    CmdIO * GetCmdIO(const std::string &) const;
};

#endif // !defined(_CMDEXEC_H_)

```

A su vez dos comandos, los “constructores” y “ejecutores”, son derivados de `CmdExec`. Estos comandos derivados son los finales, a los que el usuario puede acceder a través de los menús, como se muestra en la figura B.19.

Construyendo objetos: `CmdConstructor`

El propósito de un comando de tipo `CmdConstructor` es construir un `UVObject` desde el principio. Básicamente, `CmdConstructor` se limita a añadir una función `Work` cuya misión es devolver el nuevo objeto construido.

```

class CmdConstructor : public CmdExec
{
public:
    CmdConstructor(const UVObject *p);
    virtual ~CmdConstructor();
    virtual UVObject * Work() const = 0;
    virtual UVCmd * LoadFrom(std::istream &);
private:
    bool bInSet;
};

```

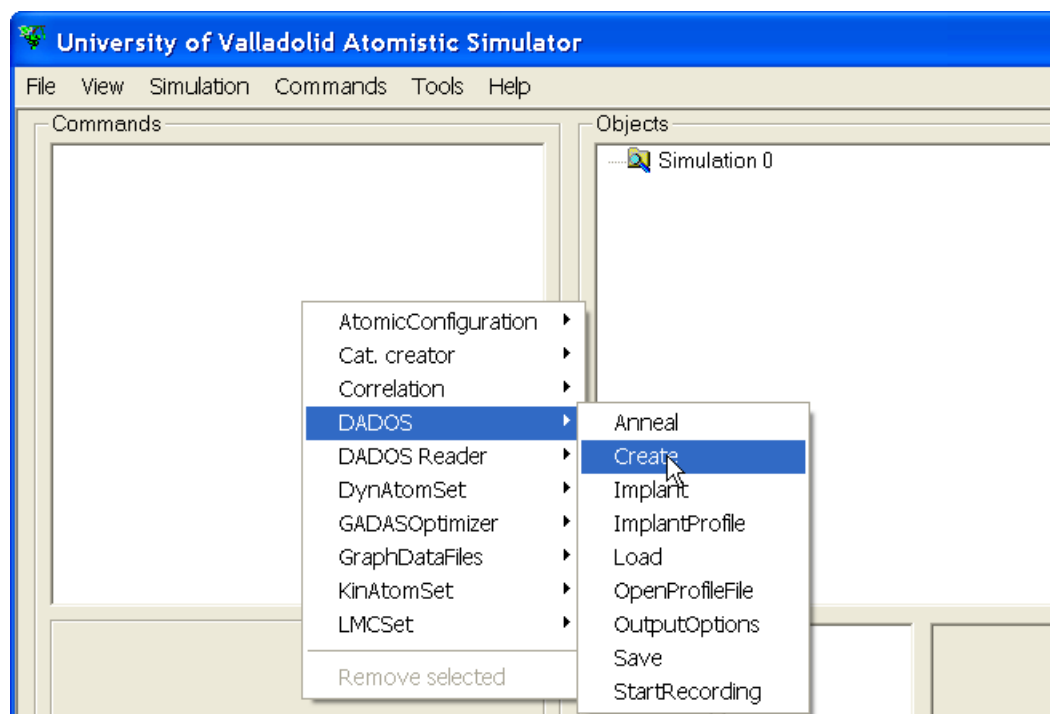


Figura B.19: Una vez que se ha empezado una simulación vacía (en nuestro caso llamada “Simulation 0”), se puede acceder a los distintos comandos de la simulación a través del menú del programa.

Trabajando sobre objetos ya construidos: CmdWorker

Una vez que un `UVObject` ha sido creado, las clases de tipo `CmdWorker` son capaces de llamar a sus funciones. De esta manera los `CmdWorker` constituyen los órdenes que el usuario desea realizar con los objetos que va creando.

El código de un `CmdWorker` es bastante simple, ya que se limita a añadir una función `Work` que, una vez que conoce el objeto sobre el que operar, invoca a una determinada función de éste.

```
class CmdWorker : public CmdExec
{
public:
    CmdWorker(const UVObject *p);
    virtual ~CmdWorker() {}
    virtual void Work(UVObject *) const = 0;
};
```

► B.5.5

Una clase que contiene a todos los comandos: UVScript

Las estructuras previamente explicadas son gestionadas por una clase llamada `UVScript`. El objetivo de esta clase es almacenar las diferentes “líneas” (de la lista de comandos a ejecutar, ver figura B.18), de la lista completa, así como borrar o insertar nuevas líneas, grabar y guardar la lista en disco y, claro está, ejecutar la lista de comandos. UVAS utiliza distintos hilos (threads) para las diferentes simulaciones, y un hilo extra para la interfaz gráfica de la aplicación que constituye lo que ve el usuario.

La definición de `UVScript` se muestra a continuación:

```
// UVScript.h: interface for the UVScript class.
//
#if !defined(_UVSCRIPT_H_)
#define _UVSCRIPT_H_

#include <vector>
#include <string>
#include "UVObject.h"
#include <map>
class CmdConstructor;

class UVScript : public std::vector<UVCmd *>
{
public:
    //Defined types
    typedef std::map<std::string , UVObject *> String2Object;
    UVScript();
    ~UVScript();
    bool RemoveScriptLine();
    void InsertScriptLine(int , UVCmd *);
    unsigned int GetCurrentLine() const { return nCurrentLine; }
    void SetCurrentLine(unsigned int nLine);
    static void ExecuteScript(UVThread *);
    const String2Object * GetpMap() { return &stlObjects; }
    //To send uptdate Message
    void SetCurrentSimulation(const Simulation * pS) { pSim = pS; }
```

```

void SaveTo(std::ostream &) const;
static void StaticLoadFrom(std::istream &, Simulation *);
bool IsOrphan(const CmdConstructor *, const UVCmd * = 0) const;
std::set<CmdConstructor *> stlConstructors;
private:
void Executor(UVCmd *, UVThread *);
String2Object stlObjects;
const Simulation * pSim;
unsigned int nCurrentLine;
};
#endif

```

La lista de comandos (el “script”) es ejecutada desde una función denominada `ExecuteScript` que consigue la línea actual de la lista y su clase `UVCmd` asociada, llamando entonces a otra función: `Executor`. `Executor` chequeará el tipo de comando del que se trata (un constructor, un comando de trabajo o un parámetro) y procederá según su tipo. En el caso en que se encuentre con un comando que no sea parámetro, llamará a la función `Work` asociada al objeto.

```

//Setting the correct UVObject of the map.
CmdWorker *pWorker;
CmdConstructor *pConstructor;
CmdIOObject *pCmdObj;

if ( (pWorker = dynamic_cast<CmdWorker *>(pPpalCom)) != 0)
{
    if (stlObjects.find(pWorker->GetObjLabel()) == stlObjects.end())
        throw UVErrors::InvalidObject(pWorker->GetObjLabel());
    UVAS::get()->updateProgressPushTag(reinterpret_cast<int>(stlObjects[
        pWorker->GetObjLabel()]), std::string("Working:~")+pWorker->Type());
    pWorker->Work(stlObjects[pWorker->GetObjLabel()]);
    UVAS::get()->updateProgressPopTag();
}

```

▷ B.6

Los datos de salida: DataBrowser

▷ B.6.1

Introducción

El objetivo del “DataBrowser” consiste en proveer una manera simple de acceder a todos los datos generados por diferentes simulaciones para así poder revisar o representar dichos datos convenientemente.

Básicamente, el “DataBrowser” almacena información sobre todas las variables y diferentes datos que pudieran ser de interés para el usuario. En dicha información se explicitan las localizaciones en memoria y el tipo de información generada, así como todo lo necesario para poder acceder posteriormente a los datos mencionados.

Esa información es almacenada en estructuras específicas. `UVData` contiene

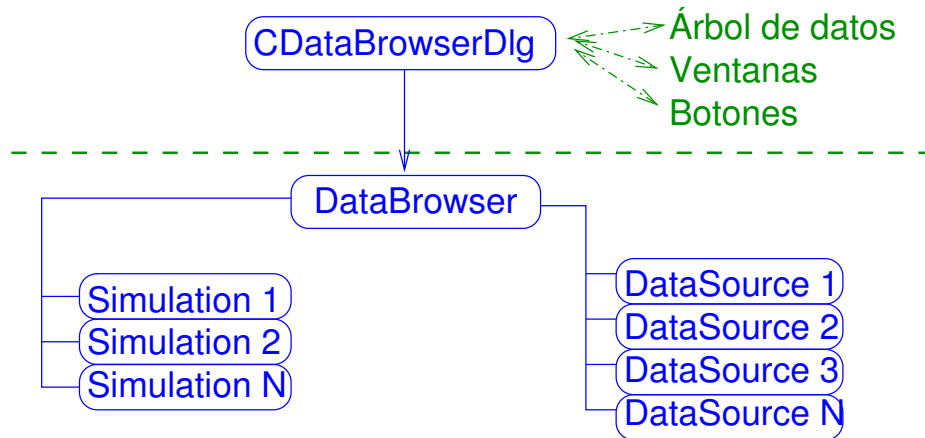


Figura B.20: Estructura interna del “Data Browser”. El objeto `CDataBrowserDlg` maneja los aspectos relacionados con el entorno gráfico, y está derivado de `DataBrowser`, que contiene los algoritmos necesarios para manejar los datos “en crudo”. El objeto `DataBrowser` contiene todas las simulaciones presentes en la sesión, así como todas las fuentes para obtener cada dato posible generado durante la simulación.

todas esas estructuras, así como los algoritmos necesarios para extraer de ellas la información en un formato útil.

Posteriormente, los objetos concretos `DataBrowser` y `CDataBrowserDlg` se encargan de obtener la información almacenada y dar acceso a ella a través de la interfaz gráfica.

Las estructuras internas de `DataBrowser` y `CDataBrowserDlg` se encuentra representadas en la figura B.20. En principio `CDataBrowserDlg` gestiona los aspectos gráficos, y está derivada de `DataBrowser` que contiene los algoritmos y rutinas de índole más general para acceder a los datos.

► B.6.2

Clases generales para manipular los datos: `UVData`

`UVData` es el nombre de una clase genérica que a su vez contiene otras muchas clases. Todas estas otras clases se encargan de implementar las estructuras diferentes que almacenarán los datos. Las clases anidadas en `UVData` se pueden organizar en dos grandes grupos. El primero está relacionado con las operaciones genéricas que se realizan sobre los datos, y el segundo con la implementación de tipos muy concretos de estos datos, como enteros, números con punto decimal, cadenas de caracteres, etc.

Las clases genéricas se han llamado `DataLine`, `DataSource`, `DataSourceLine` y `UVDataStruct`, los nombres de los tipos particulares de datos son `UVBool`, `UVChar`, `UVDouble`, `UVFloat`, `UVLongInt`, `UVUnsignedInt` y `UVVoidPointer`.

Por último `UVData` contiene otras clases muy pequeñas que se usan como

señales de error, son `IndexMismatch`, `SizeMismatch` y `TypeMismatch`.

Un almacén de información básica: `DataLine`

El objeto `DataLine` se utiliza para almacenar información básica sobre las variables y otros datos que serán usados en el `DataBrowser`. Su estructura es:

```
class DataLine
{
public:
    DataLine(std::string , const void * , const void * , TYPE,
             size_t , const UVDataStruct *);
    DataLine(std::string , const void * , const void * , TYPE,
             size_t , const UVDataStruct * , const Tag2Index &tags);

    std::string szLabel;
    UVPointer Offset;
    TYPE Type;
    size_t SizeOf;
    const UVDataStruct * pDataStruct;
    Tag2Index vectorTags;
};
```

Por tanto, una `DataLine` contiene un cadena de caracteres que identifica el nombre del dato, así como **la distancia en bytes del dato respecto al origen de la estructura que la contiene**, su tipo (entero, booleano...), tamaño y, por último, un puntero a una estructura llamada `UVDataStruct` que será comentada a continuación, pero que básicamente permite establecer una recursividad entre los diferentes datos.

Definiendo un contenedor de los objetos `DataLine`: `UVDataStruct`

Las `DataLines` guardan la información que se refiere a una variable o tipo de dato sencillo, frente a `UVDataStruct` que almacena información sobre conjuntos (estructuras) que pueden albergar varias de estas variables simples de distintos tipos. Cada elemento de `UVDataStruct` resultará ser una `UVDataLine` para cada variable involucrada en la estructura representada.

Debido a que, a veces, la variable de un objeto puede ser otro objeto por si mismo, (o estructura o vector), se necesita una forma de poder acceder a estas meta-estructuras y a su información. Es por ello que cada `DataLine` contiene el puntero de una `UVDataStruct`.

El listado de `UVDataStruct` es bastante simple:

```
class UVDataStruct : public std::vector<DataLine>
{
public:
    virtual ~UVDataStruct();
};
```

Así pues, `UVDataStruct` se limita a ser un vector (un array) de `DataLines`.

Procesando un poco la información: DataSourceLine

`DataLine` y `UVDataStruct` contienen la información “en crudo” que se refiere a las variables, estructuras, vectores, etc... de los datos del UVAS. `DataSourceLine` contiene la misma información, pero más procesada:

```
struct DataSourceLine
{
    DataSourceLine(UVPointer p, TYPE t,
        size_t S, int I, const std::string &szL);
    DataSourceLine(UVPointer p, TYPE t,
        size_t S, int I, const std::string &szL, const Tag2Index &vt);
    bool operator==(const DataSourceLine &ds) const;
    void SaveTo(std::ostream & os) const;
    static DataSourceLine LoadFrom(std::istream &);
    UVPointer pointer;
    TYPE Type;
    size_t SizeOf;
    int Index;
    std::string szLabel;
    Tag2Index vectorTags;
};
```

Así pues, el programa va traduciendo las distintas `DataLines` en `UVDataStruct`, usando para ello las funciones miembro `CDataBrowserDlg::UpdateTreeContents` y `CDataBrowser::ContinueTree`. Por ejemplo, cuando una `DataLine` contiene información sobre un `UVObject`, el traductor sustituye el nombre del objeto (una etiqueta), por la dirección real en memoria donde reside dicho objeto.

Un array de DataSourceLines: la clase DataSource

La clase `DataSource` es un array de `DataSourceLines` así como toda la serie de operaciones y algoritmos asociados a la obtención de los datos finales utilizando la información almacenada por `DataSource`. Su estructura interna está dibujada en la figura B.21.

Cada variable de datos accesibles a través del `DataBrowser` necesitan tener un `DataSource` asociado. En dicha estructura se almacena la forma de procesar y acceder a los valores correctos de la variable. Estos `DataSources` requieren ser bastante complejos debido a la estructura dinámica de los datos. Si simplemente se guardara la dirección en memoria de cada variable y una pequeña etiqueta para acceder a ella, esto no sería suficiente para obtener los datos más tarde. Ello es así ya que UVAS asigna la memoria de manera dinámica y almacena los punteros a dicha información por lo que, si únicamente guardáramos la dirección en un momento dado y el programa reasignara dicha memoria y la colocara en una posición diferente, la dirección guardada no sería correcta. Esto obliga a guardar todo el camino a seguir para conseguir la información, ya que por ejemplo, una variable concreta puede ser parte de un array de punteros a cierta estructura que pueden estar siendo movidos o borrados, etc., lo que obliga a guardar esta información y las operaciones y algoritmos que hacen posible acceder a ella de forma segura, reconstruyendo la dirección en la que se guarda la información.

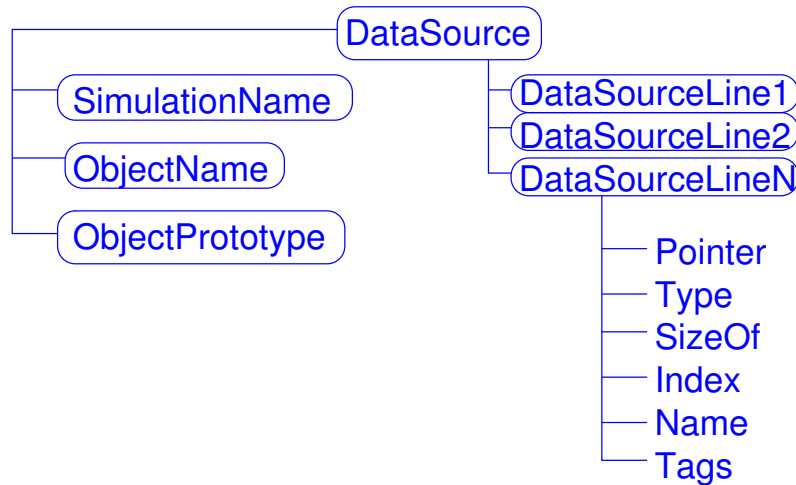


Figura B.21: Estructura interna de DataSource. DataSource contiene información sobre la simulación y el objeto del cual está almacenando datos, junto con un array de DataSourceLines, donde está la forma de acceder a dichos datos.

Con esto ya conocemos la misión del DataBrowser. Su definición de clase es bastante compleja y se lista un extracto de ella a continuación:

```

class DataSource : public std::vector<DataSourceLine>
{
private:
    //For Data begin, end and next
    mutable unsigned int CurrentIndex;
    enum DSActions { ObtainData = 0, HeadSize, NormalSize, NISize };
public:
    DataSource();
    void DataBegin() const;
    bool DataEnd() const;
    void DataNext() const;
    unsigned int DataSize() const;
    unsigned int NonIteratingSize() const;
    UVPointer GetData(DSActions = ObtainData) const;
    bool IsIterating() const;
    bool IsVectorHead() const;
    unsigned int VectorHeadSize() const;
    std::string ElementToString() const;
    double ElementToDouble() const;
    TYPE DataType() const;

    //Up and down in the series, get current index.
    int operator++();
    int operator--();
    int operator=(int);
    bool operator==(const DataSource &) const;
    int GetScrollIndex() const;
    void SetScrollIndex(int sl);
    unsigned int GetCurrentIndex() const { return CurrentIndex; }
    void SetCurrentIndex(unsigned int nl) const { CurrentIndex = nl; }

    void SaveTo(std::ostream &) const;
    void LoadFrom(std::istream &);

```

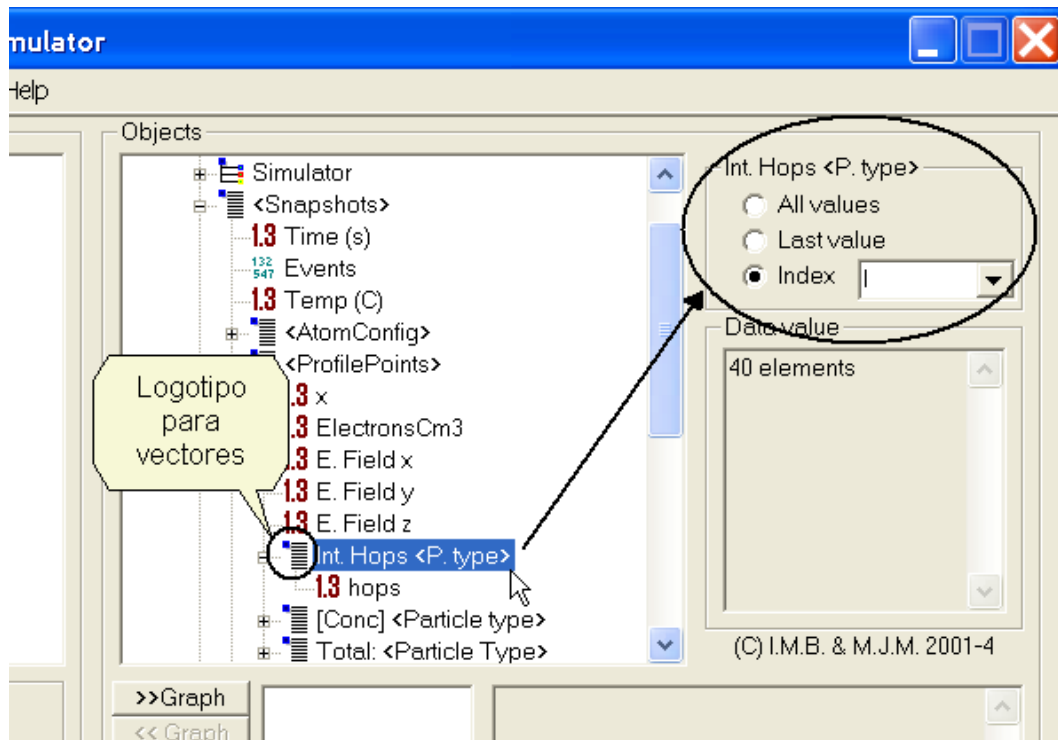


Figura B.22: El índice de un vector se puede fijar a un valor, o solicitar que se itere sobre él.

```
public:
    std::string szSimulation;
    std::string szObject;
};
```

Mucha de la complicación de esta clase es debida a que los datos a los que se quiere acceder están asociados a arrays, los cuales a su vez pueden estar anidados o contener otros arrays. Así pues, cada vez que el programa detecta un vector o array, ver figuras B.22 y B.23, hay que conocer si lo que se pretende es acceder a un elemento concreto del vector, o bien iterar sobre todos sus elementos. En el programa UVAS, cuando se anidan unos arrays dentro de otros, únicamente se permite iterar sobre uno de todos los índices, y el resto deben estar fijos.

Esta restricción sobre el índice de los vectores tiene su explicación y su sentido físico. Ello se debe a que generalmente uno de los vectores está asociado con la evolución en el tiempo, y los demás podrían ser un número de configuraciones o átomos. Así pues, una petición típica consiste en ver la evolución de un determinado átomo en el tiempo, fijando el átomo e iterando sobre el tiempo, o bien ver qué configuración presentan todos los átomos en cierto instante temporal. No obstante, ver todos los átomos evolucionando a la vez podría tener su sentido, pero ciertamente complicaría el problema mucho y, por tanto, no está implementado

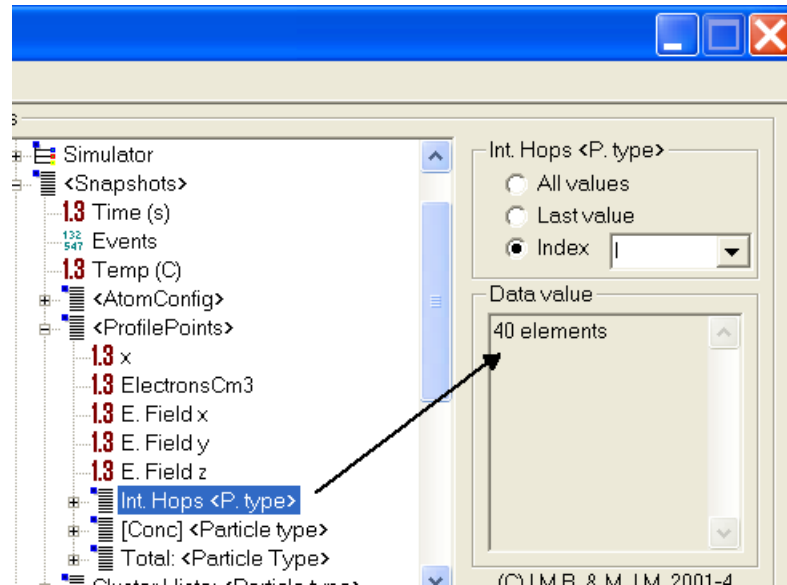


Figura B.23: Vista de cómo el “data browser” representa los vectores y sus datos asociados.

en UVAS.

Respecto a estos datos iterados, pueden ser dibujados utilizando las capacidades gráficas del programa, como en la figura B.24, o bien ser visualizados en una ventana de edición como se puede ver en la figura B.25.

Tipos concretos de datos

Se han implementado los tipos siguientes de datos:

UVBool. Encargados de la información booleana (verdadero/falso). Figura B.26.

UVChar. Encargados de un único carácter de texto.

UVDouble. Encargados de los números con punto decimal de alta precisión. Figura B.27.

UVFloat. Dedicados a los números decimales (con menos precisión/cifras significativas).

UVLongInt. Para manejar números enteros muy grandes. Figura B.28.

UVUnsignedInt. Para manejar números enteros (con menos dígitos que los anteriores). Figura B.29.

UVVoidPointer. Para manipular punteros (variables que apuntan a otras variables). Figura B.30.

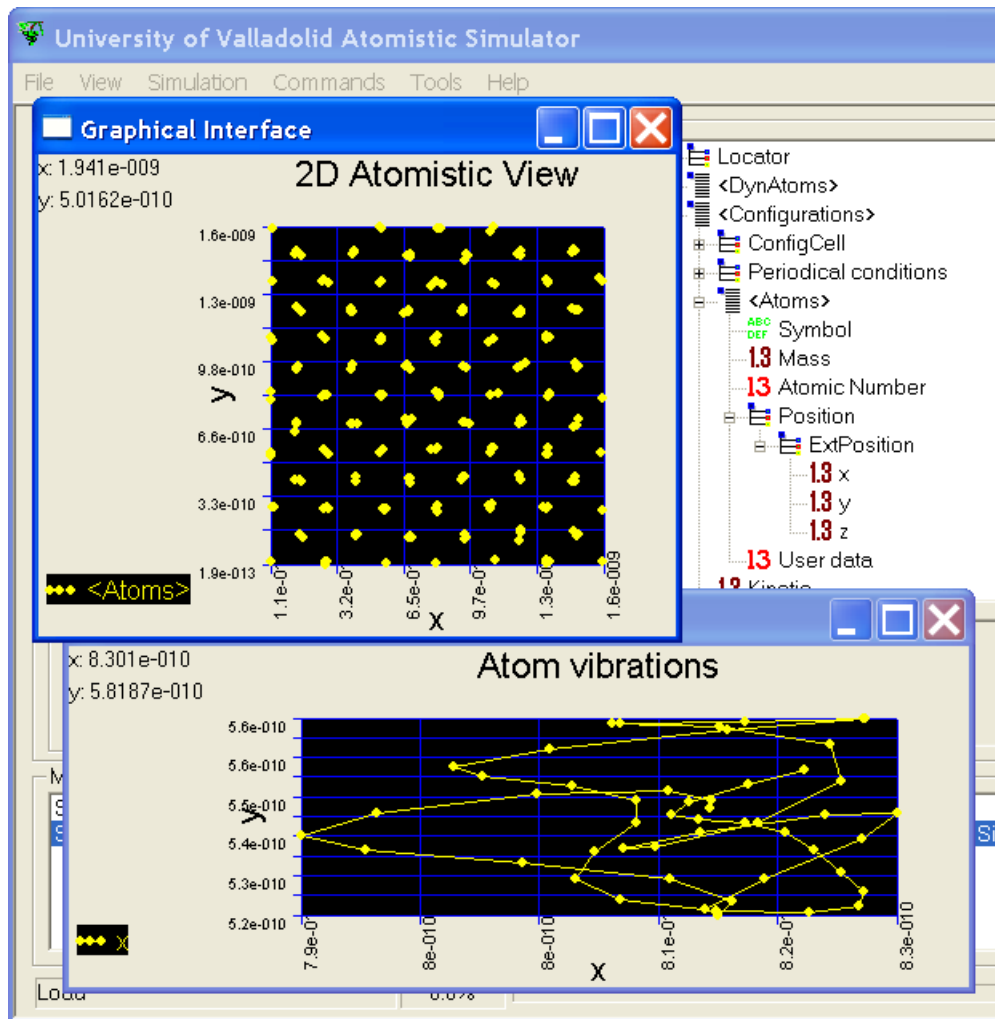


Figura B.24: Usando los “vectores” (arrays de elementos) el usuario puede iterar sobre sus diferentes índices. En la imagen, una representación está realizada iterando sobre todos los átomos en un tiempo en particular, y la otra a un determinado átomo en su evolución temporal.

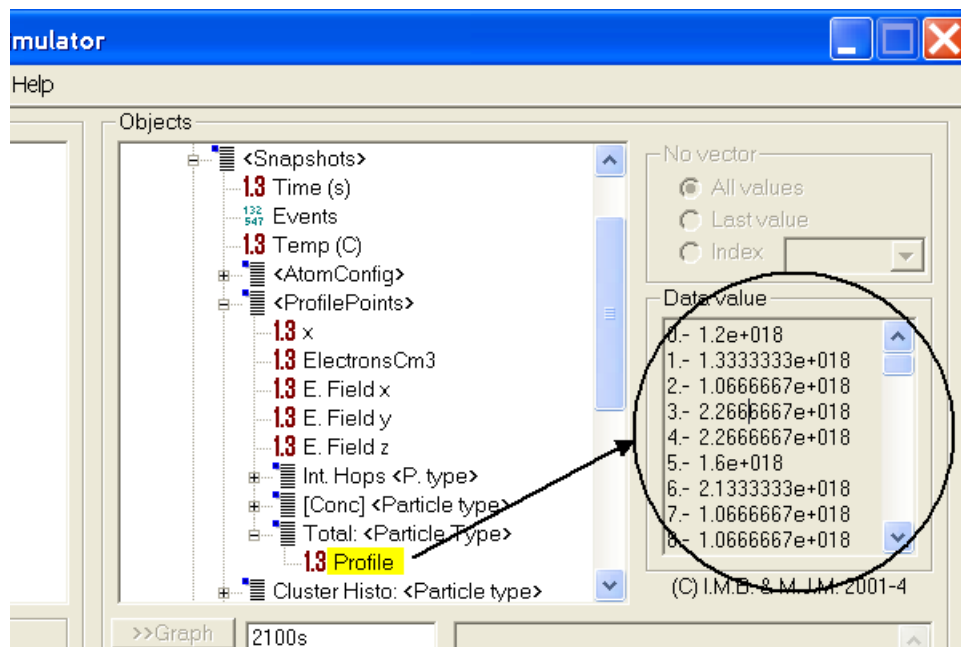


Figura B.25: Los valores de salida pueden ser representados o también visualizados como texto, lo que ayuda a que puedan ser copiados y pegados a otras aplicaciones.

UVStlString. Para manipular cadenas de caracteres. Figura B.31.

Todas las anteriores clases se definen con la “macro” siguiente:

```
#define UVDataDefineType(UVNAME, UVTYPE) class UVNAME \
{\
public:\
    UVNAME(const std::string &s="") \
    {\
        if(s.empty()) \
            szName=#UVTYPE "_value"; \
        else \
            szName=s; \
    }\
    const UVDataStruct * GetDataStruct() const \
    {\
        UVDataStruct * pds = new UVDataStruct;\
        pds->push_back(DataLine(szName, this, this, UVTYPE, 0, 0)); \
        return pds;\
    }\
private:\
    std::string szName; \
}

UVDataDefineType(UVDouble, Double);
UVDataDefineType(UVUnsignedInt, UnsignedInt);
UVDataDefineType(UVLongInt, LongInt);
UVDataDefineType(UVBool, Bool);
UVDataDefineType(UVFloat, Float);
UVDataDefineType(UVStlString, StlString);
UVDataDefineType(UVVoidPointer, Pointer);
UVDataDefineType(UVChar, Char);
```

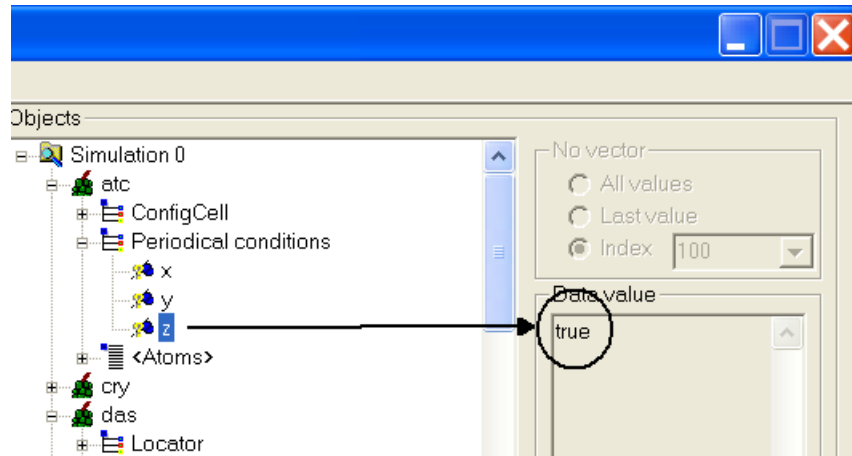


Figura B.26: Vista de un objeto UVBool funcionando: su icono, nombre y el dato que contenía.

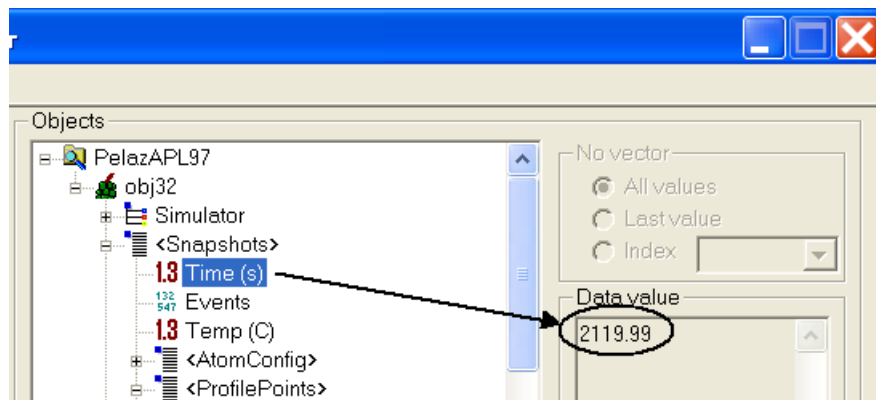


Figura B.27: Vista de un objeto UVDouble funcionando: su icono, nombre y el dato que contenía.

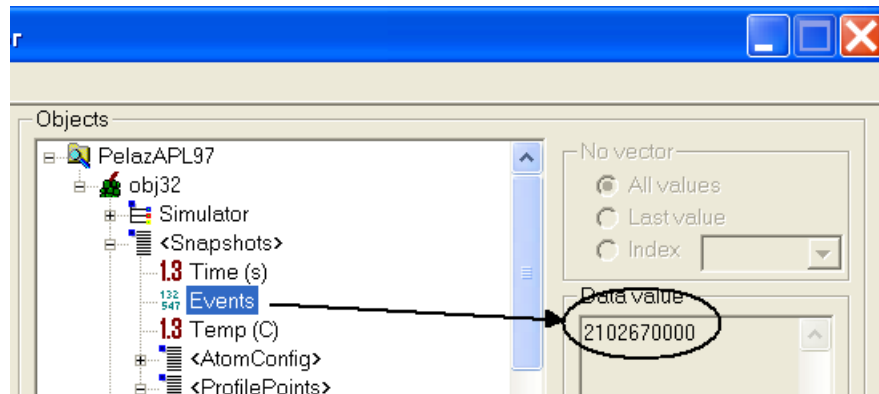


Figura B.28: Vista de un objeto UVLongInt funcionando: su icono, nombre y el dato que contenía.

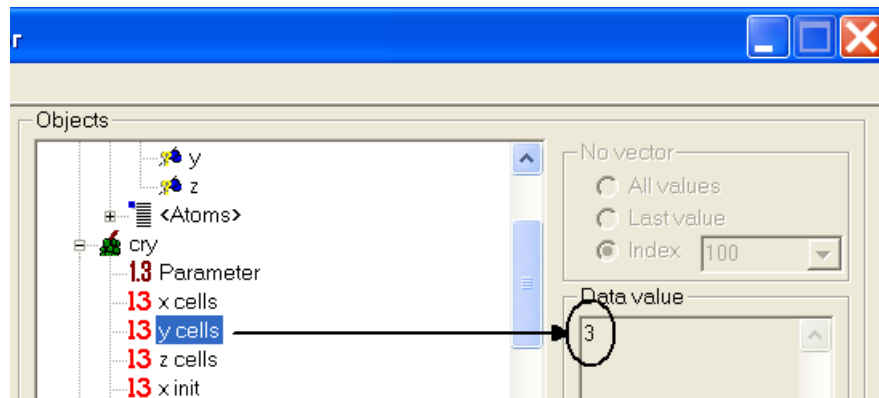


Figura B.29: Vista de un objeto UVUnsignedInt funcionando: su icono, nombre y el dato que contenía.

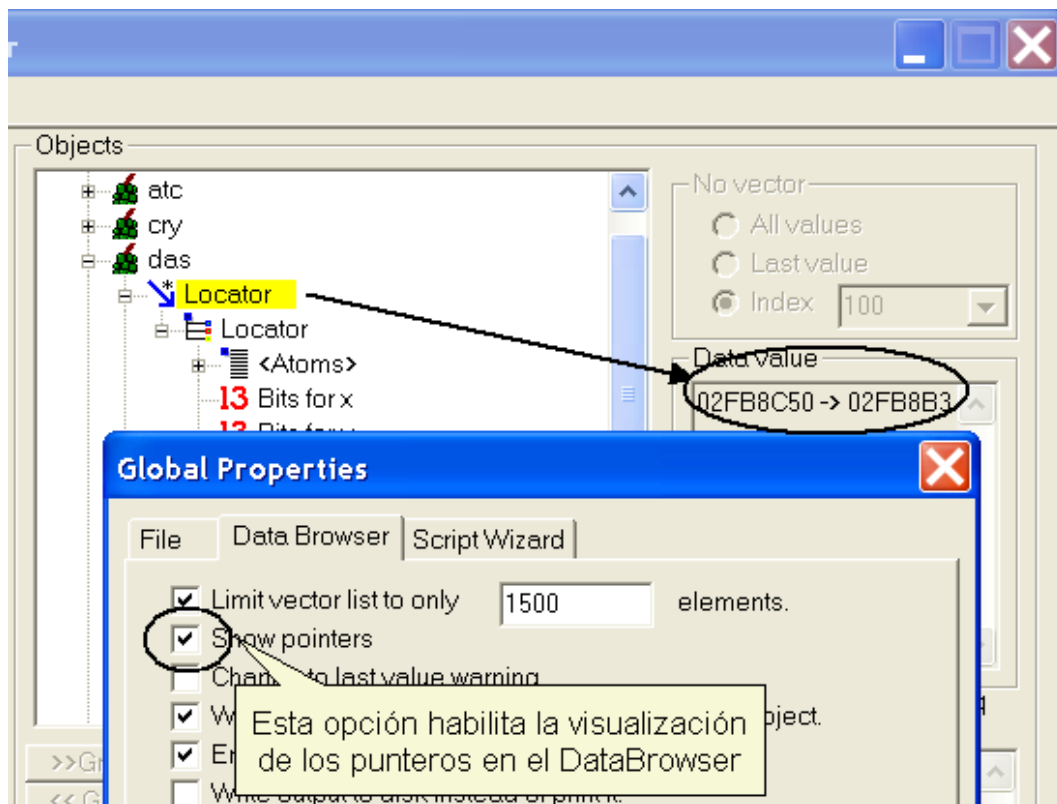


Figura B.30: Vista de un objeto UVVoidPointer funcionando: su icono, nombre y dirección a la que apuntaba.

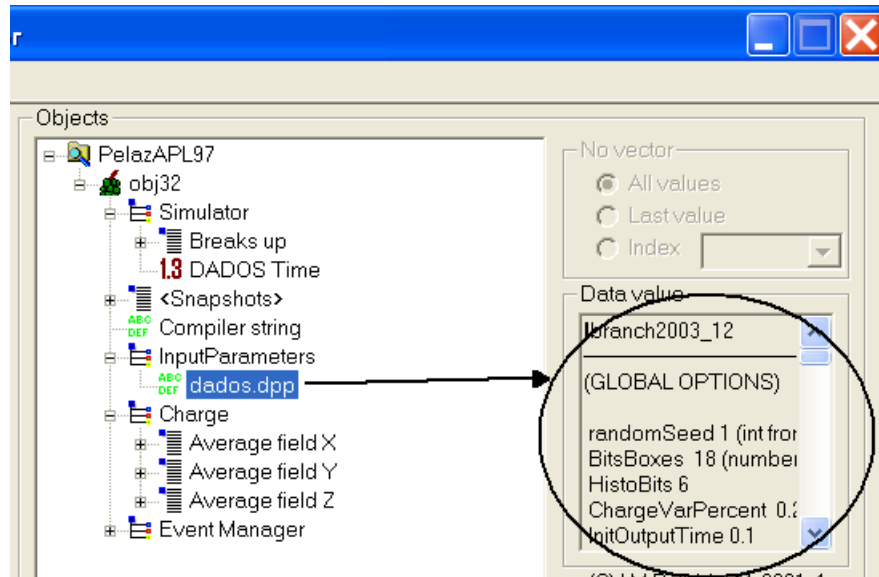


Figura B.31: Vista de un objeto UVString funcionando: su icono, nombre y cadena a la que apuntaba.

El objetivo de esta definición es proveer información que se pueda utilizar en definiciones de arrays o vectores. Debido a la forma en que UVAS ha sido programado, cuando el programador especifica la estructura adecuada para que el objeto `DataBrowserDlg` pueda mostrar sus datos asociados y cuando dicha estructura es un vector, se requiere definir una estructura anidada que se encargue de informar sobre la forma que adoptan los elementos del vector. En el caso de que dichos elementos no sean otros objetos o estructuras complejas y se traten de simples datos normales, la información ya está lista y construida a través de los anteriormente definidos `UVDouble`, `UVBool`, etc.

El "data browser" acepta otros tipos genéricos de datos (`UVClass`, figura B.33), y punteros a `UVObjects` (figura B.32), pero estos objetos fueron implementados independientemente, sin usar la anterior macro.

Señales de error

La clase `UVData` define tres tipos de señales de error (que son lanzadas como excepciones):

```
class TypeMismatch {};
class SizeMismatch {};
class IndexMismatch
{
public:
    IndexMismatch(int s, int n) { srs = s; newSrs = n; }
    int srs;
    int newSrs;
};
```

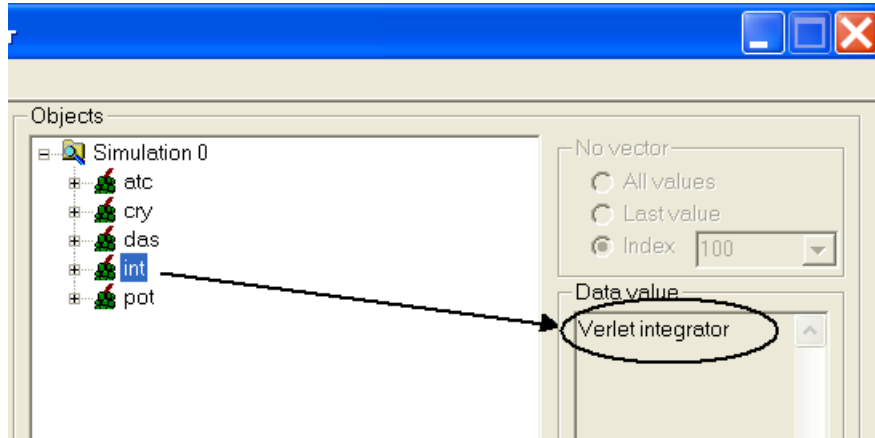


Figura B.32: Vista del acceso a un UVObject funcionando: su icono, nombre y datos a los que se accede.

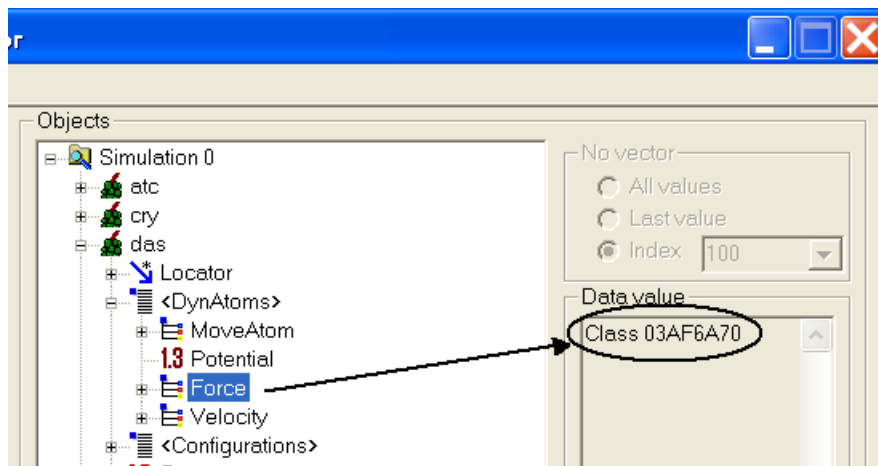


Figura B.33: Vista del acceso a una UVClass funcionando: su icono, nombre y datos a los que se accede.

TypeMismatch, que se utiliza cuando algún algoritmo recibe un tipo de `DataSource` que es de un tipo de dato diferente al que esperaba.

SizeMismatch, es utilizada cuando, al asignar datos a un gráfico para ser dibujados, el número de datos en el eje x es diferente del que hay en y .

IndexMismatch, usada si se intenta agrupar datos con diferentes índices. Por ejemplo si en una misma gráfica pretendiéramos dibujar las x en un tiempo t_1 con las y de un tiempo t_2 .

▷ B.7

Gestionando las gráficas

▷ B.7.1

Introducción

UVAS representa los siguientes tipos de gráficas:

- Gráficas en dos dimensiones, figura B.36. Presentan ejes x e y , ejes que pueden ser escalados de forma independiente. A su vez el usuario puede representar distintos tipos de puntos, o bien unir los puntos entre sí con líneas. Las representaciones 2D están especialmente indicadas para dibujar, por ejemplo, concentraciones frente al tiempo o la profundidad, pero en principio UVAS es capaz de dibujar cualquier dato en los ejes x e y .
- Gráficas en tres dimensiones, figura B.37. Presentan tres ejes, que el usuario puede rotar y desplazar con los botones izquierdo y derecho del ratón, respectivamente. Generalmente este modo de representación es usado para mostrar imágenes de configuración atomística (posiciones de átomos), aunque nuevamente, se puede representar cualquier dato que tenga tres dimensiones.

El color de los puntos o líneas representadas se puede escoger de una lista (color fijo), o bien se puede hacer depender de un campo extra de los datos representados (con lo que se añadiría una “dimensión” más a la representación). Para ello se asocia el dato en cuestión al campo de color y se deja al “data browser” gestionar todo. Utilizando esta técnica es posible generar mapas bidimensionales de color, como el de la figura B.34.

Además UVAS define ciertos atajos para la visualización de configuraciones atomísticas, que permite representar este tipo de datos con un sólo click de ratón y añadir datos extra que son mostrados como pequeñas etiquetas cuando el ratón se posa sobre los puntos representados.

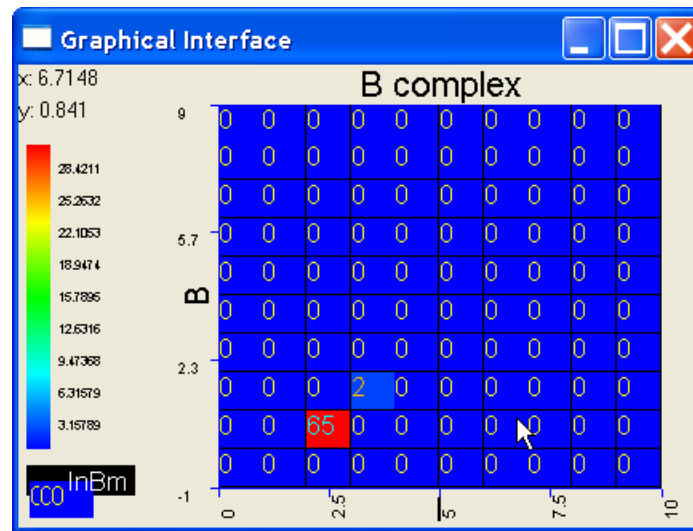


Figura B.34: Un mapa de color es un gráfico en dos dimensiones, donde el color asignado a cada celda es otra variable. De esta manera los mapas de colores ofrecen la posibilidad de representar información tridimensional en una representación bidimensional pura.

La política de implementación de los gráficos se muestra en la figura B.35. Se han creado dos grandes bloques para gráficos de 3 y 2 dimensiones. Ambos tienen una interfaz, que depende del sistema operativo usado, que se encarga de las ventanas, el ratón y los menús, junto con un código estándar C++, y los algoritmos necesarios que son añadidos al interfaz por derivación. De esta manera, las clases que no dependen de la interfaz gráfica, son derivadas de la misma clase `UVGraph`. Por último, esta clase `UVGraph` está derivada (a la vez que contiene otras) de `UVDrawRgn`, que es básicamente una clase para gestionar un área rectangular en la que se pueda dibujar, y que permita ser redimensionada y seleccionada con el ratón.

► B.7.2

Gráficos en dos dimensiones: `UVGraph2D`

`UVGraph2D` implementa todos los algoritmos y datos que se necesitan para mostrar información en una gráfica bidimensional (ver figura B.36). Sus competencias fundamentales son:

- Definir las acciones a seguir cuando las gráficas sean seleccionadas, movidas o deban ser actualizadas.
- Implementar cómo se leerá y cargará el objeto desde el disco.
- Dar información, a quien lo pudiera solicitar, sobre el título y los ejes.

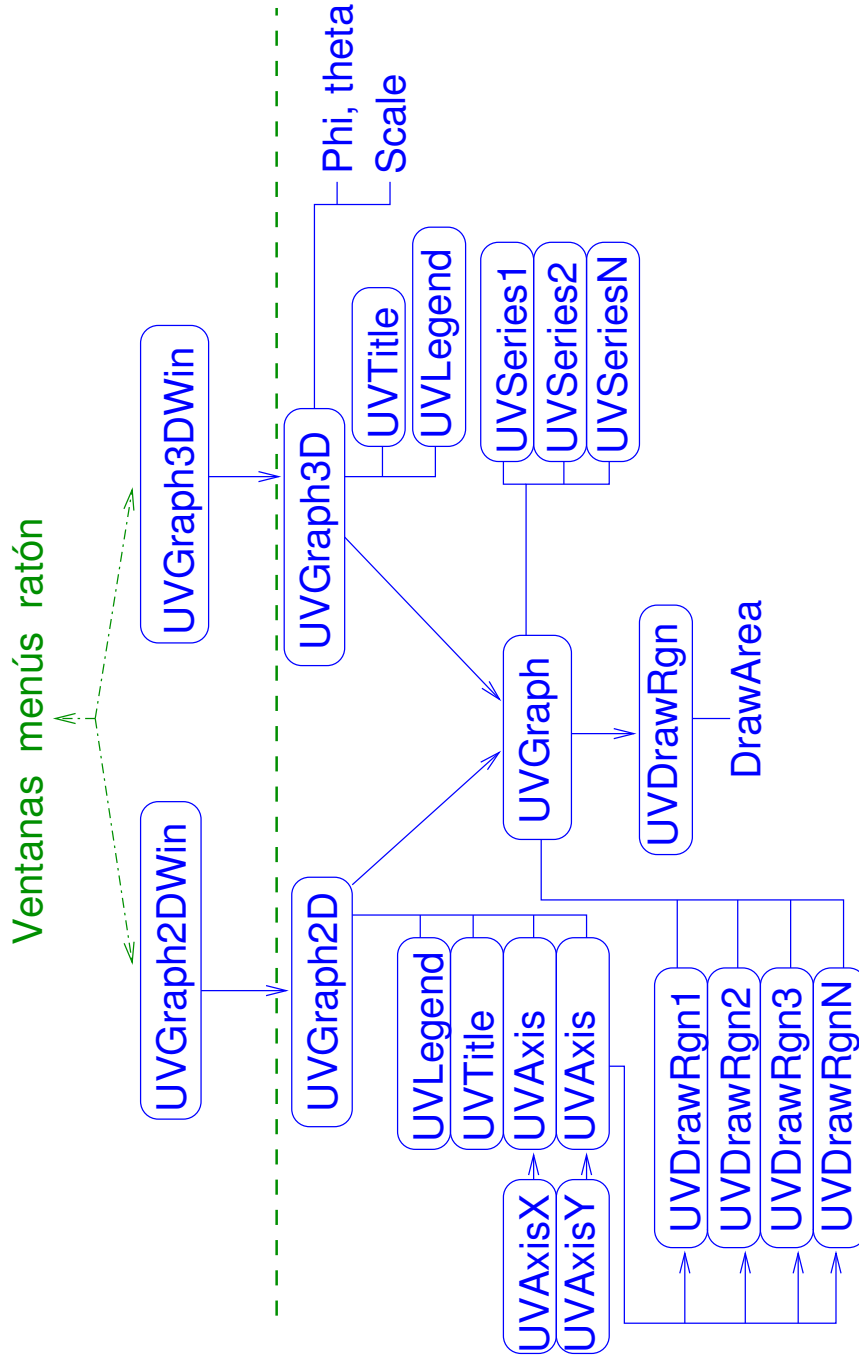


Figura B.35: Estructura interna del UVAS para gestionar las gráficas.

Estas competencias están definidas en el siguiente fichero de cabecera:

```
class UVGraph2D : public UVGraph
{
public:
    UVGraph2D();
    virtual ~UVGraph2D();
    virtual bool OnDbClick(const GDCoordPair &);
    virtual GDRect GetArea() const;
    virtual bool GrResize(const GDRect &rTotalSize, const UVGenericDrawer *pG);

    bool OnDraw(const UVGenericDrawer *pGraph);
    void SaveTo(std::ostream &os) const;
    void LoadFrom(std::istream &);
    virtual std::string ToFile(const std::string &szEle, const std::string
        &fileName, FILEFORMAT fileFormat) const;
    UVAxis * getXAxis() const { return pxAxis; }
    UVAxis * getYAxis() const { return pyAxis; }
    std::string GetGraphTitle() const { return pTitle->GetTitle(); }
protected:
    UVAxis * pxAxis;
    UVAxis * pyAxis;
    UVTitle * pTitle;
    UVLegend * pLegend;
    double minLogValue;
    unsigned int leftMargin;
};
```

► B.7.3

Gráficas en tres dimensiones: UVGraph3D

UVGraph3D implementa todos los datos y algoritmos necesarios para generar gráficas y representaciones en tres dimensiones. La figura B.37 muestra un ejemplo de este tipo de representación. Las competencias principales de UVGraph3D son:

- Definir qué sucede cuando las gráficas sean seleccionadas, movidas o actualizadas.
- Implementar el modo en que van a ser cargadas y grabadas a un fichero.
- Suministrar información sobre el estado y datos de la gráfica.

Las anteriores operaciones se declaran en el fichero de cabecera de la clase UVGraph3D como:

```
class UVGraph3D : public UVGraph
{
public:
    UVGraph3D();
    virtual ~UVGraph3D();
    virtual bool OnDbClick(const GDCoordPair &);
    bool OnDraw(const UVGenericDrawer *pGraph);
    virtual void UpdateMessage() = 0;
    virtual bool GrResize(const GDRect &r, const UVGenericDrawer *p);
    void SaveTo(std::ostream &os) const;
    void LoadFrom(std::istream &);
    std::string GetGraphTitle() const;
    void MillerIndex(int, int, int);
};
```

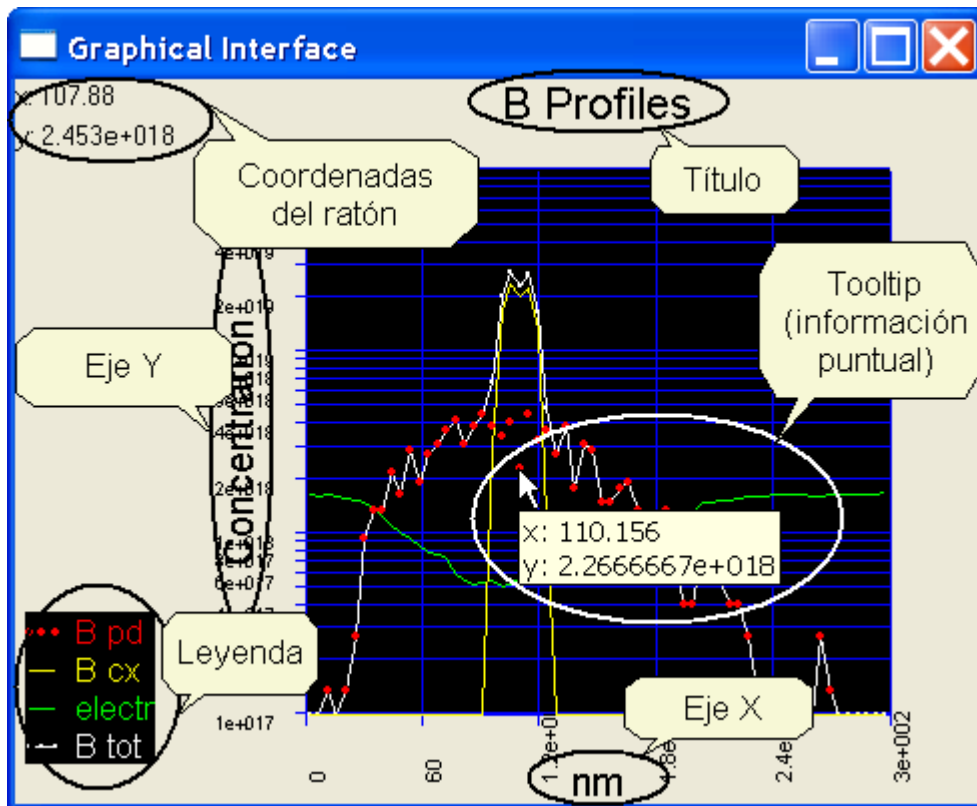


Figura B.36: Gráfica bi-dimensional donde se muestran los ejes x , y , el título y la leyenda.

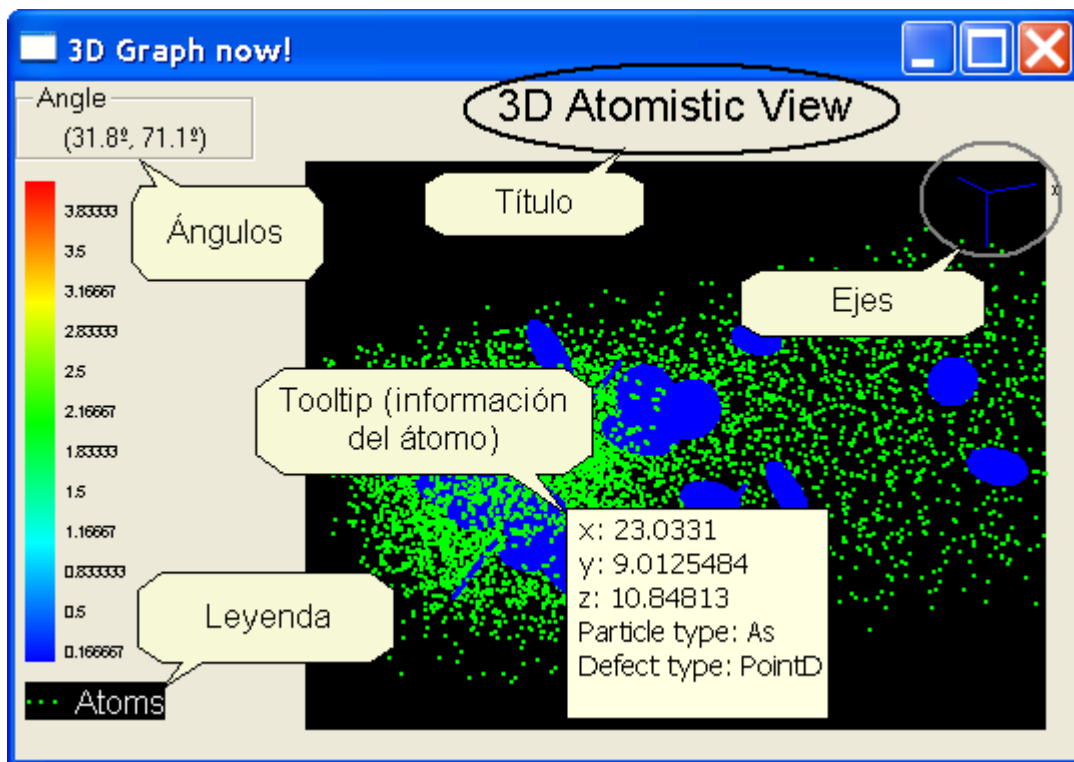


Figura B.37: Gráfica tridimensional donde se puede ver el título, los ejes x , y y z (en la esquina superior derecha) y la leyenda con los datos representados.


```

GDRect GetArea() const;
void OnAutoscale();

virtual std::string UVGraph3D::ToFile(const std::string &szEle,
const std::string &fileName, FILEFORMAT fileFormat) const;
protected:
void setScale(bool bSearchMaxMin);
UVTitle * pTitle;
UVLegend * pLegend;
double phi, theta;
scale myScale;
};

```

Las gráficas tridimensionales del UVAS no pueden escalar los ejes x , y y z de manera independiente, para evitar deformaciones en las imágenes representadas. El dibujo es escalado de manera uniforme a través de una opción de ampliación y reducción (zoom), que es manejada internamente por la estructura `scale`:

```

struct scale
{
double maxx, maxy, maxz;
double minx, miny, minz;
double diag;
double scaleFactor;
int centerX, centerY;
bool InRect(const GDCoordPair &p) const { return gdRegion.InRect(p); }
void SaveTo(std::ostream &os) const;
void LoadFrom(std::istream &is);
GDRect gdRegion;
};

```

▷ B.7.4

Clase común para las gráficas en 2 y 3 dimensiones: UVGraph

`UVGraph` es, utilizando la nomenclatura del C++, una clase virtual pura (o abstracta). Esto viene a significar que no se pueden crear objetos que utilicen dicha clase aisladamente, y que se requiere que otra clase sea derivada de `UVGraph` para definir completamente algunas funciones que están declaradas, pero no implementadas, en `UVGraph`. De esta manera `UVGraph` es una declaración de intenciones de cómo un gráfico en 2 o 3 dimensiones debe comportarse, y sobre qué funciones debe tener.

Además de esta estructura general, `UVGraph` funciona como un contenedor para el conjunto de regiones dibujables (`UVDRAWRgn`) que una gráfica requiere. Estas regiones pueden ser ejes, el título, la leyenda o el área de dibujo propiamente dicha.

El fichero de cabecera de esta clase `UVGraph` es el siguiente:

```

class UVGraph : public UVDRAWRgn
{
public:
UVGraph();
virtual ~UVGraph();
enum FILEFORMAT { XYZ, PS, EPS, PNG, FIG, GIF, RAW, POV };
//Defining pure virtual base functions.
virtual bool OnDraw(const UVGenericDrawer *);
};

```

```

virtual bool GrResize(const GDRect &, const UVGenericDrawer *);
virtual bool OnDbIClick(const GDCoordPair &);
virtual void UpdateMessage() = 0;
virtual std::string GetGraphTitle() const = 0;
virtual void OnAutoscale() = 0;
virtual void SaveTo(std::ostream &) const;
virtual void setWindowTitle(const std::string &) = 0;
void LoadFrom(std::istream &);
bool isScaled() const { return bScaled; }
virtual std::string ToFile(const std::string &, const std::string &,
    FILEFORMAT) const = 0;
std::vector<UVSeries *> stlSeries;
void getDrawArea(ExtVector &init , ExtVector &end) const;
unsigned int serieSelected;
protected:
std::vector<UVDDrawRgn *> stlDrawRgns;
bool bScaled;
unsigned int uiBackgroundColor;
};

```

Muchas de las anteriores funciones se encuentran repetidas en las clases de UVGraph2D y UVGraph3D (que también son más completas, pero menos genéricas), debido a que esas clases más concretas son derivadas de ésta mucho más general.

► B.7.5

Gestionando un área de dibujo rectangular: UVDDrawRgn

La idea con la que se construye UVDDrawRgn es la de implementar los datos y algoritmos que se necesitan para hacerse cargo de un área rectangular en la que se pueda dibujar algo, de esta manera:

Datos Básicamente el rectángulo del área contenida.

Algoritmos Los algoritmos necesarios para redibujar el contenido (actualización), cuando se cambie el tamaño del área y al seleccionar el área con doble “click”.

Así pues las anteriores ideas se implementan bastante directamente, como puede verse en el siguiente fichero de cabecera:

```

class UVDDrawRgn
{
public:
    UVDDrawRgn() {};
    virtual ~UVDDrawRgn() {};
    //Redrawing the display
    virtual bool OnDraw(const UVGenericDrawer *) = 0;
    //Recomputing the Area
    virtual bool GrResize(const GDRect &, const UVGenericDrawer *) = 0;
    //Showing the correct dialog.
    virtual bool OnDbIClick(const GDCoordPair &) = 0;
    virtual GDRect GetArea() const { return Area; }
    virtual void SaveTo(std::ostream &os) const;
    void LoadFrom(std::istream &is);
protected:
    GDRect Area;
};

```

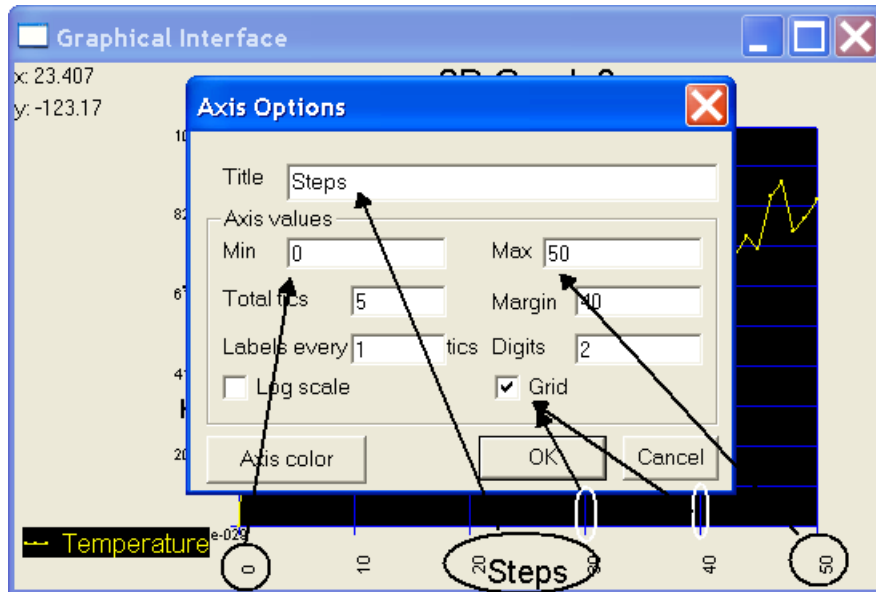


Figura B.38: Captura de pantalla de UVAxisX y su cuadro de diálogo asociado.

UVDrawRgn es una clase tan genérica que, además de UVGraph, las siguientes clases también son derivadas de ella:

UVAxisX. Gestiona el eje x en gráficas bi-dimensionales. Figura B.38.

UVAxisY. Gestiona el eje y en gráficas bi-dimensionales. Figura B.39.

UVTitle. Dibuja y gestiona el título de las diferentes gráficas. Figure B.40.

UVLegend. Controla y muestra la leyenda con los datos dibujados. Figure B.41.

► B.7.6

Un sistema de dibujo genérico: GenericDrawer

Dividir el código en dos partes, de las cuales una se encargue de los botones, ventanas, menús, ratón, etc... y la otra implementa las rutinas y algoritmos necesarios es muy ventajoso a la hora de organizar y, posteriormente, portar el programa a otras plataformas, pero por otra parte presenta el problema de cómo se puede lograr que estas rutinas genéricas puedan tener acceso a la pantalla para dibujar algo de una manera estándar y que no dependa del entorno gráfico (Windows, X-Window...) usado. Esto es: las rutinas requieren dibujar líneas, puntos y otros tipos de objetos pero, como estas acciones suelen requerir el uso de funciones implementadas en la parte dependiente del entorno gráfico, nos encontraríamos en un callejón sin salida.

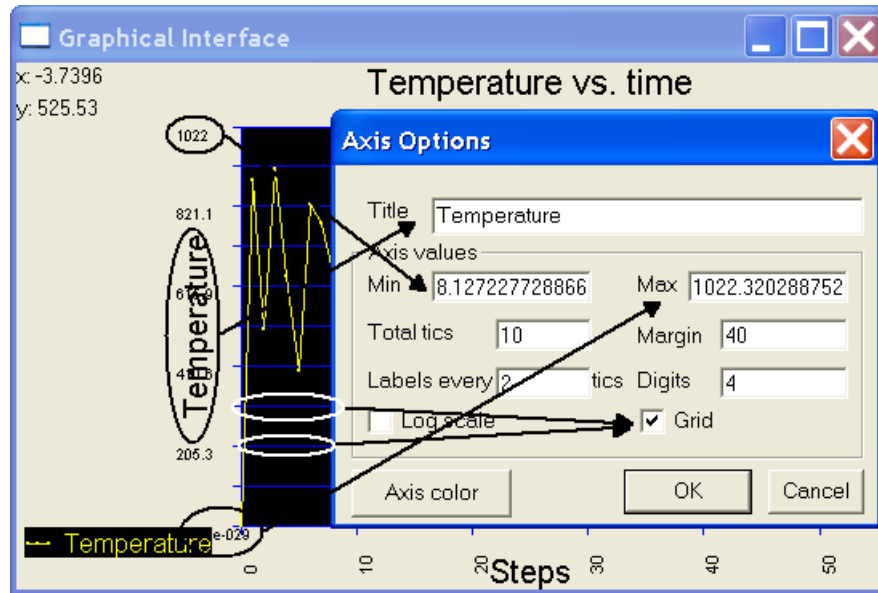


Figura B.39: Captura de pantalla de UVAxisY y su cuadro de diálogo asociado.

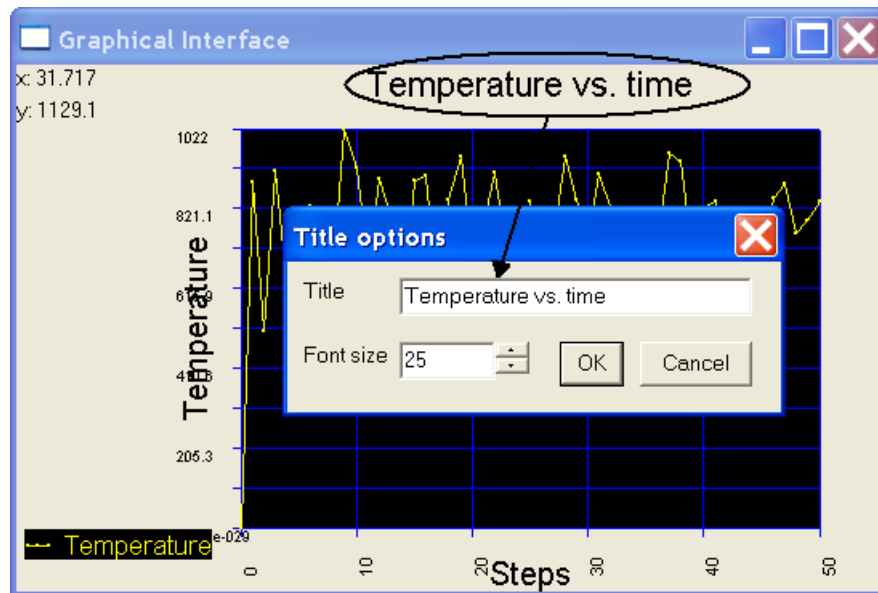


Figura B.40: Captura de pantalla de UVTitle y su cuadro de diálogo asociado.

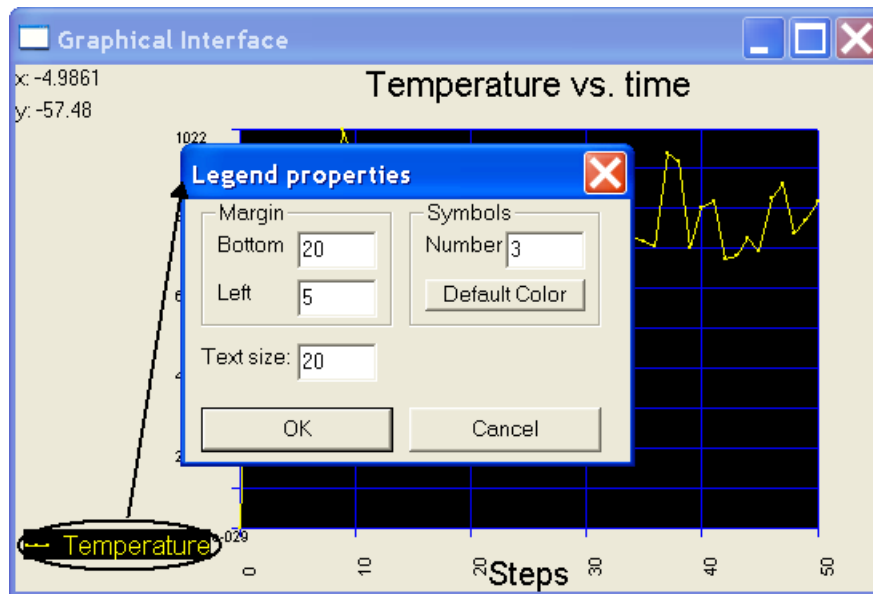


Figura B.41: Captura de pantalla de UVLegend y su cuadro de diálogo asociado.

El anterior problema se ha resuelto en UVAS usando una clase de dibujo genérico `GenericDrawer`. Esta clase es una clase virtual pura abstracta que define operaciones gráficas estándar como dibujar puntos, líneas, cuadrados, de una forma genérica, pero que no implementa ninguna de estas operaciones en absoluto. La implementación corre a cargo de otras clases, derivadas de ésta, que hacen un uso exhaustivo y por tanto, dependen, del entorno gráfico concreto utilizado, como por ejemplo las clases `CUVWinDrawer` o `GtkDrawer`. De esta manera cuando se crea un objeto tipo, por ejemplo, `UVGraph2DWin`, éste crea un `CUVWinDrawer`, y le pasa la parte genérica, `GenericDrawer` de éste a su parte genérica, `UVGraph2D`. La parte genérica opera con este gestor genérico de gráficos llamando a sus funciones que, gracias a las habilidades del lenguaje C++, terminan llamando a las funciones de `CUVWinDrawer` el cual, por fin, dibuja los círculos, texto o líneas requeridos.

Con estas ideas en mente, se ha creado el siguiente fichero de cabecera para `GenericDrawer`:

```
class UVGenericDrawer
{
public:
    UVGenericDrawer();
    virtual ~UVGenericDrawer();
    //Give x,y begin and end
    virtual void Line(const GDRect&) const = 0;
    virtual void Line(const GDCoordPair&,const GDCoordPair&) const = 0;
    //Certain shapes
    virtual void Cross(const GDCoordPair&, unsigned int) const = 0;
    virtual void Plus(const GDCoordPair&, unsigned int) const = 0;
    virtual void Circle(const GDCoordPair&, unsigned int) const = 0;
```

```

virtual void CircleColor(const GDCoordPair &cp,
    unsigned int uiSize, unsigned int color) const = 0;
virtual void RectangleColor(const GDRect &r, unsigned int color) const = 0;
//Give x,y, text and angle
virtual void Text(const GDCoordPair&,const std::string &,
    double textDegrees= 0, unsigned int textSize= 0) const = 0;
//Give text and angle, return text size.
virtual GDCoordPair TextExtent(const std::string &, double =0,
    unsigned int=0) const = 0;
// Color
virtual void SetColor(int color =-1) const;
virtual void SetTextColor(int color =-1) const =0;
unsigned int GetColor() const;
protected:
    mutable unsigned int uiColor;
};

```

que, a su vez, utiliza unas estructuras genéricas GDRect y GDCoordPair para gestionar rectángulos:

```

struct GDRect
{
    GDRect(int l=0,int t=0, int r=0, int b=0);
    int top;
    int bottom;
    int left;
    int right;
    bool InRect(const GDCoordPair &) const;
    void cutPoint(GDCoordPair &p) const;
};

```

y pares de puntos:

```

struct GDCoordPair
{
    GDCoordPair(int x=0, int y=0);
    int cx;
    int cy;
};

```

▷ B.8

Extrayendo los datos en un modo no gráfico

El dibujo y representación de los datos generados por los simuladores necesita de un usuario que, de manera interactiva, utilice los recursos del programa para seleccionar el gráfico que debe ser representado en la forma deseada. Esto es realmente útil a la hora de analizar los datos, pero puede dificultar sobremanera el método de obtención automática de éstos, que se necesitaría para la creación de un conjunto de listas de chequeo del programa u otro tipo de procesos automáticos. Por ello UVAS incorpora un modo de extraer los datos como texto cuando la simulación ha llegado a su fin. Así, el usuario llama a un programa UVAS en modo “texto” aportándole todos los datos necesarios, y al final del proceso obtiene una serie de ficheros con los resultados, sin más intervención por su parte.

Los datos que se pueden obtener son representados internamente como `UVSeries`, que a su vez son seleccionadas de todos los posibles datos mostrados por el selector de series (ver figura B.42). Para proceder así se utiliza el interfaz gráfico mostrado en la figura B.43. El usuario obtiene las `UVSeries` del selector, o bien se las pasa, y al final de la simulación estas series (que pasan a ser gestionadas por la clase `OutputData`), serán mostradas como texto simple o bien grabadas en una serie de ficheros de texto. El usuario también puede visualizar en cualquier momento estos datos limitándose a usar el botón “print” del programa, como se muestra en la figura B.44.

La estructura interna de `OutputData` se muestra en la figura B.45. Una vez más, se ha definido una clase interfaz que maneja el soporte gráfico (`COutputData`), es decir, los botones, ratón, etc... y otra parte estándar C++ que implementa todos los algoritmos de datos. La parte dependiente del entorno gráfico se deriva de la otra, facilitando así la exportación del programa a otras plataformas, puesto que se aísla el código no exportable en la clase `COutputData`.

`OutputData` contiene todas las series que deben imprimirse, así como las rutinas que se encargarán de su impresión, y de guardar y obtener de un fichero toda la estructura. Su fichero de cabecera es:

```
#if !defined(_OUTPUTDATA_H_)
#define _OUTPUTDATA_H_
#include <set>
#include <fstream>
class UVSeries;
class OutputData
{
public:
    OutputData();
    virtual ~OutputData();
    virtual void LoadFrom(std::istream &);
    virtual void SaveTo(std::ostream &) const;
    virtual void PrintOutput(const UVSeries *pSer = 0) const;
protected:
    std::set<UVSeries *> stISeries;
private:
    void printSeries(const UVSeries *) const;
};
#endif // !defined(_OUTPUTDATA_H_)
```

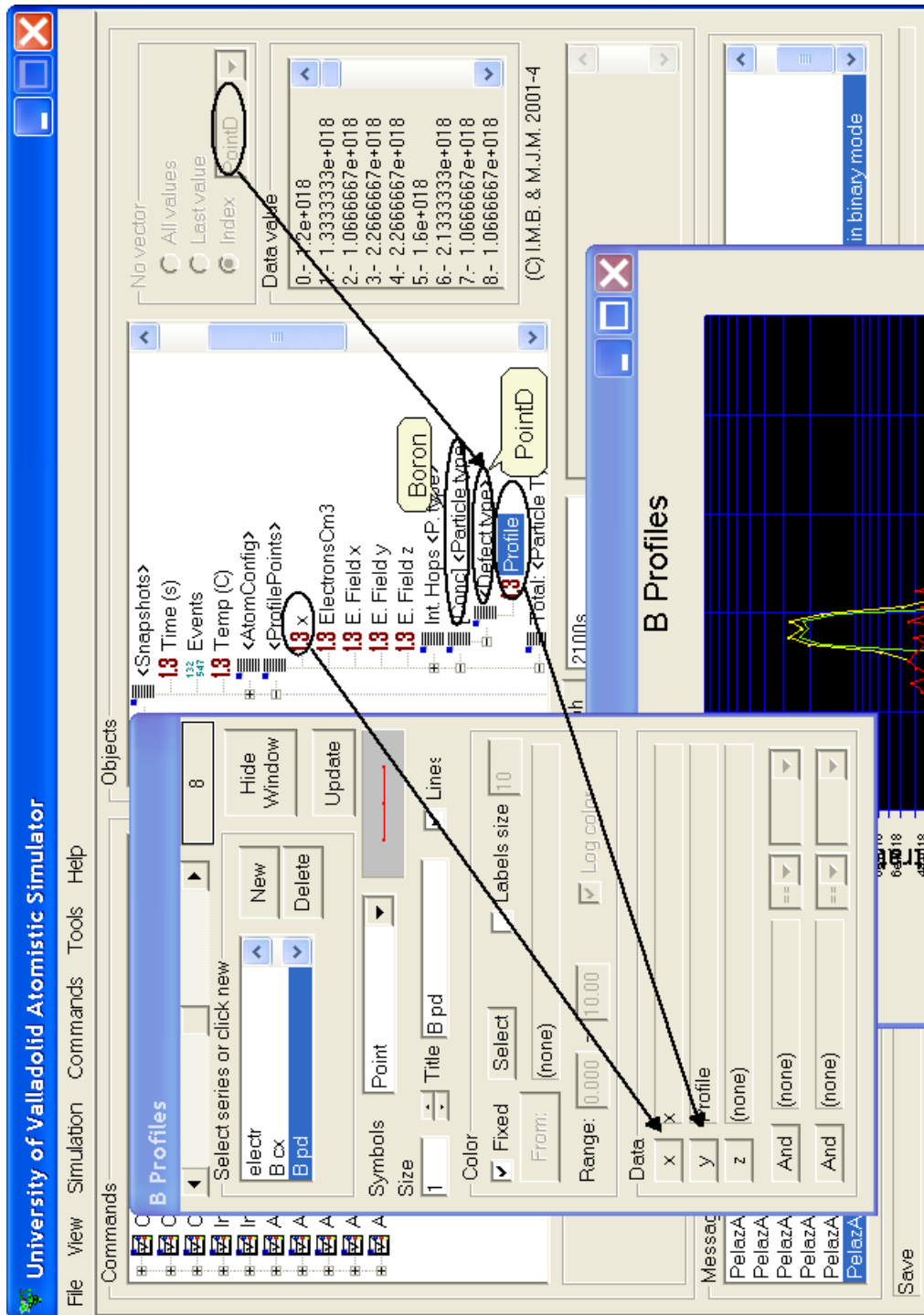


Figura B.42: El selector de series permite asignar y configurar los distintos datos que, tomados del "Data Browser" pasan a formar parte de una serie. Cada una de estas series puede ser dibujada, o bien gestionada al final del proceso para imprimir sus datos.

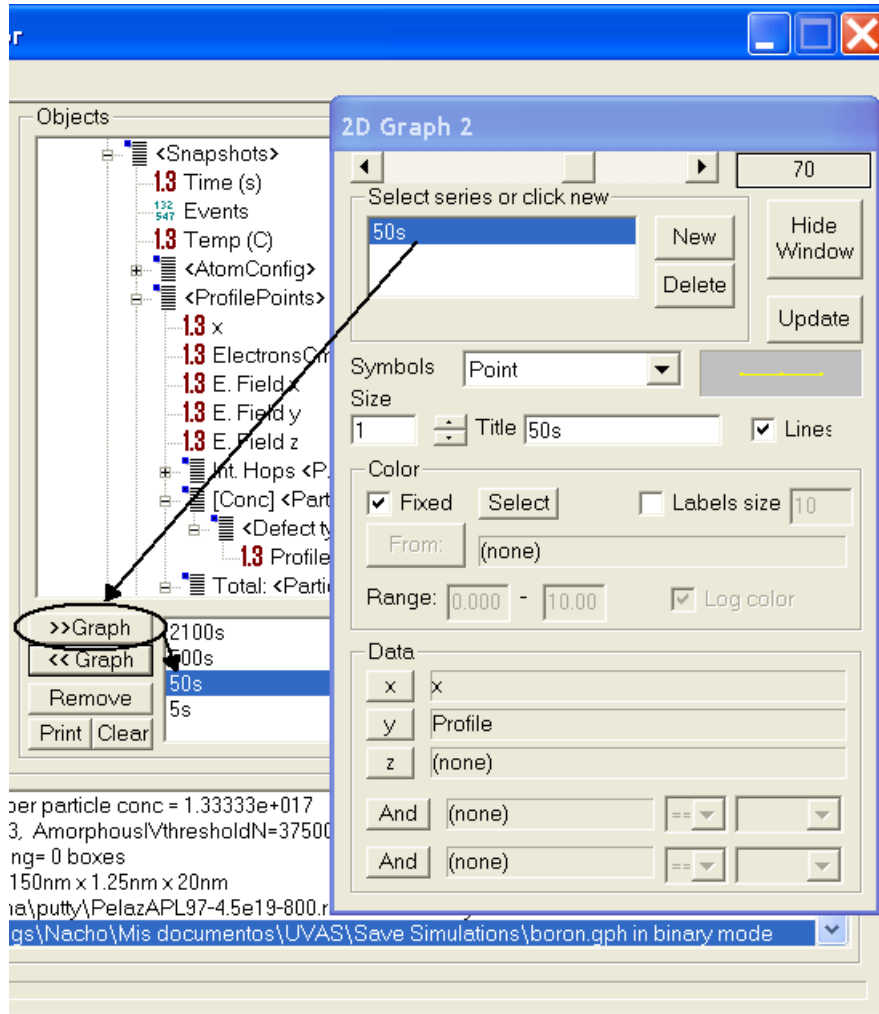


Figura B.43: Interfaz que gestiona los datos a obtener.

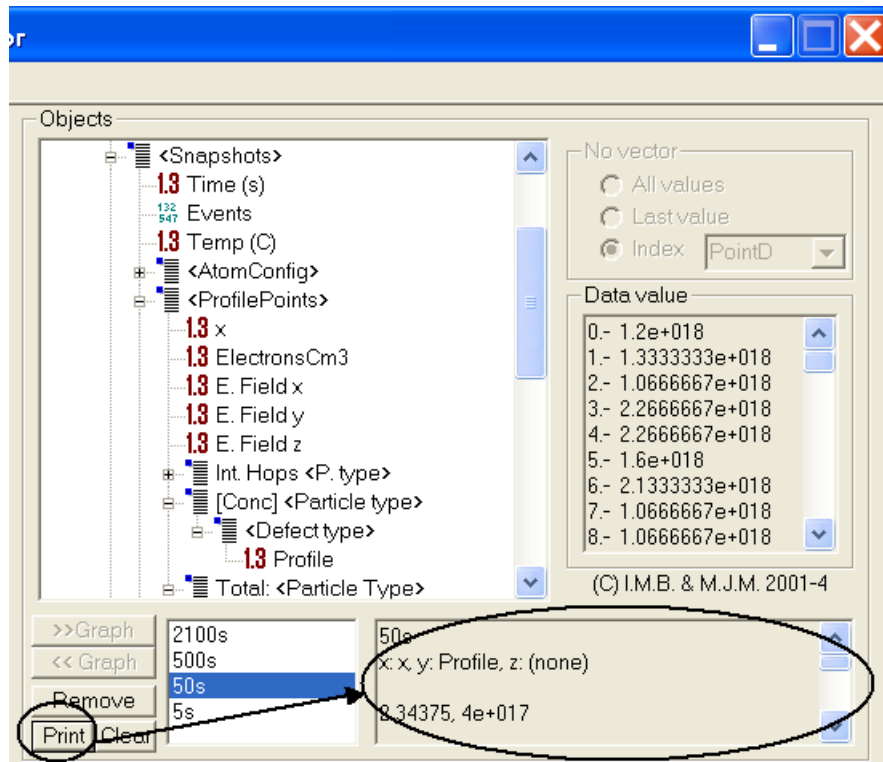


Figura B.44: Captura de pantalla de cómo se obtiene los datos de saluda utilizando la estructura del "OutputData".

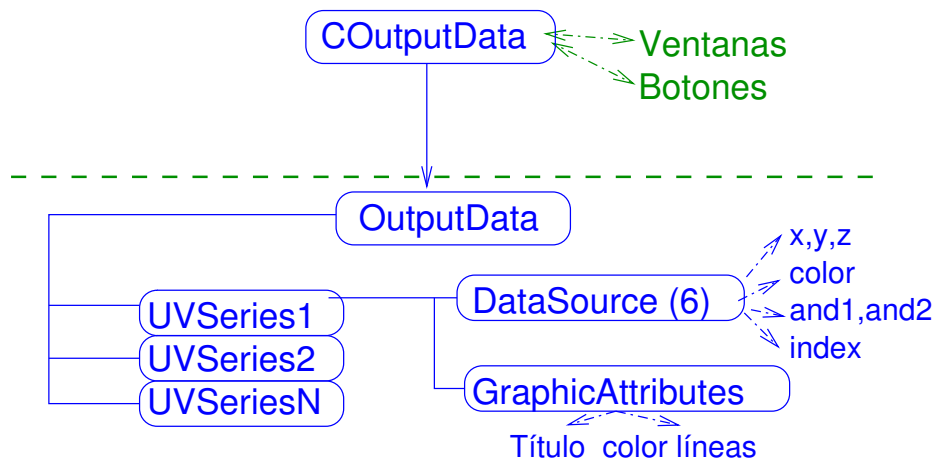


Figura B.45: Estructura interna de COutputData.

Parte VI

Índices y bibliografía

Bibliografía

- Agarwall, A., Gossmann, H.-J., Eaglesham, D. J., Herner, S. B., Fiory, A. T., and Haynes, T. E. (1999). *Appl. Phys. Lett.*, 74:2435. 96
- Alippi, P. and Colombo, R. (2000). Lattice-strain field induced by {311} self-interstitial defects in silicon. *Phys. Rev.B*, 62:1815. 87
- Aoki, N., Yaegashi, T., Takeuchi, Y., Fujiwara, M., Kusunoki, N., Sato, T., Mizushima, I., Tsunashima, Y., h. Hazama, Aritome, S., and Shirota, R. (2000). Dopant diffusion in silicon substrate during oxynitride process. volume 610, page B3.5. *Mat. Res. Soc.* 257
- Arai, A. and Takeda, S. (1997). Self-interstitial clustering in crystalline silicon. *Phys. Rev. Lett.*, 78:4265. 86
- Arias, J. (1995). *Simulacion de la implantacion ionica en cistales incluyendo baja energia y acumulacion de danado*. PhD thesis, University of Valladolid. 76
- Arias, J., Jaraiz, M., Pelaz, L., Bailon, L. A., and Barbolla, J. (1995). Low energy ion implantation simulation using a modified binary collision approximation code. *Nucl. Instr. Meth. B*, 102:228–231. 73
- Avant! (2001). *Taurus reference Manual*. Fremont, CA,USA. 304
- Barna, P. B. and Adamik, M. (1998). *Thin Solid Films*, 317:17. 61
- Berding, M. A., Sher, A., van Schilfgaarde, M., Rousseau, P. M., and Spicer, W. E. (1998). Deactivation in heavily arsenic-doped silicon. *Appl. Phys. Lett.*, 72(12):1492. 306
- Biersack, J. P. and Haggmark, L. G. (1980). A monte carlo computer program for the transport of energetic ions in amorphous targets. *Nuclear Instruments and Methods*, 174(1-2):257–269. 57, 73
- Blöchl, P. E., Smargiassi, E., Car, R., Lacks, D. B., Andreoni, W., and Pantelides, S. (1993). First-principles calculations of self-diffusion constants in silicon. *Phys. Rev. Lett.*, 70:2435. 85

- Bongiorno, A. and Colombo, L. (1998). Interaction between a monovacancy and a vacancy cluster in silicon. *Phys. Rev. B*, 57:8767. 90, 91, 126, 271, 421
- Bracht, H. (2000). Diffusion mechanisms and intrinsic point-defect properties in silicon. *Mat. Res. Soc. Bulletin*, 25(6):22. 82, 84, 85, 435
- Bracht, H., Haller, E. E., and Clark-Phelps, R. (1998). Silicon self-diffusion in isotope heterostructures. *Phys. Rev. Lett.*, 81:393–396. 85, 220, 282
- Bracht, H., Stolwijk, N. A., and Mehrer, H. (1995). Properties of intrinsic point defects in silicon determined by zinc diffusion experiments under nonequilibrium conditions. *Phys. Rev. B*, 52(16):16542. 85, 221, 261, 282, 425
- Casali, R. A., Rücker, H., and Methfessel, M. (2001). Interaction of vacancies with interstitial oxygen in silicon. *Appl. Phys. Lett.*, 78:913. 85, 97
- Caturla, M. J., Johnson, M. D., and de la Rubia, T. D. (1998a). The fraction of substitutional boron in silicon during ion implantation and thermal annealing. *Applied Physics Letters*, 72(21):2736–8. 195
- Caturla, M. J., Lilak, A., Johnson, M. D., Giles, M., de la Rubia, T. D., Law, M., and Foad, M. (1998b). Atomic scale modeling of boron transient diffusion in silicon. In Matsuo, J., Takaoka, G., and Yamada, I., editors, *1998 International Conference on Ion Implantation Technology*. 195
- Centoni, S. A., Sadigh, B., Gilmer, G. H., Caturla, M. J., Lenosky, T. J., de la Rubia, T. D., and Musgrave, C. B. APS/123-QED. 304
- Cerofolini, G. F., Manini, P., Meda, L., Polignano, M. L., Queirolo, G., Nava, F., Ottaviani, G., and Gallorini, M. (1986). Thermodynamic and kinetic properties of arsenic-implanted silicon. *Thin Solid Films*, 135(1):59–72. 199, 424
- Chadi, D. J. and Chang, K. J. (1988). Magic numbers for vacancy aggregation in crystalline si. *Phys. Rev. B*, 38:1523. 90
- Chadi, D. J., Citrin, P. H., Park, C. H., Adler, D. L., and andH. J. Gossmann, M. A. M. (1997). Fermi-level-pinning defects in highly n-doped silicon. *Phys. Rev. Lett.*, 79(24):4834. 306
- Chao, H. S., Griffin, P. B., and Plummer, J. D. (1997). The influence of amorphizing implants on boron diffusion in silicon. In *Mat. Res. Soc. Symp. Proc. Vol 469*, volume 469, pages 347–352. Mat. Res. Soc. 246, 247, 427
- Chichkine, M. P. and de Souza, M. M. (2002). Dynamics of self-interstitial cluster formation in silicon. *Phys. Rev. B*, 66:045205. 25, 252, 260

- Chichkine, M. P., de Souza, M. M., and Narayanan, E. M. S. (2002). Growth of precursors in silicon using pseudopotential calculations. *Phys. Rev. Lett.*, 88(8):085501. 25, 86, 252
- Colombeau, B., Cristiano, F., Altibelli, A., Bonafos, C., Assayag, G. B., and Claverie, A. (2001). Atomistic simulations of extrinsic defects evolution and transient enhanced diffusion in silicon. *Appl. Phys. Lett.*, 78:940. 89, 123, 421
- Cowern, N., van de Walle, G., Zalm, P. C., and Oostra, D. J. (1992). Reactions of point defects and dopant atoms in silicon. *Phys. Rev. Lett.*, 69(1):116–119. 195, 286, 287, 424, 429
- Cowern, N. E. B., Alquier, D., Omri, M., Claverie, A., and Nejim, A. (1999a). Transient enhanced diffusion in preamorphized silicon: the role of the surface. *Nucl. Instr. Meth. B*, 148:257–261. 95, 151, 241, 242, 253, 254, 256, 427
- Cowern, N. E. B., Janssen, K. T. F., van de Walle, G. F. A., and J.-Gravesteijn, D. (1990). Impurity diffusion via an intermediate species: The b-si system. *Phys. Rev. Lett.*, 65:2434. 97, 285
- Cowern, N. E. B., Mannino, G., Stolk, P. A., Roozeboom, F., Huizing, H., van Berkum, J., Cristiano, F., Claverie, A., and Jaraiz, M. (1999b). Cluster ripening and transient enhanced diffusion in silicon. *Materials Science in Semiconductor Processing*, 2:369–376. 221, 425
- Cowern, N. E. B., Mannino, G., Stolk, P. A., Roozeboom, F., Huizing, H. G. A., van Berkum, J. G. M., Cristiano, F., Claverie, A., and Jaraiz, M. (1999c). Energetics of self-interstitial clusters in Si. *Phys. Rev. Lett.*, 82(22):4460–4463. 86, 87, 224, 225, 258, 259, 421, 425, 427
- Cowern, N. E. B. and Rafferty, C. S. (2000). Enhanced diffusion in silicon processing. *Mat. Res. Soc. Bulletin*, 25(6):39. 97
- Cowern, N. E. B., van de Valle, G. F. A., Gravesteijn, D. J., and Vriezema, C. J. (1991). Experiments on atomic-scale mechanisms of diffusion. *Phys. Rev. Lett.*, 67(2):212–215. 195, 223, 224, 286, 287, 424, 425, 429
- Cristiano, F., Grisolia, J., Colombeau, B., Omri, M., de Mauduit, B., Claverie, A., Giles, L. F., and Cowern, N. E. B. (2000). Formation energies and relative stability of perfect and faulted dislocation loops in silicon. *J. Appl. Phys.*, 87(12):8420. 89, 96, 123, 124, 421, 435
- Dabrowski, J. (2000). *Special defects in Semiconducting Materials*. Zurich. 82, 83
- Dannefaer, S., Masher, P., and Kerr, D. (1986). Monovacancy formation enthalpy in silicon. *Phys. Rev. Lett.*, 56(20):2195. 85

- de la Rubia, T. D. and Gilmer, G. H. (1995). Structural transformations and defect production in ion implanted silicon: A molecular dynamics simulation study. *Phys. Rev. Lett.*, 74(13):2507. 91, 93
- de Mauduit, B., Laanab, L., Bergand, C., Faye, M. M., Assayag, G. B., and Claverie, A. (1994). Identification of eor defects due to the regrowth of amorphous layers created by ion bombardment. *Nuc. Instrum. Methods Res. B*, 84(2):190. 88
- de Souza, M. M. (2002). (private communication). 262
- de Walle, C. G. V., Laks, D. B., Neumark, G. F., and Pantelides, S. T. (1993). First principles calculations of solubilities and doping limits: Li, na and n in znse. *Phys. Rev. B*, 47(15):9245–9434. 173, 279, 286
- de Walle, C. G. V., McFeely, F. R., and Pantelides, S. T. (1988). Fluorine-silicon reactions and the etching of crystalline silicon. *Phys. Rev. Lett.*, 61(16):1867–1870. 202, 203
- Deaven, D. M. and Ho, K. M. (1995). Molecular geometry optimization with a genetic algorithm. *Phys. Rev. Lett.*, 75(2):288–291. 39
- Dokumaci, O., Ronsheim, P., D’emic, C., Domenicucci, A. G., Hegde, S., Kozlowaki, P., and Wong, H.-S. P. (1999). Transient enhanced diffusion and dose loss of indium in silicon. In *Mat. Res. Soc. Sym. Proc.*, volume 568, pages 205–210. Mat. Res. Soc. 137
- Dokumaci, O., Rousseau, P., Luning, S., Krishnamoorthy, V., Jones, K. S., and Law, M. E. (1995). Transmission electron microscopy analysys of heavily as-doped, laser and thermally annealed layers in silicon. *J. Appl. Phys.*, 78(2):828. 310
- Dorigo, M., Bonabeau, E., and Theraulaz, G. (2000). Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16:851–871. 33
- Dunham, T., Fastenko, P., Qin, Z., and Diebel, M. (2003). *IEICE Trans. Electron.*, E86-C:276. 307
- Eaglesham, D. J., Stolk, P. A., Gossmann, H. J., and Poate, J. M. (1994). Implantation and transient b diffusion in si: The source of the interstitials. *Appl. Phys. Lett.*, 65(18):2305. 87, 88, 252, 256, 258, 260, 421, 428
- Elliman, R. G., Linnros, J., and Brown, W. L. (1988). *Mat. Res. Soc. Bulletin*, 100:363. 93, 133

- Estreicher, S. K., Gharaibeh, M., Fedders, P. A., and Ordejon, P. (2001). Unexpected dynamics for self-interstitial clusters in silicon. *Phys. Rev. Lett.*, 86:1247–1250. 25, 252
- Estreicher, S. K., Hastings, J. L., and Fedders, P. A. (1997). The ring-hexavacancy in silicon: A stable and inactive defect. *Appl. Phys. Lett.*, 70(4):432. 90
- Ezaki, T., Ikezawa, T., and Hane, M. (2002). Investigation of realistic dopant fluctuation induced device characteristics variation for sub-100nm CMOS by using atomistic 3d process/device simulator. pages 311–314. IEDM. 252, 320
- Fahey, P. M., Griffin, P. B., and Plummer, J. D. (1989). Point defects and dopant diffusion in silicon. *Reviews of Modern Physics*, 61(2):289–384. 82, 85, 97, 98, 171, 279, 280, 281, 282, 302, 303, 304
- Fair, R. B. (1981a). *Impurity doping processes in Silicon*, chapter Concentration profiles of diffused dopants in silicon, page 315. North-Holland. 192, 193, 199, 424
- Fair, R. B. (1981b). *Impurity doping processes in Silicon*. North-Holland. 279, 285, 286, 303, 304, 305, 429, 430
- Falster, R. and Voronkov, V. V. (2000). Intrinsic point defects and their control in silicon crystal growth and wafer processing. *Mat. Res. Soc. Bulletin*, 25(6):28. 90
- Ghoshtagore, R.Ñ. (1971). Donor diffusion dynamics in silicon. *Phys. Rev. B*, 3(2):397. 199, 424
- Giese, A., Bracht, H., Stolwijk, N. A., and Baither, D. (2000). Microscopic defects in silicon induced by zinc out-diffusion. *Materials Science and Engineering B*, 71:160. 221, 281, 425
- Giles, M. (1991). Transient phosphorus diffusion below the amorphization threshold. *Journal of the Electrochemical Society*, 138(4):1160. 271
- Gilmer, G. H., de la Rubia, T. D., Stock, D. M., and Jaraiz, M. (1995). Diffusion and interactions of point defects in silicon: molecular dynamics simulations. *Nucl. Instr. Meth. B*, 102:247–55. 25, 252, 263
- Gilmer, G. H., Huand, H., de la Rubia, T. D., Torre, J. D., and Baumann, F. (2000). Lattice monte carlo models of thin film deposition. *Thin Solids Films*, (365):189–200. 61, 68
- Goldberg, R. D., Williams, J. S., and Elliman, R. G. (1995). Amorphization of silicon by elevated temperature ion irradiation. *Nucl. Instrum. Methods Phys. Res. B*, 106:242. 229, 230, 231, 295, 296, 426, 430

- Griffin, P. B., Packan, P. A., and Plummer, J. D. (1991). Consistent models for point defects in silicon. pages 4–7. Int. Workshop on VLSI process and device modelling (VPAD). 253
- Haddara, Y. M., Folmer, B. T., Law, M. E., and Buyuklimanli, T. (2000). Accurate measurements of the intrinsic diffusivities of boron and phosphorous in silicon. *Appl. Phys. Lett.*, 77(13):1976. 192, 424
- Hane, M., Ikezawa, T., and Gilmer, G. H. (2000). Di-interstitial diffusivity and migration path calculations based on tight-binding hamiltonian molecular dynamics. pages 119–122. International conference on simulation of semiconductor processes and devices, SISPAD. 25, 252, 253, 260, 289
- Harris, R. D., Newton, J. L., and Watkins, G. D. (1982). Negative-u properties for interstitial boron in silicon. *Phys. Rev. Lett.*, 48(18):1271–4. 283
- Harris, R. D., Newton, J. L., and Watkins, G. D. (1987). Negative-u defect: Interstitial boron in silicon. *Phys. Rev. B*, 36(2):1094–1104. 283
- Harrison, W. A. (1998). Diffusion and carrier recombination by interstitials in silicon. *Phys. Rev. B*, 57(16):9727–9735. 282
- Heermann, D. W. (1990). *Computer simulation methods*. Springer-Verlag, 2 edition. ISBN 0-387-55210-7. 15, 27
- Hernandez-Mangas, J. M. (2000). *Simulación BCA de la Implantación iónica en materiales cristalinos: mejora de los modelos físicos y del tiempo de cálculo*. PhD thesis, University of Valladolid. 76
- Herner, S., Gossmann, H.-J., and Tung, R. T. (1998). Native point defects in Si with a SiO_2 film: Single crystal versus polycrystalline. *Appl. Phys. Lett.*, 72(18):2289. 96
- Hill, C. (1981). *Semiconductor Silicon*. Electrochem. Soc., NY. USA. 192, 199, 424
- Ho, K.-M., Shvartsburg, A. A., Pan, B., Lu, Z.-Y., Wang, C.-Z., Wacker, J. G., Fye, J. L., and Jarrold, M. F. (1998). Structures of medium-sized silicon clusters. *Nature (London)*, 392:582–584. 38
- Hobler, G. and Moroz, V. (2001). Initial conditions for transient enhanced diffusion: Beyond the plus-factor approach. In Tsoukalas, D. and Tsamis, C., editors, *Simulation of semiconductor processes and devices 2001*, pages 35–37, Vienna. SISPAD. 271

- Hockney, R. W. and Eastwood, J. W. (1988). *Computer simulation using particles*. Adam Hilger by IOP Publishing LTD, 335 East 45th Street, New York, NY 10017-3483, USA. ISBN 0-85274-392-0. 27
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. MIT Press. 33
- Holland, O. W. and White, C. W. (1991). Ion-induced damage and amorphization in si. *Nuc. Instr. Meth. B*, 59/60:353. 90, 421
- Holland, O. W., Xie, L., Nielsen, B., and Zhou, D. S. (1996). *J. Electronics Mat.*, 25:99. 94, 421
- Hourahine, B., Jones, R., Safonov, N., Öberg, S., Briddon, P. R., and Streicher, S. K. (2000). Identification of the hexavacancy in silicon with the b_{80}^4 optical center. *Phys. Rev. B*, 61(19):12594. 90
- Hu, S. M. (1973). *Atomic Diffusion in Semiconductors*. Plenum, London. 98
- Hu, S. M. (1981). *Defects in semiconductors*. North-Holland, New York. 96, 257
- Hu, S. M. (1994). *Mater. Sci. Eng*, R13:105. 82
- Huang, H., Gilmer, G. H., and de la Rubia, T. D. (1998). An atomistic simulator for thin film deposition in three dimensions. *J. App. Phys.*, 84(7):3636–3649. 61, 63, 68, 70, 420
- Huang, M. B., Simpson, T. W., and Mitchell, I. V. (1997). Effects of phosphorus doping on boron transient enhanced diffusion. *Appl. Phys. Lett.*, 70(9):1146. 239, 426
- Ishikawa, Y., Sakina, Y., Tanaka, H., Matsumoto, S., and Niimi, T. (1982). The enhanced diffusion of arsenic and phosphorous in silicon by thermal oxidation. *J. Electrochem. Soc.*, 129(3):644–8. 199, 424
- Jaeger, R. C. (1997). *Introduction to microelectronic fabrication*. Modular series on solid state devices. Addison-Wesley publishing company. 4
- Jain, S. C. and Roulston, D. J. (1991). A simple expression for band gap narrowing. *Solid State Electronics*, 34:453–465. 174, 288
- Jaraiz, M., Arias, J., Rubio, J. E., Bailon, L. A., and Barbolla, J. (1993). Computer simulation of point-defect distributions generated by ion implantation. *Nulc. Instr. Meth. B*, 80-81:172–5. 76

- Jaraiz, M., Castrillo, P., Pinacho, R., Martin-Bragado, I., and Barbolla, J. (2001). Atomistic front-end process modelling: a powerful tool for deep-submicron device fabrication. In Tsoukalas, D. and Tsamis, C., editors, *Simulation of Semiconductor Processes and Devices 2001*, pages 10–17, Viena. Simulation of semiconductor processes and devices, SISPAD. 76, 82, 252, 271, 272, 279, 288, 292, 294, 302
- Jaraiz, M., Castrillo, P., Pinacho, R., Pelaz, L., Barbolla, J., Gilmer, G. H., and Rafferty, C. S. (2000). Atomistic modeling of complex silicon scenarios. In Agarwall, A., Pelaz, L., Vuong, H.-H., Packan, P., and Kase, M., editors, *Si Front-End Processing- Physics and Technology of Dopant-Defect Interactions II*, volume 610, pages B11.1.1–B11.1.6. Mat. Res. Soc. 274
- Jaraiz, M., Gilmer, G. H., Poate, J. M., and de la Rubia, T. D. (1996). Atomistic calculations of ion implantation in si: Point defect and transient enhanced diffusion phenomena. *Appl. Phys. Lett.*, 68(3):409. 133, 271
- Jaraiz, M., Pelaz, L., Rubio, E., Barbolla, J., Eaglesham, D. J., Gossmann, H. J., and Poate, J. M. (1998). Atomistic modeling of point and extended defects in crystalline materials. volume 532, pages 43–53. Mat. Res. Soc. 252, 258, 279
- Johnson, M. D., Caturla, M. J., and de la Rubia, T. D. (1998). A kinetic monte-carlo study of the effective diffusivity of the silicon self-interstitial in the presence of carbon and boron. *Journal of Applied Physics*, 84(4):1963–7. 206
- Kang, Y.-S. and Ho, P. S. (1997). Thickness dependent mechanical behavior of submicron aluminum films. *Journal of Electronic Materials*, 26(7):805. 61, 420
- Kendall, D. L. (1969). *Semiconductor Silicon*. Electrochem. Soc., NY. USA. 199, 424
- Kennedy, D. P. (1971). In *Proc. IEEE*, volume 59, page 335. I.E.E.E. 199, 424
- Kim, J., Wilkins, J. W., Khan, F. S., and Canning, A. (1997). Extended si—p[311—p] defects. *Phys. Rev. B*, 55(24):16186. 87
- Kim, R., Aoki, T., Furuta, Y., Kobayashi, H., Xia, J., Saito, T., Kamakura, Y., and Taniguchi, K. (2000). Transient enhanced diffusion of arsenic by self-implantation. In *Mat. Res. Soc. Symp. 2000*, volume 610. Mat. Res. Soc. 240, 310, 311, 426, 430
- Kohyama, M. and Takeda, S. (1992). Atomic structure and energy of the {113} planar interstitial defects in si. *Phys. Rev. B*, 46(19):12 305–316. 87
- Kopal, Z. (1961). *Numerical Analysis*. Wiley, New York. 75

- Larsen, K. K., Privitera, V., Coffa, S., Priolo, F., Campisano, S. U., and Carnera, A. (1996). Trap-limited migration of si self-interstitials at room temperature. *Phys. Rev. Lett.*, 76(9):1493–1496. 261
- Law, M. E., Gilmer, G. H., and Jaraiz, M. (2000). Simulation of defects and diffusion phenomena in silicon. *Mat. Res. Soc. Bulletin*, 25(6):45. 92, 421
- Law, M. E., Haddara, Y. M., and Jones, K. S. (1998). Effect of the silicon/oxide interface on interstitials: Di-interstitial recombination. *J. Appl. Phys.*, 84:3555–3560. 252, 253
- Lawther, D. W., Myler, U., Simpson, P. J., Rousseau, P. M., Griffin, P. B., and Plummer, J. D. (1995). Vacancy generation resulting from electrical deactivation of arsenic. *Appl. Phys. Lett.*, 67(24):3575. 306, 308
- Lee, Y. H. and Corbett, J. W. (1974). Epr study of defects in neutron-irradiated silicon: Quenched-in alignment under $\{110\}$ -uniaxial stress. *Phys. Rev. B*, 9(10):4351. 90
- Lenosky, T. J., Sadigh, B., Theiss, S. K., Caturla, M. J., and de la Rubia, T. D. (2000). Ab initio energetics of boron-interstitial clusters in crystalline si. *Applied Physics Letters*, 77(12):1834–6. 195
- Li, J. and Jones, K. S. (1998). $\{311\}$ defects in silicon: The source of the loops. *Appl. Phys. Lett.*, 73(25):3748. 227, 228, 425
- Lim, D. R., Rafferty, C. S., and Klemens, F. P. (1995). The role of the surface in transient enhanced diffusion. *Appl. Phys. Lett.*, 67:2302–2304. 95, 151, 253
- Lindhard, J. and Schaff, M. (1961). Energy dissipation by ions in the kev region. *Phys. Rev.*, 124:128. 75
- Lita, A. E. and Jr., J. E. S. (1998). Characterization of surface structure in sputtered al films: Correlation to microstructure evolution. *J. Appl. Phys.*, 85(2):876. 61
- Lita, A. E. and Jr., J. E. S. (2000). Effects of grain growth on dynamic surface scaling during the deposition of al polycrystalline thin films. *Phys. Rev. B*, 61(11):7692. 61
- Liu, J. (1996). PhD thesis, University of Florida. 118
- Liu, X.-Y., Windl, W., and Masquelier, M. P. (2000). Ab initio modeling of boron clustering in silicon. *Appl. Phys. Lett.*, 77(13):2018. 256
- Man, K. F., Tang, K. S., and Kwong, S. (1999). *Genetic Algorithms*. Springer, london edition. 33

- Marques, L. A. (2003). (private communication). 257
- Marques, L. A., Pelaz, L., Aboy, M., Enriquez, L., and Barbolla, J. (2003). Microscopic description of the irradiation-induced amorphization in silicon. *Phys. Rev. Lett.*, 91:135504. 294
- Marques, L. A., Pelaz, L., Hernandez, J., Barbolla, J., and Gilmer, G. H. (2001). Stability of defects in crystalline silicon and their role in amorphization. *Phys. Rev. B*, 64:045214. 25, 92, 133, 252
- Marsaglia, G. and Zaman, A. (1987). Toward a universal random number generator. Technical report, Florida State University. 52
- Martin-Bragado, I., Jaraiz, M., Castrillo, P., Pinacho, R., Barbolla, J., and de Souza, M. M. (2003). Mobile silicon di-interstitial: Surface, self-interstitial clustering, and transient enhanced diffusion phenomena. *Phys. Rev. B*, 68(19):195204–1/5. 25, 251, 271
- Martin-Bragado, I., Jaraiz, M., Castrillo, P., Pinacho, R., Rubio, J. E., and Barbolla, J. (2004a). Ion implant simulations: Kinetic monte carlo annealing assessment of the dominant features. *Appl. Phys. Lett.*, 84(24):4962. 270
- Martin-Bragado, I., Jaraiz, M., Castrillo, P., Pinacho, R., Rubio, J. E., and Barbolla, J. (2004b). A kinetic monte carlo annealing assessment of the dominant features from ion implant simulations. In *EMRS 2004 conference proceedings*. EMRS. 270
- Martin-Bragado, I., Pinacho, R., Castrillo, P., Jaraiz, M., Rubio, J. E., and Barbolla, J. (2004c). Physical modeling of fermi-level effects for decanano device process simulations. In *EMRS 2004 conference proceedings*. EMRS. 278, 304, 436
- Masters, B. J. and Fairfield, J. M. (1969). Arsenic isoconcentration diffusion studies in silicon. *J. Appl. Phys.*, 40(6):2390. 199, 424
- Mathiot, D. and Pfister, J. C. (1984). Dopant diffusion in silicon: A consistent view involving nonequilibrium defects. *J. Appl. Phys.*, 55(10):3518–3530. 192, 276, 285, 286, 343, 424, 429
- Matsumoto, S., Gibbons, J. F., Deline, V., and Evans, C. A. (1980). High-temperature scanning cw laser-induced diffusion of arsenic and phosphorus in silicon. *Appl. Phys. Lett.*, 37(9):821–4. 199, 424
- Matsumoto, S., Ishikawa, Y., and Niimi, T. (1983). Oxidation enhanced and concentration dependent diffusions of dopants in silicon. *J. Appl. Phys.*, 54(9):5049. 303

- Mogi, T. K., Thomson, M. O., Gossmann, H.-J., Poate, J. M., and Luftman, H. S. (1996). Thermal nitridation enhanced diffusion of sb and si(100) doping superlattices. *Appl. Phys. Lett.*, 69(9):1273. 85
- Moroz, V., Hurat, P., and Pramanik, D. (2003). Atomistic process simulation of decananometer devices. *Synopsys TCAD Times, News and Information on Physical Simulation of Semiconductor Technology*, pages 1–3. 318
- Moroz, V., Strecker, N., and Jaraiz, M. (2002). Exploring methods for adequate simulation of sub-100 nm devices. In Baccarani, G., Gnani, E., and Rudan, M., editors, *Proceedings of the 32nd European Solid state device research Conference 2002*, pages 299–302. 279, 288
- Mueller, D. C., Alonso, E., and Fichtner, W. (2003). Arsenic deactivation in si: Electronic structure and charge states of vacancy-impurity clusters. *Phys. Rev. B*, 68(4):045208. 306
- Nichols, C. S., de Walle, C. G. V., and Pantelides, S. T. (1989). Mechanisms of equilibrium and nonequilibrium diffusion of dopants in silicon. *Phys. Rev. Lett*, 62(9):1049–1052. 304
- Nobili, D., Solmi, S., Parisini, A., Derdour, M., Amigliato, A., and Moro, L. (1994). Precipitation, aggregation, and diffusion in heavily arsenic-doped silicon. *Phys. Rev. B*, 49(4):2477. 306
- Nobili, D., Solmi, S., and Shao, J. (2001). Clustering equilibrium and deactivation kinetics in arsenic doped silicon. *J. Appl Phys.*, 90(1):101–107. 302, 306, 308
- Northrup, J. E. and Zhang, S. B. (1993). Dopant and defect energetics: Si in gaas. *Phys. Rev. B.*, 47(11):6791–6794. 173, 279, 286
- Oen, O. S. and Robinson, M. T. (1976). “computer studies of the reflection of light ions from solids”. *Nucl. Instrum. Methods*, 132:674. 76
- Oh, Y.-S. and Ward, D. E. (1998). A calibrated model for trapping of implanted dopants at material interface during thermal annealing. In *IEDM 1998 proceedings*, 18.6, page 509/512. I.E.D.M. 154
- Ohno, K., Esfarjani, K., and Kawazoe, Y. (1999). *Computational materials science. From Ab Initio to Monte Carlo methods*. Solid-State sciences. Springer. ISBN 3-540-63961-6. 57
- Pandey, K. C., Erbil, A., III, G. S. C., Boehme, R. F., and Vanderbilt, D. (1988). Annealing of heavily arsenic-doped silicon: Electrical deactivation and a new defect complex. *Phys. Rev. Lett.*, 61(11):1282. 306

- Pelaz, L., Gilmer, G. H., Jaraiz, M., Herner, S. B., Gossmann, H.-J., Eaglesham, D. J., Hobler, G., Rafferty, C. S., and Barbolla, J. (1998). Modeling of the ion mass effect on transient enhanced diffusion: Deviation from the +1 model. *App. Phys. Lett.*, 73(10):1421–1423. 271
- Pelaz, L., Gilmer, G. H., Venezia, V. C., Gossmann, H.-J., Jaraiz, M., and Barbolla, J. (1999). Modeling of the effects of dose, dose rate, and implant temperature on transient enhanced diffusion. *App. Phys. Lett.*, 74(14):2017–2019. 195, 279
- Pelaz, L., Jaraiz, M., Gilmer, G. H., Gossmann, H.-J., Rafferty, C. S., Eaglesham, D., and Poate, J. M. (1997). B diffusion and clustering in ion implanted si: The role of b cluster precursors. *App. Phys. Lett.*, 70(17):2285–2287. 98, 195, 236, 237, 261, 271, 276, 426, 428
- Pelaz, L., Marques, L. A., Aboy, M., Barbolla, J., and Gilmer, G. (2003). Atomistic modeling of amorphization and recrystallization in silicon. *Appl. Phys. Lett.*, 82(13):2038. 294
- Persson, C., Lindefelt, U., and Sernelius, B. E. (1999). Band gap narrowing in n-type and p-type 3c-, 2h-, 4h-, 6hsic, and si. *J. App. Phys.*, 86(8):4419–4427. 174, 288
- Pinacho, R., Castrillo, P., Jaraiz, M., Martin-Bragado, I., and Barbolla, J. (2002). Carbon in silicon: Modeling of diffusion and clustering mechanisms. *J. Appl. Phys.*, 92(3):1582. 206, 285
- Pinacho, R., Jaraiz, M., Rubio, J. E., Martin-Bragado, I., and Barbolla, J. (2004). Comprehensive, physically based modelling of as in si. In *E-MRS Spring Meeting Proceedings 2004*. 285, 291, 295, 301
- Plummer, J. D., Deal, M. D., and Griffin, P. B. (2000). *Silicon VLSI Technology: Fundamentals, practice and modeling*. Electronics and VLSI series. Prentice Hall. 4, 192, 285, 286, 424, 429
- Raabe, D. (1998). *Computational Materials Science. The simulation of materials microstructures and properties*. Wiley-VCH. ISBN 3-527-29541-0. 57
- Rafferty, C. (1997). Progress in predicting transient diffusion. In *Internat. Conf. on Simulation of Semicond. Processes and Devices (SISPAD 1997)*. 158, 279
- Rafferty, C., Gilmer, G., Jaraiz, M., Eaglesham, D., and Gossmann, H. (1996). Simulation of cluster evaporation and transient enhanced diffusion in silicon. *Appl. Phys. Lett.*, 68(17):2395. 269
- Ramamoorthy, M. and Pantelides, S. T. (1996). Complex dynamical phenomena in heavily arsenic doped silicon. *Phys. Rev. Lett.*, 76(25):4753. 304, 306

- Ranki, V., Nissilä, J., and Saarinen, K. (2002). Formation of vacancy-impurity complexes by kinetic processes in highly as-doped si. *Phys. Rev. Lett.*, 88(10):105506. 306
- Rao, V. and Rao, H. (1993). *C++ Neural networks and Fuzzy logic*. MIS:Press, 115 West 18th Street. New York, NY 10011. 33
- Rata, I., Shvartsburg, A. A., Horoi, M., Frauenheim, T., Su, K. W. M., and Jackson, K. A. (2000). Single-parent evolution algorithm and the optimization of si clusters. *Phys. Rev. Lett.*, 85(3):546–549. 38
- Robinson, M. T. (1989). Slowing-down time of energetic atoms in solids. *Phys. Rev. B*, 40(16):10717. 272
- Robinson, M. T. (1992). The binary collision approximation: Background and introduction. In Biersack, J. P., editor, *Radiation Effects and Defects in Solids. Proceedings of the Interational Conference on Computer simulations of Radiation Effects in Solids, COSIRES 1992*, volume 130-131, pages 3–20. 56, 76
- Robinson, M. T. and Torrens, I. M. (1974). Computer simulation of atomic-displacement cascades in solids in the binary-collision approximation. *Phys. Rev. B*, 9:5008. 57, 73, 76, 219, 258
- Roorda, S. and Sinke, W. C. (1989). Rapid nucleation in pulsed laser heated amorphous si. *Appl. Surf. Sci.*, 36:588. 94
- Rousseau, P. M., Griffin, P. B., Fang, W. T., and Plummer, J. D. (1998). Arsenic deactivation enhanced diffusion: A time, temperature and concentration study. *J. Appl. Phys.*, 84(7):3593. 238, 302, 306, 307, 308, 309, 426, 430
- Rubin, M., Saha, S., Lutze, J., Nouri, F., Scott, G., and Pramanik, D. (1999). Interface interstitial recombination rate and the reverse short channel effect. volume 568, pages 213–218. *Mat. Res. Soc.* 257
- Rubio, J. E., Jaraiz, M., Martin-Bragado, I., Hernandez-Mangas, J. M., Barbolla, J., and Gilmer, G. H. (2003). Atomistic monte carlo simulations of three-dimensional polycrystalline thin films. *J. App. Phys.*, 94(1):163–168. 61, 63, 68, 420
- Rubio, J. E., Jaraiz, M., Martin-Bragado, I., Pinacho, R., Castrillo, P., and Barbolla, J. (2004). Physically based modelling of damage, amorphization, and recrystallization for predictive device-size process simulation. In *European MRS conference proceedings*. European MRS. 291, 302, 310
- Sadigh, B., Lenosky, T. J., Caturla, M.-J., Quong, A. A., Benedict, L. X., de la Rubia, T. D., Giles, M. M., Foad, M., Spataru, C. D., and Louie, S. G. (2002).

- Large enhancement of boron solubility in silicon due to biaxial stress. *Appl. Phys. Lett.*, 80(25):4738–4740. 173
- Sadigh, B., Lenosky, T. J., Theiss, S. K., Caturla, M. J., de la Rubia, T. D., and Foad, M. A. (1999). Mechanism of boron diffusion in silicon: an ab initio and kinetic monte carlo study. *Physical Review Letters*, 83(21):4341–4. 283
- Schultz, P. J., Jagadish, C., Ridgway, M. C., Elliman, R. G., and Williams, J. S. (1991). *Phys. Rev. B*, 44:9118. 93, 295, 296, 421, 430
- Seeger, A. and Cheek, K. P. (1968). *Phys. Status Solidi*, 29:455. 83
- Several (2003). International technology roadmap for semiconductors. <http://public.itrs.net>. 6, 320
- Skarlatos, D., Onri, M., Claverie, A., and Tsoukalas, D. (1999). *J. Electrochem. Soc.*, 146:2276. 96, 421
- Solmi, S. and Nobili, D. (1998). High concentration diffusivity and clustering of arsenic and phosphorus in silicon. *J. Appl. Phys.*, 83(5):2484. 306
- Stillinger, F. H. and Weber, T. A. (1985). Computer simulation of local order in condensed phases of silicon. *Phys. Rev. B*, 31(8):5262–5271. 16, 17, 18, 38
- Stolk, P. A., Gossmann, H. J., Eaglesham, D. J., Jacobson, D. C., Poate, J. M., and Luftman, H. S. (1995). Trap-limited interstitial diffusion and enhanced boron clustering in silicon. *Appl. Phys. Lett.*, 66(5):568. 98
- Stolk, P. A., Gossmann, H. J., Eaglesham, D. J., Jacobson, D. C., Rafferty, C. S., Gilmer, G. H., Jaraiz, M., Poate, J. M., Luftman, H. S., and Haynes, T. E. (1997). Physical mechanism of transient enhanced dopant diffusion in ion-implanted silicon. *J. Appl. Phys.*, 81:6031–6050. 88, 97, 103, 225, 226, 260, 271, 421, 425
- Sützle, T. and Hoos, H. H. (2000). Max-min ant system. *Future Generation Computer Systems*, 16:889–914. 33
- Synopsys (2003). *Taurus reference Manual*. Mountain View, CA,USA, 2003.1 edition. 279, 288
- Sze, S. M. (1981). *Physics of semiconductor devices*. John Wiley & Sons, Inc., New York, 2 edition. 160, 281, 288, 436
- Sze, S. M., editor (1983). *VLSI technology*. Series in Electrical Engineering. McGraw-Hill. 4
- Takahashi, A., Soneda, N., Ishino, S., and Yagawa, G. (2003). Structures and energies of vacancy-cu clusters in α -fe. *Phys. Rev. B*, 67(2):024104. 271

- Takeda, S. (1991). An atomic model of electron-irradiation-induced defects on {113} in si. *Jap. J. Appl. Phys.*, 30:L639. 87, 118
- Tan, T. Y. and Gössele, U. (1985). *Appl. Phys. A: Solids Surf.*, 37:1. 85
- Tang, M., Colombo, L., Zhu, J., and de la Rubia, T. D. (1997). Intrinsic point defects in crystalline silicon: Tight-binding molecular dynamics studies of self-diffusion, interstitial-vacancy recombination, and formation volumes. *Phys. Rev. B*, 55(21):14279. 85, 294
- Theiss, S. K., Caturla, M. J., de la Rubia, T. D., Johnson, M. C., Ural, A., and Griffin, P. B. (1999). Linking ab initio energetics to experiment: kinetic monte carlo simulation of transient enhanced diffusion of b in si. In Bulatov, V. V., de la Rubia, T. D., Phillips, R., Kaxiras, E., and Ghoniem, N., editors, *Multiscale-Modelling-of-Materials.*, pages 291–5. Mater. Res. Soc. 320
- Thompson, C. V. (2000). Structure evolution during processing of polycrystalline films. *Annu. Rev. Mater. Sci.*, 30:159–190. 61
- Ueki, T., Itsumi, M., and Takeda, T. (1997). Octahedral void defects observed in the bulk of czochralski silicon. *Appl. Phys. Lett.*, 70(10):1248. 90
- Venezia, V. C., Duffy, R., Pelaz, L., Aboy, M., Heringa, A., Griffin, P. B., Wang, C. C., Hopstaken, M. J. P., Tamminga, Y., Dao, T., Pawlak, B. J., and Roozeboom, F. (2003). Dopant redistribution effects in preamorphized silicon during low temperature annealing. In *IEDM 2003 proceedings*. IEDM. 137
- Voter, A. F. (1997). Hyperdynamics: Accelerated molecular dynamics of infrequent events. *Phys. Rev. Lett.*, 78(20):3908. 92
- Wall, M. (1996). Galib: A c++ library of genetic algorithm components. <http://lancet.mit.edu/ga/>. Massachusetts institute of technology. 38
- Watkins, G. D. (1975). Defects in irradiated silicon: Epr and electron-nuclear double resonance of interstitial boron. *Phys. Rev. B*, 12(12):5824–39. 285
- Watkins, G. D. (1992). *Deep centers in semiconductors*. Switzerland, 2 edition. 90
- Watkins, G. D. (2000). Intrinsic defects in silicon. *Materials in Semiconductor Processing*, 3:227–235. 254, 281, 283
- Watkins, G. D. and Corbett, J. W. (1965). Defects in irradiated silicon: Electron paramagnetic resonance of the divacancy. *Phys. Rev.*, 138(2A):A543. 90
- Watkins, G. D., Troxell, J. R., and Chatterjee, A. P. (1979). *Inst. Phys. Conf. Serf.*, 46(16). 84, 85, 421

- Windl, W., Bunea, M. M., Stumpf, R., Dunham, S. T., and Masquelier, M. P. (1999). First-principles study of boron diffusion in silicon. *Phys. Rev. Lett.*, 83(21):4345. 283
- Yu, M., Huang, R., Zhang, X., Wang, Y., and Oka, H. (2003). Atomistic simulation of rta annealing for shallow junction formation characterizing both bed and ted. In Iwai, H., editor, *IEICE Transactions of Electronics*, volume E86-C, pages 295–300. SISPAD, Electronic Society. 320
- Zhang, L. H., Jones, K. S., Chi, P. H., and Simons, D. S. (1995). Transient enhanced diffusion without $\{311\}$ defects in low energy b^+ -implanted silicon. *Appl. Phys. Lett.*, 67:2025–2027. 252
- Zhu, J. (1998). Ab initio pseudopotential calculations of dopant diffusion in si. *Computational Materials Science*, 12:309–318. 281
- Zimmermann, H. and Ryssel, H. (1992). *Appl. Phys. A: Solids Surf.*, 55:121. 85

Índice de figuras

| | |
|--|----|
| 1.1. Pasos de fabricación de un transistor NMOS. Su descripción se encuentra en el texto. | 5 |
| 2.1. Diferentes métodos de simulación, teniendo en cuenta sus rangos de tiempos típicos y su ámbito determinista o estocástico. | 13 |
| 2.2. Un integrador resuelve las ecuaciones de movimiento calculando el paso siguiente en función de la trayectoria pasada y las fuerzas en el momento presente (y/o pasado). | 15 |
| 2.3. Contribución a dos cuerpos del potencial de Stillinger–Weber. . . | 18 |
| 2.4. Estructura interna de <code>DynAtomSet</code> | 21 |
| 2.5. Diferentes tipos información atómica en UVAS. | 22 |
| 2.6. Forma en que se divide el espacio en el <code>Locator</code> para facilitar la búsqueda de vecinos. | 25 |
| 2.7. Un ad-atom, que fue un intersticial que se dirigió hacia la superficie y fue atrapado por ésta. | 26 |
| 2.8. Proyección bidimensional de átomos (cuadrados pequeños), en la que se muestra el di-intersticial y su entorno (cuadrados grandes) marchando hacia la superficie y siendo recombinado por ésta. . . . | 28 |
| 2.9. Evolución temporal de la energía de los átomos frente al eje x . El I_2 está rodeado por un conjunto de átomos bastante energéticos, que se van moviendo hacia la izquierda y terminan desapareciendo en la superficie. | 29 |
| 3.1. Los algoritmos genéticos seleccionan los mejores individuos de una población después de que se han permitido cruces y mutaciones entre sus miembros. | 34 |
| 3.2. Mutación especular: Una vez que se ha calculado donde está el centro de masas del cluster, se borran la mitad de los átomos y se reconstruyen como su reflejo respecto al punto centro de masas. . | 39 |

| | | |
|------|--|----|
| 3.3. | Para cruzar dos clusters se genera aleatoriamente un plano que divida los clusters padre en dos partes, y posteriormente se crea la descendencia como la mitad de cada padre. A veces, es necesario desplazar el plano para que las dos mitades sumen el número adecuado de átomos. | 40 |
| 3.4. | El operador de reemplazo intenta mantener una cierta variabilidad en la población, asegurándose una diferencia mínima de ΔE entre todos los individuos de ésta. | 41 |
| 3.5. | Clusters aislados de tamaño $n = 4, 5, 6, 7, 8, 9, 10$ y 14 usando un potencial Lennard–Jones. | 43 |
| 3.6. | Clusters aislados de tamaños $n = 4, 5, 6, 7, 8, 9, 10$ y 14 usando un potencial de Stillinger–Weber. | 44 |
| 3.7. | Variación de la energía durante las distintas generaciones del proceso de minimización por algoritmos genéticos para un cluster aislado de tamaño 4 usando el potencial Stillinger–Weber. | 45 |
| 5.1. | Microfotografía de una capa de aluminio de 120 nm tomada usando la técnica “Bright field TEM”. Referencia (Kang and Ho, 1997). | 61 |
| 5.2. | Energía de enlace para átomos de aluminio en una red FCC, dependiendo de su número de coordinación. Referencia (Huang et al., 1998). | 63 |
| 5.3. | Diferentes granos, simulados con un código completamente tridimensional de tipo Lattice Monte Carlo atómico. (Rubio et al., 2003). | 63 |
| 5.4. | Cerca de las fronteras de grano aparecen varios destinos posibles para los átomos que están saltando, de los cuales alguno puede pertenecer a orientaciones de red distintas. Se permite el salto a estos sitios siempre que se encuentren suficientemente cerca, es decir, a una distancia de primeros vecinos o menor. | 64 |
| 5.5. | Formación de la red y clasificación de los átomos según sus posiciones. | 66 |
| 5.6. | Proyección tridimensional de un cristal antes y después del recocido. Los átomos rojos han sido movidos en los pasos anteriores. | 68 |
| 5.7. | Los átomos que están situados en bordes agudos tienen una probabilidad más alta de moverse, de tal manera que estos fenómenos colectivos de migración van llevando a la formación de caras cristalográficas. | 69 |
| 5.8. | Simulación del depósito de 1×10^3 , 2×10^3 y 5×10^4 átomos de aluminio y posterior recocido a -73°C . Referencia (Huang et al., 1998). | 70 |
| 5.9. | Cubo de aluminio recocido a 700°C y enfriado después hasta -73°C . Simulación realizada con el módulo LMC del simulador UVAS. | 70 |
| 6.1. | Seguimiento de los proyectiles tras las colisiones. | 73 |

| | |
|---|----|
| 6.2. Colisión binaria (referencia desde el centro de masas). | 74 |
| 7.1. Algunas configuraciones para (a) una vacante, (b) un intersticial propio y, (c) una impureza intersticial. Tomado de (Watkins et al., 1979). | 84 |
| 7.2. Energía de disociación para pequeños clusters de intersticiales propios. Tomado de (Cowern et al., 1999c). | 87 |
| 7.3. Corte de una vista por microscopía TEM de un defecto extendido de intersticiales tipo $\{311\}$. El defecto es como una cinta alargada. Ref. (Eaglesham et al., 1994; Stolk et al., 1997). | 88 |
| 7.4. Vista por microscopía TEM de los discos de dislocación. Tomado de la referencia (Cristiano et al., 2000). | 89 |
| 7.5. Energías de formación para los discos de dislocación imperfectos (faulted FDL) y perfectos (PDL). Tomado de (Colombeau et al., 2001). | 89 |
| 7.6. Vista de los voids (huecos) por microscopía TEM. Ref. (Holland and White, 1991). | 90 |
| 7.7. Energía de enlace de clusters de vacantes en función de su tamaño, calculada con "Tight-Binding". Ref. (Bongiorno and Colombo, 1998). | 91 |
| 7.8. Simulación por dinámica molecular de una serie de cascadas de iones de Si a 5 keV después de un recocido a alta temperatura. a) 0.1 ps, b) 2.5 ps y c) 9 ps. Hay $1,6 \cdot 10^5$ átomos en la simulación, pero solamente los átomos con una energía potencial más grande que 0.2 eV están dibujados. Tomado de la referencia (Law et al., 2000). | 92 |
| 7.9. Espectro RBS de acanalamiento para implantaciones de Si en Si con 1 MeV de energía, dosis de 10^{15} cm^{-2} y las diferentes temperaturas de sustrato indicadas en la figura. Hay una dependencia crítica, que está cerca de la temperatura ambiente. Tomado de (Schultz et al., 1991). | 93 |
| 7.10. Patrones de dañado de distinto tipo, que han surgido de la implantación de dos iones, con casi la misma masa (a la izquierda As y a la derecha Ge), y que sugieren que los efectos de tipo químico o eléctrico pueden jugar de un papel central a la hora de dilucidar la morfología de los procesos de dañado. Tomado de la referencia (Holland et al., 1996). | 94 |
| 7.11. Bucles de dislocación después de a) 90 min. de recocido a 900°C en un ambiente inerte y b) en un ambiente oxidante. La comparación tanto del número como del tamaño de los bucles permite hacer una estimación de la frecuencia de emisión de I_s desde la superficie oxidante a la temperatura dada. Tomado de la referencia (Skarlatos et al., 1999). | 96 |

| | |
|---|-----|
| 7.12. Esquema de la formación, migración y ruptura de un defecto puntual emparejado (mecanismo kick-out). | 97 |
| 7.13. Región de simulación de drenador y fuente para un transistor NMOSFET durante un recocido. | 99 |
| 8.1. En la técnica kMC no se incluyen los átomos de la red, sino que tan sólo los átomos pertenecientes a los defectos son tomados en cuenta. | 103 |
| 8.2. DADOS divide el espacio en pequeñas cajas, dividiendo cada eje por un número que es potencia de 2. | 106 |
| 8.3. Al tomarse el tipo de material en pequeñas cajas rectangulares individuales, las intercaras entre los materiales pueden presentar una estructura escalonada. | 107 |
| 8.4. Un algoritmo que selecciona un evento de manera aleatoria, pero de acuerdo a la frecuencia con que cada uno de ellos debe suceder, dependiendo de su configuración atómica instantánea. | 109 |
| 8.5. Un defecto puntual tiene tres coordenadas, junto con dos números para especificar el tipo de <i>partícula</i> y <i>defecto</i> . Se mueve una distancia λ en direcciones ortogonales y puede interactuar con otras partículas siempre que se encuentre en su radio de captura. | 110 |
| 8.6. Los clusters son conglomerados desordenados de partículas I o V . Pueden crecer atrapando partículas, o bien disolverse por emisión o recombinación. | 115 |
| 8.7. Pequeños clusters de intersticiales y vacantes simulados con DADOS. | 115 |
| 8.8. Energía necesaria para emitir una partícula desde un cluster de tamaño n | 117 |
| 8.9. Valores concretos y continuos de las energías de enlace para clusters de intersticiales. | 117 |
| 8.10. Valores concretos y continuos de las energías de enlace para clusters de vacantes. | 117 |
| 8.11. Los defectos de tipo $\{311\}$ son cadenas de intersticiales orientadas. Estos defectos pueden atrapar y emitir I así como recombinarse con vacantes. | 119 |
| 8.12. $\{311\}$ s extraídos de una simulación de DADOS. También hay presentes dos pequeños bucles de dislocación. Las esferas rojas son intersticiales de Si, las amarillas átomos de As. | 119 |
| 8.13. La anchura de un $\{311\}$ cambia con el tamaño. | 120 |
| 8.14. Los bucles de dislocación son discos planos de intersticiales. Crecen atrapando nuevos I s y se disuelven a través de la emisión de dichos I s o recombiándose con V s. | 122 |
| 8.15. Bucles de dislocación en una simulación de DADOS, junto con algunos $\{311\}$ que también pueden verse en la imagen. Las esferas rojas son I y las amarillas As. | 123 |

- 8.16. Energías de enlace de I a una dislocación, en función de su tamaño. 124
- 8.17. Los voids son aglomeraciones con forma esferoidal de vacantes.
Pueden atrapar y emitir Vs , y recombinarse con Is 125
- 8.18. Los pockets amorfos son aglomeraciones desordenadas de partículas con una cierta barrera (generalmente baja) para su recombinación. Crecen con la captura de Is y Vs y disminuyen recombinando pares IV 128
- 8.19. Pockets amorfos creados en una simulación de DADOS después de implantar una cascada de arsénico. Esferas rojas: Intersticiales. Verdes: Vacantes. Amarillas: Arsénico. 128
- 8.20. Energía de recombinación de un par IV dentro de un pocket amorfo, en función del tamaño, ver texto. 130
- 8.21. Una vez que no es posible recombinar más pares IV , el pocket amorfo se disuelve y forma pequeños clusters. 131
- 8.22. Esquema de los perfiles de dañado para tiempos $t_1 < t_2$, utilizado para explicar el modelo semi-atomístico de amorfización. 132
- 8.23. Evolución del drenador y canal de un transistor mientras recristaliza. Se ha implantado As a 35 keV y con dosis $1 \times 10^{15} \text{ cm}^{-2}$, haciendo a continuación un recocido a 600 °C. Se ve la evolución a $t = 0, 5, 20, 40, 60, 90$ y 100 s mientras recristaliza. 135
- 8.24. Las impurezas difunden saltando una distancia λ en direcciones ortogonales. A su vez pueden romperse en varias partículas o cambiar a otras configuraciones. 139
- 8.25. Energías relacionadas con la reacción $A_i \rightarrow A + I$ 142
- 8.26. Los clusters de impurezas son aglomeraciones desordenadas de dopantes junto con Is (o Vs). DADOS almacena separadamente los Is (Vs) de las impurezas en estos clusters. Los mecanismos de crecimiento y decrecimiento se relacionan con la captura de nuevas partículas, la emisión de éstas, la recombinación, y el mecanismo de Frank-Turnbull. 144
- 8.27. Emisión Frank-Turnbull: El cluster emite un intersticial generando también una vacante. 147
- 8.28. Emisión complementaria: El cluster genera un par IV , almacena la vacante y emite el intersticial junto con una impureza. 149
- 8.29. Las intercaras de DADOS son el conjunto de caras entre cajas con materiales distintos. 151
- 8.30. La longitud de recombinación (L_r) sirve para modelizar cómo de buena es una intercara atrapando defectos puntuales. L_r es la distancia entre la intercara y el punto en el cual la prolongación de la concentración de los defectos puntuales se encontraría con su valor en equilibrio. 152

| | |
|---|-----|
| 8.31. El flujo de I que se recombina en la superficie será el número de intersticiales contenidos en una distancia de salto de la intercara ($[I] \times \lambda$) que salten a la superficie por segundo ($1/6 \times \nu_m$) cuando la probabilidad de recombinación es 1.) | 153 |
| 8.32. Concentración de átomos de boro sustitucional frente a su máxima concentración en estado estacionario y siguiendo el modelo de Oh-Ward. | 157 |
| 8.33. Energía de formación de I^+ | 159 |
| 8.34. Equilibrio entre diferentes estados de carga de I_s y V_s a 600°C de temperatura. | 162 |
| 8.35. La concentración de la especie cargadas varía con la concentración electrónica. | 164 |
| 8.36. Barreras de migración para un I^+ | 166 |
| 8.37. Energías de formación, migración y activación del boro dependiendo del nivel de Fermi. | 172 |
| 8.38. Relación entre los saltos de difusión (λ) y los saltos largos (Λ) | 172 |
| 8.39. Forma habitual de medir la solubilidad (extrínseca) de un dopante: $[A]_{\text{max}}^{\text{extrinsic}}$ | 173 |
| 8.40. La anchura del gap depende de la concentración local de electrones. | 175 |
| 8.41. Modelo de DADOS para las rampas de temperatura: Pocos pasos largos a temperaturas bajas y muchos cortos a temperaturas altas. | 179 |
| | |
| 9.1. Difusión intrínseca del boro. Líneas: Valores experimentales para distintas referencias (Plummer et al., 2000; Fair, 1981a; Mathiot and Pfister, 1984; Hill, 1981; Haddara et al., 2000). Símbolos: Valores teóricos calculados con los parámetros de entrada de DADOS. | 192 |
| 9.2. Difusión extrínseca del boro. Líneas: Valores teóricos calculados con los parámetros de entrada de DADOS. Símbolos: Valores experimentales tomados de la referencia (Fair, 1981a). | 193 |
| 9.3. Saltos largos del boro (Λ). Líneas: Valores teóricos. Símbolos: Valores experimentales tomados de (Cowern et al., 1991; Cowern et al., 1992). | 195 |
| 9.4. Energías potenciales para los clusters de boro con intersticiales. | 196 |
| 9.5. Difusión intrínseca del arsénico. Líneas: valores experimentales tomados de las referencias (Fair, 1981a; Ghoshtagore, 1971; Kennedy, 1971; Cerofolini et al., 1986; Matsumoto et al., 1980; Kendall, 1969; Ishikawa et al., 1982; Masters and Fairfield, 1969; Hill, 1981). Símbolos: valores teóricos calculados con los parámetros de entrada de DADOS. | 199 |
| 9.6. Energías potenciales para los clusters de arsénico con vacantes ($As_n V_m$). | 201 |
| 9.7. Energías potenciales para clusters de flúor con vacantes ($F_n V_m$). | 204 |

| | |
|--|-----|
| 9.8. Energías potenciales de los clusters de impureza del carbono con I_s ($C_n I_m$). | 206 |
| 9.9. Energías potenciales para clusters de fósforo con vacantes ($P_n V_m$). | 211 |
| 9.10. Energías potenciales para clusters de oxígeno con vacantes $O_n V_m$ | 213 |
| 10.1. Concentración de intersticiales y vacantes en equilibrio a 600 °C. Líneas: valores teóricos. Símbolos: resultados simulados. | 220 |
| 10.2. Productos $D(V)[V]$ y $D(I)[I]$ a diferentes temperaturas. Símbolos: datos experimentales (Bracht et al., 1995; Cowern et al., 1999b; Giese et al., 2000). Líneas: simulaciones. | 221 |
| 10.3. Difusión de una punta de arsénico (régimen intrínseco). 100 s. a 1000 °C. Símbolos: simulador de continuo. Líneas: resultados de DADOS. | 222 |
| 10.4. Difusión intrínseca del boro. 50 s. a 1000 °C. Símbolos: resultados simulados con un simulador de continuo. Líneas: valores generados con DADOS. | 223 |
| 10.5. Difusión de una punta de boro a 900 °C durante 300 s. Símbolos: valores experimentales antes y después del recocido (Cowern et al., 1991). Líneas: simulación de DADOS. | 224 |
| 10.6. Evolución de la supersaturación de I_s con el tiempo. Se ha realizado una implantación de Si en Si de $2 \times 10^{13} \text{ cm}^{-2}$, 40 keV seguida de 100000 s. a 600 °C, 10000 s. a 700 °C o 2000 s a 800 °C. Símbolos: resultados experimentales (Cowern et al., 1999c). Líneas: simulaciones. | 225 |
| 10.7. Disolución de los defectos tipo $\{311\}$. Se hace una implantación de Si en Si de $5 \times 10^{13} \text{ cm}^{-2}$, 40 keV y unos recocidos posteriores de 200000 s a 670 °C, 100000 s a 705 °C, 5000 s a 738 °C y 400 s a 815 °C. Líneas: simulación. Símbolos: resultados experimentales tomados de (Stolk et al., 1997). | 226 |
| 10.8. Resultados experimentales y simulaciones realizadas para una implantación de Si en Si de 100 keV, $2 \cdot 10^{14} \text{ cm}^{-2}$ y un recocido posterior a 800 °C. Se muestra la evolución de los $\{311\}$ s y los defectos en función del tiempo, donde el decaimiento de los $\{311\}$ s se debe a la transformación y posterior absorción por parte de los bucles de dislocación. Líneas: simulación. Símbolos: datos experimentales divididos por un factor de corrección, como se explica en el texto, tomados de (Li and Jones, 1998). | 228 |
| 10.9. Concentración máxima de pares IV en función de la temperatura para distintas velocidades de implantación (en $\text{cm}^{-2}\text{s}^{-1}$) de Si en Si, 80 keV de energía y una dosis de 10^{15} cm^{-2} . Líneas: ajustes a la ecuación 10.1. Símbolos: resultados de DADOS. | 229 |

| | | |
|-------|--|-----|
| 10.10 | Concentración máxima de pares IV en función de la temperatura para distintas velocidades de implantación (en $\text{cm}^{-2}\text{s}^{-1}$) de Ge en Si, 80 keV de energía y una dosis de 10^{15}cm^{-2} . Líneas: ajustes a la ecuación 10.1. Símbolos: resultados de DADOS. | 230 |
| 10.11 | Temperatura de transición (T_C) en función de la dosis de Si y Ge implantados en Si. Líneas: simulación. Símbolos: resultados experimentales tomados de (Goldberg et al., 1995). | 231 |
| 10.12 | Evolución de los defectos puntuales I y V con el tiempo (proyección bidimensional en los ejes x,y). 16 s de recocido a 600°C tras leer una concentración de dañado de $1 \times 10^{22} \text{cm}^{-3}$. Como esta concentración se encuentra por debajo del umbral de amorfización, no queda una zona recristalizada limpia, sin clusters. Cruces (+): clusters. Aspas (\times): pockets amorfos. | 233 |
| 10.13 | Evolución de los defectos puntuales I y V con el tiempo (proyección bidimensional en los ejes x,y). 100 s de recocido a 600°C tras leer una concentración de dañado de $2 \times 10^{22} \text{cm}^{-3}$. Al estar esta concentración por encima del umbral de dañado se produce un frente de recristalización que va limpiando dicho dañado. Cruces (+): clusters. Aspas (\times): pockets amorfos. | 234 |
| 10.14 | Perfil de arsénico antes y después del paso del frente de recristalización. El ejemplo corresponde a una simulación en la que se ha implantado 2 keV de arsénico, con dosis 10^{15}cm^{-2} sobre silicio preamorfizado, y a la que se sometió después a un recocido a 700°C durante 2 horas. | 235 |
| 10.15 | Perfiles simulados y experimentales de una <i>spike</i> de boro con concentración $4,5 \times 10^{19} \text{cm}^{-3}$ tras la implantación de Si en Si a 40 keV de energía y $9 \times 10^{13} \text{cm}^{-2}$ de concentración y el recocido durante 5, 50, 500 y 2100 s a 800°C . Línea continua: datos experimentales tomados de (Pelaz et al., 1997). Línea discontinua: simulaciones. | 237 |
| 10.16 | Recocido durante 480 s a 750°C de una capa enterrada de boro (concentración $1 \times 10^{18} \text{cm}^{-3}$). La difusión del boro es incrementada mediante arsénico implantado y derretido por láser ($4,5 \times 10^{20} \text{cm}^{-3}$ de concentración de As desde 0 hasta 200 nm). Líneas: simulación. Símbolos: Resultados experimentales de (Rousseau et al., 1998) | 238 |
| 10.17 | Perfiles simulados y experimentales para una implantación de boro de 40 keV de energía y $3 \times 10^{14} \text{cm}^{-2}$ de concentración, tras un recocido de 1 hora a 750°C . Los símbolos son los resultados SIMS experimentales (Huang et al., 1997), y las líneas los procedentes de la simulación. | 239 |
| 10.18 | Implantación de As con $5 \times 10^{14} \text{cm}^{-2}$ de dosis y 30 keV de energía, seguido de un recocido de 5 horas a 720°C . Símbolos: resultados experimentales de (Kim et al., 2000). Líneas: Simulación. | 240 |

| | | |
|-------|---|-----|
| 10.19 | TED tras la implantación de B a 3 keV de energía y un recocido de 1 s a 900 °C. El boro difunde según la distancia de los defectos EOR creados previamente a 80 o 180 nm. Símbolos: valores experimentales de (Cowerm et al., 1999a). Líneas: simulaciones. . . | 242 |
| 10.20 | Proceso de in-diffusion del arsénico a 600, 800 y 1000 °C. Líneas: simulación. Símbolos: solubilidad. | 243 |
| 10.21 | In-diffusion del boro a 600, 800 y 1000 °C. Líneas: simulación. Símbolos: solubilidad. | 244 |
| 10.22 | Difusión extrínseca del arsénico usando condiciones de Maxwell-Boltzmann a distintos tiempos. Líneas: simulación. Símbolos: valores teóricos. | 245 |
| 10.23 | Difusión extrínseca del boro para distintos tiempos, usando condiciones de Maxwell-Boltzmann. Líneas: simulación. Símbolos: valores teóricos. | 245 |
| 10.24 | Redistribución de dopantes en un unión <i>PN</i> . Se implantó arsénico con una concentración de $2 \times 10^{15} \text{ cm}^{-2}$ con 50 keV de energía, que fue posteriormente recocido a 750 °C durante 2 horas. Símbolos: datos experimentales de (Chao et al., 1997). Líneas: simulación. . | 247 |
| 11.1. | Perfiles de profundidad del boro antes y después del recocido de una implantación superficial de boro en silicio preamorfizado con Ge, tomado de la Ref. (Cowerm et al., 1999a). Línea continua: perfil de implantación de boro a 3 keV. Símbolos: perfiles SIMS después de 1s de recocido a 900 °C. Líneas punteadas: perfiles simulados de intersticiales (suponiendo que no se recombinan en la superficie) y di-intersticiales según este trabajo, (escala lineal, unidades arbitrarias). | 254 |
| 11.2. | Diagrama de energía que muestra los diferentes procesos de emisión y captura que están involucrados en la formación de un BI_2 | 255 |
| 11.3. | Diagrama que muestra las energías involucradas en la emisión y captura de I_1 e I_2 desde los clusters. | 255 |
| 11.4. | Supersaturación de intersticiales en función del tiempo para tres temperaturas de recocido distintas, después de una implantación de Si de 40 keV, $2 \times 10^{13} \text{ cm}^{-2}$. Los símbolos representan los valores experimentales, que han sido tomados de la Ref. (Cowerm et al., 1999c). Las líneas son resultado de las simulaciones, usando el modelo presentado en esta trabajo, el cual incluye la movilidad del I_2 | 258 |
| 11.5. | Energías de enlace para los clusters de intersticiales, $E_b^I(I_n)$, en función de sus tamaños. Cuadrados: modelo anterior, que sólo utilizada el I móvil. (Cowerm et al., 1999c) Triángulos: modelo actual. | 259 |

- 11.6. Concentración de clusters de intersticiales, después de una implantación de Si de 40 keV, $5 \times 10^{13} \text{ cm}^{-2}$, en función del tiempo y temperatura de recocido. Los símbolos son los datos experimentales extraídos de la referencia (Eaglesham et al., 1994), las líneas representan los valores simulados, utilizado el modelo presentado aquí, que incluye la movilidad del I_2 260
- 11.7. Emisión y captura de un I_2 264
- 11.8. Evolución temporal de la supersaturación durante el recocido a 738°C de una implantación de silicio en silicio de 40 keV, $5 \times 10^{13} \text{ cm}^{-2}$. La componente del ion se corresponde con el ion implantado (modelo “+1”) y FP con el dañado (FPs) generado por el ion. Líneas continuas: evolución obtenida leyendo las coordenadas de la implantación BCA. Líneas punteadas: evolución obtenida leyendo los perfiles implantados y generando coordenadas aleatorias. 273
- 11.9. Línea punteada: perfil $I - V$ muy ruidoso proveniente de una simulación de implantación BCA pequeña (≈ 110 cascadas) de silicio en silicio 40 keV, $5 \times 10^{13} \text{ cm}^2$. Símbolos: el mismo perfil para 14500 cascadas. Figura insertada: la supersaturación para el muestreo de $I - V$ pequeño (línea punteada), para el largo (símbolos) y para los perfiles completos de I, V , es prácticamente la misma. 275
- 11.10 Perfiles simulados y experimentales de una *spike* de boro de $4,5 \times 10^{19} \text{ cm}^{-3}$ después de una implantación de silicio de 40 keV, $9 \times 10^{13} \text{ cm}^{-2}$ y 50 s de recocido a 800°C . Línea continua: datos experimentales (Pelaz et al., 1997). Línea discontinua: simulaciones hechas con los perfiles complejos de I, V o sólo con los perfiles netos de $I - V$. Hay que destacar que los perfiles netos de $I - V$ no aportan partículas suficientes para inmovilizar la *spike* de boro. 276
- 11.11 Energías de formación (E_f , líneas punteadas) y de difusión ($E_{dif} = E_f + E_m$, líneas continuas) de la vacante de silicio para distintas cargas (V^j) en función del nivel de Fermi (e_F). Se han usado los parámetros listados en la tabla 11.1. Se mide e_F en el borde de la banda de valencia, y se toma $E_g = 1.17 \text{ eV}$ como gap para una temperatura $T = 0 \text{ K}$. Los valores mínimos para $E_f(V^j)$ y $E_{dif}(V^j)$ están dibujados con líneas gruesas. E_{dif}^{int} se corresponde con la energía de difusión para el silicio en condiciones intrínsecas ($e_F = e_i$) y los círculos indican los cruces de energías correspondientes a los distintos niveles de carga e_{V^j} 280

11.12 Líneas discontinuas: energía de formación de B_i^j a partir de B^- ($= E_f(B_i^j) + E_f(B^-)$) en función del nivel de Fermi y usando los valores de la tabla 11.1. Línea continua: energía de activación de la contribución de B_i^j a la difusividad efectiva del boro en condiciones de equilibrio, E_{dif} (ver ecuación. 11.34). Las energías de migración ($E_m(B_i^j)$) son las diferencias entre las líneas continuas y discontinuas. El punto indica el valor de E_{dif}^{int} y el resto de notación es como la de la figura 11.11. 284

11.13 Difusividad efectiva del boro en condiciones de equilibrio, D_B^* . Los valores obtenidos con nuestro conjunto de parámetros están comparados con ajustes experimentales para materiales intrínsecos ((Refs. (Fair, 1981b; Mathiot and Pfister, 1984; Plummer et al., 2000) y con valores experimentales para materiales dopados tipo n (Fair, 1981b) ($N_D = 1,5 \times 10^{20} \text{ cm}^{-3}$). 286

11.14 *Plot* de Arrhenius para los saltos largos del B_i (Λ). Líneas: valores calculados usando nuestro conjunto de parámetros para materiales intrínsecos y silicio dopado n y p . Símbolos: valores experimentales en condiciones casi-intrínsecas. (Cowern et al., 1991; Cowern et al., 1992). 287

11.15 Perfiles de concentración de arsénico normalizados en condiciones intrínsecas y extrínsecas para distintos tiempos a 800°C . Línea discontinua: difusión intrínseca calculada teóricamente. Símbolos huecos: perfiles simulados mediante kMC y normalizados con el tiempo, en condiciones intrínsecas. Línea continua: difusión teórica extrínseca, para $n = 20n_i$. Símbolos rellenos: perfiles atomísticos kMC normalizados con el tiempo para $n = 20n_i$. Para lograr una comparación adecuada hubo que usar estadística de Maxwell-Boltzmann en las simulaciones. 289

11.16 Redistribución de los dopantes en una unión PN . El boro se encontraba inicialmente con una concentración cercana a $2 \times 10^{18} \text{ cm}^{-3}$ (región P). En la región N^+ se ha implantado arsénico, 10 KeV , $5 \times 10^{14} \text{ cm}^{-2}$. Posteriormente se ha recocido toda la muestra durante 10 minutos a 750°C 290

11.17 Histograma de pockets amorfos para implantaciones de iones de Si y Ge implantados a 80 keV de energía con una dosis de $4 \times 10^{12} \text{ cm}^{-2}$ (por debajo del nivel de amorfización) y a una velocidad de dosis de $5 \times 10^{12} \text{ cm}^{-2}\text{s}^{-1}$ a -200°C 293

| | |
|---|-----|
| 11.18(a) Dañado calculado frente a la temperatura del sustrato, para implantaciones de Si en Si de 80 keV, 10^{15} cm^{-2} con velocidades de implantación entre $2 \times 10^{11} \text{ cm}^{-2}\text{s}^{-1}$ y $5 \times 10^{13} \text{ cm}^{-2}\text{s}^{-1}$. (b) Temperatura de transición cristal/amorfo en función de la velocidad de implantación. Los cuadrados se corresponden con las simulaciones en (a) y las estrellas y triángulos con los experimentos hechos a 80 keV por (Goldberg et al., 1995) y 1 MeV por (Schultz et al., 1991) respectivamente. | 296 |
| 11.19 Contribución de las diferentes estructuras de defectos al daño total para una implantación de Si en Si de 80 keV, 10^{15} cm^{-2} con una velocidad de implantación de $2 \times 10^{11} \text{ cm}^{-2}\text{s}^{-1}$, en función de la temperatura de la muestra. | 297 |
| 11.20 Dependencia con la dosis de la fracción de daños debidos a diferentes estructuras de defectos para una implantación de 80 keV de (a) Si y (b) Ge con una velocidad de $5 \times 10^{12} \text{ cm}^{-2}\text{s}^{-1}$ y a -20°C . | 299 |
| 11.21(a) y (b) representan la dependencia de la fracción de daño con la dosis para diferentes defectos para una simulación de implantación de Si a 80 keV, una temperatura de 70°C y una velocidad de implantación de $5 \times 10^{12} \text{ cm}^{-2}\text{s}^{-1}$. Para poder establecer una comparación, (a) está realizada sin límite de concentración, pero en (b) la concentración máxima de partículas en una caja está limitada a 10^{20} cm^{-3} con el objeto de ahorrar recursos de cálculo. | 300 |
| 11.22 Difusividad del As frente a la concentración total de As para temperaturas distintas. Los símbolos son resultados experimentales tomados de la referencia (Fair, 1981b), y las líneas son los valores predichos por DADOS. | 305 |
| 11.23 Resultados SIMS para las estructuras usadas en el experimento de (Rousseau et al., 1998), que consisten en una capa enterrada de B y una capa superficial de As implantada y posteriormente recocida con fusión láser. Después de 2 h. a 700°C , los datos SIMS muestran la difusión incrementada para la capa enterrada de B. La simulación hecha con DADOS (líneas) ajusta bien los perfiles de As y B, y también hay que destacar que la simulación de DADOS predice una concentración de As eléctricamente activa cercana a $2 \times 10^{20} \text{ cm}^{-3}$, en un buen acuerdo con el experimento. | 309 |
| 11.24 Perfiles de As tomados con SIMS por (Kim et al., 2000) para muestras de As implantado a $5 \times 10^{14} \text{ cm}^{-2}$ y recocido posteriormente a dos temperaturas diferentes: a) 720°C durante 300 min y b) 820°C durante 1 y 30 min. Se incluyen también las simulaciones hechas con DADOS para estas condiciones. | 311 |
| A.1. Energía de formación de A_i | 329 |
| A.2. Significado de L_r | 332 |

| | |
|---|-----|
| A.3. Energía de formación de un par IV | 335 |
| A.4. Configuraciones normal y U^- | 337 |
| A.5. Energías de difusión y estados de carga del boro. | 341 |
| A.6. Distancia de salto (λ) y salto largo (Λ). | 341 |
| B.1. Vista del Simulador Atomístico de la Universidad de Valladolid, UVAS. | 348 |
| B.2. La ventana de mensajes muestra el cuaderno de bitácora de las simulaciones, y una barra de progreso va mostrando al usuario el porcentaje de simulación que se ha realizado. | 350 |
| B.3. Entorno común para realizar simulaciones atomísticas. La ventana de la izquierda (con el árbol de comandos) muestra los comandos de entrada, la de la derecha (un árbol de datos), permite navegar por los datos de salida. | 351 |
| B.4. Estructura del UVAS desde un punto de vista del programador. Grandes partes del código son C++ estándar, y por tanto se pue- den compilar en todas las plataformas existentes. Otras porciones, en cambio, dependen del sistema operativo (Unix, Windows) y de- ben ser “portadas”. En la medida en que ha sido posible se han separado y aislado las clases estándar de aquéllas que dependían del sistema operativo. | 352 |
| B.5. Diferentes partes del código del UVAS desde el punto de vista del usuario. | 354 |
| B.6. Cuadro de diálogo que permite al usuario cambiar las opciones globales del programa UVAS. | 356 |
| B.7. Vista parcial del “Script Wizard”. Se utiliza una estructura de árbol para mostrar los comandos y sus opciones. | 357 |
| B.8. Vista del comando de recocido. Este comando requiere un objeto sobre el que operar, el cual se asigna a través del parámetro “Ob- ject label”, que a su vez crea o ajusta (“Create”) los constructores adecuados. Después de ello se muestran los parámetros del recoci- do, para que el usuario los cambie si ésta es su elección: el tiempo y las temperaturas inicial y final. | 359 |
| B.9. Jerarquía de clases de comandos en UVAS. | 360 |
| B.10. Estructura interna del “Script Wizard”. La clase <code>CScriptWizard</code> se encarga de manejar todo el interfaz gráfico (botones, ventan- as...). Esta clase, mediante la utilización del entorno UVAS a modo de pasarela, consigue el <code>UVScript</code> actual, que contiene los coman- dos que deben ejecutarse, así como sus objetos y gráficos asociados. | 360 |
| B.11. Algunas clases de tipo <code>CmdIO</code> funcionando. | 362 |
| B.12. Diálogo asociado y <code>CmdIOBool</code> | 364 |
| B.13. Diálogo asociado y <code>CmdIODouble</code> | 364 |
| B.14. Diálogo asociado y <code>CmdIOUnsignedInt</code> | 365 |

| | |
|---|-----|
| B.15. Diálogo asociado y <code>CmdIOFileName</code> | 365 |
| B.16. Diálogo asociado y <code>CmdIOObject</code> | 366 |
| B.17. Diálogo asociado y <code>CmdIOString</code> | 366 |
| B.18. Un “script” de ejecución con tres comandos. | 367 |
| B.19. Una vez que se ha empezado una simulación vacía (en nuestro caso llamada “Simulation 0”), se puede acceder a los distintos comandos de la simulación a través del menú del programa. | 369 |
| B.20. Estructura interna del “Data Browser”. El objeto <code>CDataBrowserDlg</code> maneja los aspectos relacionados con el entorno gráfico, y está derivado de <code>DataBrowser</code> , que contiene los algoritmos necesarios para manejar los datos “en crudo”. El objeto <code>DataBrowser</code> contiene todas las simulaciones presentes en la sesión, así como todas las fuentes para obtener cada dato posible generado durante la simulación. | 372 |
| B.21. Estructura interna de <code>DataSource</code> . <code>DataSource</code> contiene información sobre la simulación y el objeto del cual está almacenando datos, junto con un array de <code>DataSourceLines</code> , donde está la forma de acceder a dichos datos. | 375 |
| B.22. El índice de un vector se puede fijar a un valor, o solicitar que se itere sobre él. | 376 |
| B.23. Vista de cómo el “data browser” representa los vectores y sus datos asociados. | 377 |
| B.24. Usando los “vectores” (arrays de elementos) el usuario puede iterar sobre sus diferentes índices. En la imagen, una representación está realizada iterando sobre todos los átomos en un tiempo en particular, y la otra a un determinado átomo en su evolución temporal. | 378 |
| B.25. Los valores de salida pueden ser representados o también visualizados como texto, lo que ayuda a que puedan ser copiados y pegados a otras aplicaciones. | 379 |
| B.26. Vista de un objeto <code>UVBool</code> funcionando: su icono, nombre y el dato que contenía. | 380 |
| B.27. Vista de un objeto <code>UVDouble</code> funcionando: su icono, nombre y el dato que contenía. | 380 |
| B.28. Vista de un objeto <code>UVLongInt</code> funcionando: su icono, nombre y el dato que contenía. | 381 |
| B.29. Vista de un objeto <code>UVUnsignedInt</code> funcionando: su icono, nombre y el dato que contenía. | 381 |
| B.30. Vista de un objeto <code>UVVoidPointer</code> funcionando: su icono, nombre y dirección a la que apuntaba. | 382 |
| B.31. Vista de un objeto <code>UVString</code> funcionando: su icono, nombre y cadena a la que apuntaba. | 383 |

| | |
|--|-----|
| B.32. Vista del acceso a un <code>UVObject</code> funcionando: su icono, nombre y datos a los que se accede. | 384 |
| B.33. Vista del acceso a una <code>UVClass</code> funcionando: su icono, nombre y datos a los que se accede. | 384 |
| B.34. Un mapa de color es un gráfico en dos dimensiones, donde el color asignado a cada celda es otra variable. De esta manera los mapas de colores ofrecen la posibilidad de representar información tridimensional en una representación bidimensional pura. | 386 |
| B.35. Estructura interna del <code>UVAS</code> para gestionar las gráficas. | 387 |
| B.36. Gráfica bi-dimensional donde se muestran los ejes x , y , el título y la leyenda. | 389 |
| B.37. Gráfica tridimensional donde se puede ver el título, los ejes x , y y z (en la esquina superior derecha) y la leyenda con los datos representados. | 390 |
| B.38. Captura de pantalla de <code>UVAxisX</code> y su cuadro de diálogo asociado. | 393 |
| B.39. Captura de pantalla de <code>UVAxisY</code> y su cuadro de diálogo asociado. | 394 |
| B.40. Captura de pantalla de <code>UVTitle</code> y su cuadro de diálogo asociado. | 394 |
| B.41. Captura de pantalla de <code>UVLegend</code> y su cuadro de diálogo asociado. | 395 |
| B.42. El selector de series permite asignar y configurar los distintos datos que, tomados del “Data Browser” pasan a formar parte de una serie. Cada una de estas series puede ser dibujada, o bien gestionada al final del proceso para imprimir sus datos. | 398 |
| B.43. Interfaz que gestiona los datos a obtener. | 399 |
| B.44. Captura de pantalla de cómo se obtienen los datos de salud utilizando la estructura del “OutputData”. | 400 |
| B.45. Estructura interna de <code>COutputData</code> | 400 |

Índice de tablas

| | |
|---|-----|
| 7.1. Propiedades termodinámicas de los intersticiales propios (I) y las vacantes (V), obtenidas de estudios experimentales y teóricos. Referencia (Bracht, 2000). | 84 |
| 8.1. Opciones globales de DADOS para la regulación del espacio. | 106 |
| 8.2. Valores internos de DADOS para las frecuencias mínimas y máximas. | 108 |
| 8.3. Parámetros para los intersticiales y vacantes, tomados del fichero de parámetros de DADOS. | 110 |
| 8.4. Interacciones permitidas y prohibidas. Se listan todas las especies y cargas. (i) | 111 |
| 8.5. Interacciones permitidas y prohibidas. (ii) | 112 |
| 8.6. Interacciones permitidas y prohibidas. (iii) | 113 |
| 8.7. Valores de los parámetros utilizados en los clusters de I s y V s. | 118 |
| 8.8. Valores de parámetros para los defectos{311}. | 121 |
| 8.9. Parámetros para los bucles de dislocación. (Cristiano et al., 2000). | 124 |
| 8.10. Valores utilizados en los parámetros de un void. | 126 |
| 8.11. Valores definidos para los pockets amorfos. | 130 |
| 8.12. Parámetros de DADOS para el modelo de amorfización. | 133 |
| 8.13. Valores de los parámetros de recristalización usados por DADOS en la ecuación 8.9. | 135 |
| 8.14. Probabilidad de los dopantes para quedar en la caja donde están y no viajar con el frente de recristalización | 137 |
| 8.15. Concentración máxima de dopantes después de la recristalización | 138 |
| 8.16. Valores de los parámetros de DADOS para la difusión de impurezas en silicio. | 141 |
| 8.17. Valores de los parámetros de DADOS para la difusión de impurezas en óxido de silicio. | 142 |
| 8.18. Valores de los parámetros de DADOS para la difusión de impurezas en nitruro de silicio. | 142 |
| 8.19. Valores de los parámetros de DADOS para formación de enlaces. | 143 |
| 8.20. Interacciones permitidas y prohibidas con las intercaras. | 150 |

| | |
|--|-----|
| 8.21. Parámetros de DADOS relacionados con la recombinación y emisión de defectos puntuales (I , V). | 152 |
| 8.22. Parámetros de DADOS relacionados con la solubilidad de dopantes e impurezas. | 155 |
| 8.23. Lista de especies cargadas de DADOS. | 158 |
| 8.24. Parámetros para los niveles electrónicos a $T = 0$ K. | 160 |
| 8.25. Parámetros relacionados con la dependencia de los niveles de carga con la temperatura. | 161 |
| 8.26. Parámetros usados en DADOS para actualizar la carga de manera estática. | 165 |
| 8.27. Parámetros de DADOS (prefactores y energías de migración) para diferentes estados de carga. | 169 |
| 8.28. Parámetros de DADOS usados para el modelizado del estrechamiento del gap, que dependen de la concentración de dopantes. | 174 |
| 8.29. Parámetros de DADOS utilizados para la generación de las rampas de temperatura. | 178 |
| | |
| 11.1. Parámetros utilizados para los estados cargados del la vacante de silicio (V^j), su intersticial (I^j), y el boro intersticial (B_i^j). D_0 es el prefactor de la difusividad (ecuación 11.28), E_m la energía de migración y e_t el nivel de carga medido desde el borde de la banda de valencia. Estamos suponiendo que e_t escala con la energía del gap como $e_t(T) = e_t(0)E_g(T)/E_g(0)$ (para $T \neq 0$, ver Ref. (Sze, 1981)). Los prefactores para las energías de formación y concentraciones en equilibrio de I^0 y V^0 también se muestran. Otros parámetros que se están usando, pero no están en la tabla, son $D_{bk,0}(B_i^-) = 5 \text{ cm}^2/\text{s}$ y $E_b(B_i^-) = 0,4 \text{ eV}$. $D_{bk,0}$ es el prefactor de ruptura, expresado en unidades de difusividad ($= (\lambda^2/6) \nu_{bk,0}$). $D_{bk,0}(B_i^0)$ y $E_b(B_i^0)(T)$ pueden obtenerse de las ecuaciones 11.32 y 11.33. Por último decir que se ha tomado la distancia de salto λ como la distancia a segundos vecinos (0,384 nm) y el volumen efectivo de captura en $v_{\text{capt}} = 3\lambda^3$, el valor para nuestra implementación. | 281 |
| 11.2. Parámetros de difusión para las especies móviles de As incluidas en DADOS, donde el super-índice denota el estado de carga del defecto. Las difusividades se relacionan con la expresión general $D_X = D_0 \times \exp(-E_m/k_B T) \text{ cm}^2\text{s}^{-1}$. Se asume que los niveles profundos en el gap varían proporcionalmente al estrechamiento del gap con la temperatura, por lo que también variarían las energías de enlace de las especies con la temperatura (Martin-Bragado et al., 2004c). | 304 |

| | |
|---|-----|
| 11.3. Energías de enlace totales para los clusters As_nV_m . Hay que señalar que, debido a su baja energía, es posible formar clusters As_2V , As_3V y As_4V cuando los átomos de As estén lo suficientemente cerca ($As+As+As \rightarrow As_3V+I$), es decir, cuando su concentración sea muy alta. | 307 |
|---|-----|

Índice de parámetros de DADOS

| | | |
|---|-----|----------------------------|
| <h1 style="font-size: 2em; color: blue;">A</h1> | | |
| Acn1_2..... | 174 | |
| Acn1_3..... | 174 | |
| Acn1_4..... | 174 | |
| Acp1_2..... | 174 | |
| Acp1_3..... | 174 | |
| Acp1_4..... | 174 | |
| Agap..... | 161 | |
| AmorphizationThreshold..... | 133 | |
| As..... | 196 | |
| Asi..... | 196 | |
| AsiP..... | 196 | |
| AsNitride..... | 196 | |
| AsOxide..... | 196 | |
| AsV..... | 196 | |
| AsVM..... | 196 | |
| AsVP..... | 196 | |
| Avn1_2..... | 174 | |
| Avn1_3..... | 174 | |
| Avn1_4..... | 174 | |
| Avp1_2..... | 174 | |
| Avp1_3..... | 174 | |
| Avp1_4..... | 174 | |
| <h1 style="font-size: 2em; color: blue;">B</h1> | | |
| B..... | 189 | |
| Bgap..... | 161 | |
| Bi..... | 189 | |
| BiM..... | 189 | |
| | | BiP..... 189 |
| | | BitsBoxes..... 106 |
| | | BNitride..... 189 |
| | | BOxide..... 189 |
| | | burgVectMod..... 124 |
| <h1 style="font-size: 2em; color: blue;">C</h1> | | |
| | | C..... 205 |
| | | ChargeVarPercent..... 165 |
| | | Ci..... 205 |
| | | Csol0_As..... 155, 200 |
| | | Csol0_B..... 155, 194 |
| | | Csol0_F..... 155, 203 |
| | | Csol0_Ph..... 155, 210 |
| <h1 style="font-size: 2em; color: blue;">D</h1> | | |
| | | D0AmorphousPocket..... 130 |
| | | D0FS_I..... 152, 186 |
| | | D0FS_V..... 152, 189 |
| | | Db_AsiP..... 143, 198 |
| | | Db_AsVP..... 143, 198 |
| | | Db_BiM..... 143, 191 |
| | | Db_Ci..... 143, 205 |
| | | Db_IniM..... 143, 215 |
| | | Db_InVM..... 143, 215 |
| | | Db_OV..... 143, 212 |
| | | Db_PhiP..... 143, 209 |
| | | Db_PhVP..... 143, 209 |
| | | Dm_As..... 141, 198 |
| | | Dm_Asi..... 141, 198 |

| | | |
|-----------------------|----------|---|
| Dm_AsiP | 169, 198 | E |
| Dm_AsNitride | 142, 198 | |
| Dm_AsOxide | 142, 198 | |
| Dm_AsV | 141, 198 | |
| Dm_AsVM | 169, 198 | |
| Dm_AsVP | 169, 198 | |
| Dm_B | 141, 191 | |
| Dm_Bi | 141, 191 | |
| Dm_BiM | 169, 191 | |
| Dm_BiP | 169, 191 | |
| Dm_BNitride | 142, 191 | |
| Dm_BOxide | 142, 191 | |
| Dm_C | 141, 205 | |
| Dm_Ci | 141, 205 | |
| Dm_F | 141, 202 | |
| Dm_I | 110, 186 | |
| Dm_IM | 169, 186 | |
| Dm_In | 141, 215 | |
| Dm_Ini | 141, 215 | |
| Dm_IniM | 169, 215 | |
| Dm_InV | 141, 215 | |
| Dm_InVM | 169, 215 | |
| Dm_IP | 169, 186 | |
| Dm_O | 141, 212 | |
| Dm_OV | 141, 212 | |
| Dm_Ph | 141, 209 | |
| Dm_Phi | 141, 209 | |
| Dm_PhiP | 169, 209 | |
| Dm_PhNitride | 142, 209 | |
| Dm_PhOxide | 142, 209 | |
| Dm_PhV | 141, 209 | |
| Dm_PhVM | 169, 209 | |
| Dm_PhVP | 169, 209 | |
| Dm_V | 110, 188 | |
| Dm_VM | 169, 188 | |
| Dm_VMM | 169, 188 | |
| Dm_VP | 169, 188 | |
| Dm_VPP | 169, 188 | |
| dTBase | 178 | |
| dTLimit | 178 | |
| E311toDLoop | 121 | |
| e_AsiP0 | 160, 198 | |
| e_AsVM0 | 160, 198 | |
| e_AsVP0 | 160, 198 | |
| e_BiM0 | 160, 191 | |
| e_BiP0 | 160, 191 | |
| e_IM0 | 160, 186 | |
| e_IniM0 | 160, 215 | |
| e_InVM0 | 160, 215 | |
| e_IP0 | 160, 186 | |
| e_PhiP0 | 160, 209 | |
| e_PhVM0 | 160, 209 | |
| e_PhVP0 | 160, 209 | |
| e_VM0 | 160, 188 | |
| e_VMM0 | 160, 188 | |
| e_VP0 | 160, 188 | |
| e_VPP0 | 160, 188 | |
| EAmorphousPocket | 130 | |
| EAmorphousPocketLarge | 130 | |
| EAmorphousPocketSmall | 130 | |
| Eb_AsiP | 143, 198 | |
| Eb_AsVP | 143, 198 | |
| Eb_BiM | 143, 191 | |
| Eb_Ci | 143, 205 | |
| Eb_IniM | 143, 215 | |
| Eb_InVM | 143, 215 | |
| Eb_LargeCluster | 118 | |
| Eb_OV | 143, 212 | |
| Eb_PhiP | 143, 209 | |
| Eb_PhVP | 143, 209 | |
| Eb_SmallestCluster | 118 | |
| Eform_I | 152, 186 | |
| Eform_V | 152, 189 | |
| Eg0 | 161 | |
| Em_As | 141, 198 | |
| Em_Asi | 141, 198 | |
| Em_AsiP | 169, 198 | |
| Em_AsNitride | 142, 198 | |
| Em_AsOxide | 142, 198 | |
| Em_AsV | 141, 198 | |

| | | | | |
|------------------------------|----------|---------------------|-------------------|-----|
| Em_AsVM..... | 169, 198 | G | gamma..... | 124 |
| Em_AsVP..... | 169, 198 | | | |
| Em_B..... | 141, 191 | H | HistoBits..... | 106 |
| Em_Bi..... | 141, 191 | | | |
| Em_BiM..... | 169, 191 | I | I..... | 185 |
| Em_BiP..... | 169, 191 | | | |
| Em_BNitride..... | 142, 191 | IM..... | In..... | 214 |
| Em_BOxide..... | 142, 191 | | | |
| Em_C..... | 141, 205 | Ini..... | IniP..... | 214 |
| Em_Ci..... | 141, 205 | | | |
| Em_F..... | 141, 202 | InV..... | InVM..... | 214 |
| Em_I..... | 110, 186 | | | |
| Em_IM..... | 169, 186 | IP..... | | 185 |
| Em_In..... | 141, 215 | | | |
| Em_Ini..... | 141, 215 | M | MaxIVStorage..... | 133 |
| Em_IniM..... | 169, 215 | | | |
| Em_InV..... | 141, 215 | MAXVOIDSDIAMnm..... | MIN311SIZE..... | 121 |
| Em_InVM..... | 169, 215 | | | |
| Em_IP..... | 169, 186 | MINVoidsSIZE..... | mu..... | 124 |
| Em_O..... | 141, 212 | | | |
| Em_OV..... | 141, 212 | N | Nc300..... | 161 |
| Em_Ph..... | 141, 209 | | | |
| Em_Phi..... | 141, 209 | nInit..... | nu..... | 124 |
| Em_PhiP..... | 169, 209 | | | |
| Em_PhNitride..... | 142, 209 | Nv300..... | | 161 |
| Em_PhOxide..... | 142, 209 | | | |
| Em_PhV..... | 141, 209 | O | O..... | 211 |
| Em_PhVM..... | 169, 209 | | | |
| Em_PhVP..... | 169, 209 | P | OV..... | 211 |
| Em_V..... | 110, 188 | | | |
| Em_VM..... | 169, 188 | Ph..... | | 207 |
| Em_VMM..... | 169, 188 | | | |
| Em_VP..... | 169, 188 | | | |
| Em_VPP..... | 169, 188 | | | |
| Erecryst..... | 135 | | | |
| Esol_As..... | 155, 200 | | | |
| Esol_B..... | 155, 194 | | | |
| Esol_F..... | 155, 203 | | | |
| Esol_Ph..... | 155, 210 | | | |
| expNc..... | 161 | | | |
| expNv..... | 161 | | | |
| exponent_Cluster..... | 118 | | | |
| ExponentAmorphousPocket..... | 130 | | | |

| | | | |
|-----------------|-----|----------------------|----------|
| Phi..... | 207 | Surf_Eh_F | 155, 203 |
| PhiP | 207 | Surf_Eh_Ph | 155, 210 |
| PhNitride | 207 | Surf_h0_As | 155, 200 |
| PhOxide | 207 | Surf_h0_B | 155, 194 |
| PhV | 207 | Surf_h0_F | 155, 203 |
| PhVM | 207 | Surf_h0_Ph | 155, 210 |
| PhVP | 207 | SurfMaxTrap_As | 155, 200 |

R

| | |
|----------------|----------|
| RecLnm_I | 152, 186 |
| RecLnm_V | 152, 189 |

S

| | |
|-----------------------------|----------|
| Segreg_eV_AsNitride..... | 155, 200 |
| Segreg_eV_AsOxide..... | 155, 200 |
| Segreg_eV_BNitride..... | 155, 194 |
| Segreg_eV_BOxide..... | 155, 194 |
| Segreg_eV_FNitride..... | 203 |
| Segreg_eV_FOxide..... | 203 |
| Segreg_eV_PhNitride..... | 155, 210 |
| Segreg_eV_PhOxide..... | 155, 210 |
| Segreg_pref_AsNitride..... | 155, 200 |
| Segreg_pref_AsOxide..... | 155, 200 |
| Segreg_pref_BNitride..... | 155, 194 |
| Segreg_pref_BOxide..... | 155, 194 |
| Segreg_pref_FNitride..... | 203 |
| Segreg_pref_FOxide..... | 203 |
| Segreg_pref_PhNitride | 155, 210 |
| Segreg_pref_PhOxide..... | 155, 210 |
| size0_311toDLoop | 121 |
| Surf_e_a0_As..... | 155, 200 |
| Surf_e_a0_B..... | 155, 194 |
| Surf_e_a0_F | 155, 203 |
| Surf_e_a0_Ph..... | 155, 210 |
| Surf_Ee.a.As | 155, 200 |
| Surf_Ee.a.B | 155, 194 |
| Surf_Ee.a.F | 155, 203 |
| Surf_Ee.a.Ph..... | 155, 210 |
| Surf_Eh.As..... | 155, 200 |
| Surf_Eh.B..... | 155, 194 |

V

| | |
|-----------------|-----|
| V | 187 |
| V0recryst | 135 |
| VM | 187 |
| VMM | 187 |
| VP | 187 |
| VPP | 187 |

Índice general

Símbolos

| | | | |
|---------------------|----------|---------------------------------------|---------------|
| $(e/a)_0(As)$ | 155 | B | 191 |
| $(e/a)_0(B)$ | 155 | $BNitride$ | 189, 191 |
| $(e/a)_0(F)$ | 155 | $BOxide$ | 189, 191 |
| $(e/a)_0(P)$ | 155 | B^- | 189 |
| $A_{cn1/2}$ | 174 | B_g | 161 |
| $A_{cn1/3}$ | 174 | B_i | 191 |
| $A_{cn1/4}$ | 174 | B_i^+ | 189, 191 |
| $A_{cp1/2}$ | 174 | B_i^- | 189, 191 |
| $A_{cp1/3}$ | 174 | B_i^0 | 189 |
| $A_{cp1/4}$ | 174 | C | 205 |
| $A_{vn1/2}$ | 174 | C^0 | 205 |
| $A_{vn1/3}$ | 174 | $C_{0,Sol}(As)$ | 155 |
| $A_{vn1/4}$ | 174 | $C_{0,Sol}(B)$ | 155 |
| $A_{vp1/2}$ | 174 | $C_{0,Sol}(F)$ | 155 |
| $A_{vp1/3}$ | 174 | $C_{0,Sol}(P)$ | 155 |
| $A_{vp1/4}$ | 174 | C_i | 205 |
| A_g | 161 | C_i^0 | 205 |
| As | 198 | $D_{emission,0}^{interfaz}(I)$ | 333 |
| $AsNitride$ | 196, 198 | $D_{emission,0}^{interface}(I)$ | 152, 186 |
| $AsOxide$ | 196, 198 | $D_{emission,0}^{interface}(V)$ | 152, 189 |
| As^+ | 196 | $D_{0,recomb}$ | 130 |
| As_V | 198 | $D_{b,0}(As_V^+)$ | 143, 198 |
| As_V^+ | 196, 198 | $D_{b,0}(As_i^+)$ | 143, 198 |
| As_V^- | 196, 198 | $D_{b,0}(B_i^-)$ | 143, 191 |
| As_V^0 | 196 | $D_{b,0}(C_i)$ | 143, 205 |
| As_i | 198 | $D_{b,0}(In_V^-)$ | 143, 215 |
| As_i^+ | 196, 198 | $D_{b,0}(In_i^-)$ | 143, 215 |
| As_i^0 | 196 | $D_{b,0}(O_V)$ | 143, 212 |
| | | $D_{b,0}(P_V^+)$ | 143, 209 |
| | | $D_{b,0}(P_i^+)$ | 143, 209 |
| | | $D_{m,0}(As)$ | 141, 142, 198 |
| | | $D_{m,0}(AsNitride)$ | 198 |

| | | | |
|---------------------------|---------------|--------------------------------------|---------------|
| $D_{m,0}(AsOxide)$ | 198 | $E_{Oxi,g}(B)$ | 155 |
| $D_{m,0}(As_V)$ | 141, 198 | $E_{Oxi,g}(P)$ | 155 |
| $D_{m,0}(As_V^+)$ | 169, 198 | $E_{Sol}(As)$ | 155 |
| $D_{m,0}(As_V^-)$ | 169, 198 | $E_{Sol}(B)$ | 155 |
| $D_{m,0}(As_i)$ | 141, 198 | $E_{Sol}(F)$ | 155 |
| $D_{m,0}(As_i^+)$ | 169, 198 | $E_{Sol}(P)$ | 155 |
| $D_{m,0}(B)$ | 141, 142, 191 | $E_{\{311\} \rightarrow loop}$ | 121 |
| $D_{m,0}(BNitride)$ | 191 | $E_b(As_V^+)$ | 143, 198 |
| $D_{m,0}(BOxide)$ | 191 | $E_b(As_i^+)$ | 143, 198 |
| $D_{m,0}(B_i)$ | 141, 191 | $E_b(B_i^-)$ | 143, 191 |
| $D_{m,0}(B_i^+)$ | 169, 191 | $E_b(C_i)$ | 143, 205 |
| $D_{m,0}(B_i^-)$ | 169, 191 | $E_b(In_V^-)$ | 143, 215 |
| $D_{m,0}(C)$ | 141, 205 | $E_b(In_i^-)$ | 143, 215 |
| $D_{m,0}(C_i)$ | 141, 205 | $E_b(O_V)$ | 143, 212 |
| $D_{m,0}(F)$ | 141, 202 | $E_b(P_V^+)$ | 143, 209 |
| $D_{m,0}(I)$ | 110, 186 | $E_b(P_i^+)$ | 143, 209 |
| $D_{m,0}(I^+)$ | 169, 186 | $E_f(I)$ | 152, 186 |
| $D_{m,0}(I^-)$ | 169, 186 | $E_f(V)$ | 152, 189 |
| $D_{m,0}(In)$ | 141, 215 | $E_g(T = 0K)$ | 161 |
| $D_{m,0}(In_V)$ | 141, 215 | $E_h(As)$ | 155 |
| $D_{m,0}(In_V^-)$ | 169, 215 | $E_h(B)$ | 155 |
| $D_{m,0}(In_i)$ | 141, 215 | $E_h(F)$ | 155 |
| $D_{m,0}(In_i^-)$ | 169, 215 | $E_h(P)$ | 155 |
| $D_{m,0}(O)$ | 141, 212 | $E_m(As)$ | 141, 142, 198 |
| $D_{m,0}(O_V)$ | 141, 212 | $E_m(AsNitride)$ | 198 |
| $D_{m,0}(P)$ | 209 | $E_m(AsOxide)$ | 198 |
| $D_{m,0}(PNitride)$ | 209 | $E_m(As_V)$ | 141, 198 |
| $D_{m,0}(POxide)$ | 209 | $E_m(As_V^+)$ | 169, 198 |
| $D_{m,0}(P_V)$ | 141, 209 | $E_m(As_V^-)$ | 169, 198 |
| $D_{m,0}(P_V^+)$ | 169, 209 | $E_m(As_i)$ | 141, 198 |
| $D_{m,0}(P_V^-)$ | 169, 209 | $E_m(As_i^+)$ | 169, 198 |
| $D_{m,0}(P_i)$ | 141, 209 | $E_m(B)$ | 141, 142, 191 |
| $D_{m,0}(P_i^+)$ | 169, 209 | $E_m(BNitride)$ | 191 |
| $D_{m,0}(Ph)$ | 141, 142 | $E_m(BOxide)$ | 191 |
| $D_{m,0}(V)$ | 110, 188 | $E_m(B_i)$ | 141, 191 |
| $D_{m,0}(V^+)$ | 169, 188 | $E_m(B_i^+)$ | 169, 191 |
| $D_{m,0}(V^-)$ | 169, 188 | $E_m(B_i^-)$ | 169, 191 |
| $D_{m,0}(V^{++})$ | 169, 188 | $E_m(C)$ | 141, 205 |
| $D_{m,0}(V^{--})$ | 169, 188 | $E_m(C_i)$ | 141, 205 |
| $E_{Nit,g}(As)$ | 155 | $E_m(F)$ | 141, 202 |
| $E_{Nit,g}(B)$ | 155 | $E_m(I)$ | 110, 186 |
| $E_{Nit,g}(P)$ | 155 | $E_m(I^+)$ | 169, 186 |
| $E_{Oxi,g}(As)$ | 155 | $E_m(I^-)$ | 169, 186 |

| | | | |
|--------------------------|----------|---|--------------|
| $E_m(In)$ | 141, 215 | In_i | 215 |
| $E_m(In_V)$ | 141, 215 | In_i^- | 214, 215 |
| $E_m(In_V^-)$ | 169, 215 | In_i^0 | 214 |
| $E_m(In_i)$ | 141, 215 | $K_{\nu \rightarrow \text{int}}$ | 108 |
| $E_m(In_i^-)$ | 169, 215 | L_r | 151 |
| $E_m(O)$ | 141, 212 | $L_r(I)$ | 152, 186 |
| $E_m(O_V)$ | 141, 212 | $L_r(V)$ | 152, 189 |
| $E_m(P)$ | 209 | $MaxStorage$ | 133 |
| $E_m(PNitride)$ | 209 | $N_{0,loop}$ | 121 |
| $E_m(POxide)$ | 209 | N_{slice} | 106 |
| $E_m(P_V)$ | 141, 209 | $N_c(T = 300K)$ | 161 |
| $E_m(P_V^+)$ | 169, 209 | $N_v(T = 300K)$ | 161 |
| $E_m(P_V^-)$ | 169, 209 | O | 212 |
| $E_m(P_i)$ | 141, 209 | O^0 | 211 |
| $E_m(P_i^+)$ | 169, 209 | O_V | 212 |
| $E_m(Ph)$ | 141, 142 | O_V^0 | 211 |
| $E_m(V)$ | 110, 188 | P | 209 |
| $E_m(V^+)$ | 169, 188 | $PNitride$ | 207, 209 |
| $E_m(V^-)$ | 169, 188 | $POxide$ | 207, 209 |
| $E_m(V^{++})$ | 169, 188 | P^+ | 207 |
| $E_m(V^{--})$ | 169, 188 | P^- | 165 |
| $E_{(e/a)}(As)$ | 155 | P^- | 165 |
| $E_{(e/a)}(B)$ | 155 | P_V | 209 |
| $E_{(e/a)}(F)$ | 155 | P_V^+ | 207, 209 |
| $E_{(e/a)}(P)$ | 155 | P_V^- | 207, 209 |
| $E_{AmPoc,L}$ | 130 | P_V^0 | 207 |
| $E_{AmPoc,S}$ | 130 | P_{deposit} | 137 |
| $E_{AmPoc,recomb}$ | 130 | $P_d(i, j)$ | 338 |
| $E_{b,L}(I)$ | 118 | P_f^q | 170 |
| $E_{b,L}(V)$ | 118 | P_i | 209 |
| $E_{b,S}(I)$ | 118 | P_i^+ | 207, 209 |
| $E_{b,S}(V)$ | 118 | P_i^0 | 207 |
| E_{recrys} | 135 | V | 188 |
| F | 202 | V^+ | 83, 187, 188 |
| I | 186 | V^- | 83, 187, 188 |
| I^+ | 185, 186 | V^0 | 187 |
| I^- | 185, 186 | V^{++} | 187, 188 |
| I^0 | 185 | V^{--} | 187, 188 |
| In | 215 | $V_{0,recrys}$ | 135 |
| In^- | 214 | $[As]_{\text{max}}^{\text{recrys}}$ | 138, 200 |
| In_V | 215 | $[B]_{\text{max}}^{\text{recrys}}$ | 138, 194 |
| In_V^- | 214, 215 | $[C]_{\text{max}}^{\text{recrys}}$ | 138 |
| In_V^0 | 214 | $[F]_{\text{max}}^{\text{recrys}}$ | 138, 203 |

| | | | |
|-------------------------------------|--------------|------------------------------|----------------------------|
| $[In]_{\max}^{\text{recrys}}$ | 138 | $e(0, -)(As_V^-)$ | 160, 198 |
| $[O]_{\max}^{\text{recrys}}$ | 138 | $e(0, -)(B_i^-)$ | 160, 191 |
| $[P]_{\max}^{\text{recrys}}$ | 138, 210 | $e(0, -)(I^-)$ | 160, 186 |
| ΔT_{base} | 178 | $e(0, -)(In_V^-)$ | 160, 215 |
| ΔT_{init} | 178 | $e(0, -)(In_i^-)$ | 160, 215 |
| Δt_{min} | 108 | $e(0, -)(P_V^-)$ | 160, 209 |
| L_r | 332 | $e(0, -)(V^-)$ | 160, 188 |
| Λ | 171, 342 | e_F | 160 |
| Φ | 325 | $g_{\text{Nit},0}(As)$ | 155 |
| η | 168 | $g_{\text{Nit},0}(B)$ | 155 |
| γ | 124 | $g_{\text{Nit},0}(P)$ | 155 |
| λ | 95, 104, 171 | $g_{\text{Oxi},0}(As)$ | 155 |
| Amorphizationthreshold | 133 | $g_{\text{Oxi},0}(B)$ | 155 |
| ChargeVariation | 165 | $g_{\text{Oxi},0}(P)$ | 155 |
| Cluster \rightarrow Void | 126 | $h_0(As)$ | 155 |
| Cluster \rightarrow {311} | 121 | $h_0(B)$ | 155 |
| NParticles _{max} | 108 | $h_0(F)$ | 155 |
| exp N_c | 161 | $h_0(P)$ | 155 |
| exp N_v | 161 | n_{init} | 178 |
| maxbits | 106 | <i>implant</i> | 175 |
| μ | 124 | read profile | 175 |
| ν | 124 | {311} | <i>véase</i> defecto {311} |
| ν_{\max} | 108 | índice de coordinación | 62 |
| ν_{\min} | 108 | | |
| $\sigma_{\max}(As)$ | 155 | | |
| $\sigma_{\max}(B)$ | 155 | | |
| $\sigma_{\max}(F)$ | 155 | | |
| $\sigma_{\max}(P)$ | 155 | | |
| τ | 104 | | |
| a | 130 | | |
| $a(I)$ | 118 | | |
| $a(V)$ | 118 | | |
| b | 124 | | |
| $e(++,+)(V^{++})$ | 160 | | |
| $e(+,++)(V^{++})$ | 188 | | |
| $e(+,0)(As_V^+)$ | 160, 198 | | |
| $e(+,0)(As_i^+)$ | 160, 198 | | |
| $e(+,0)(B_i^+)$ | 160, 191 | | |
| $e(+,0)(I^+)$ | 160, 186 | | |
| $e(+,0)(P_V^+)$ | 160, 209 | | |
| $e(+,0)(P_i^+)$ | 160, 209 | | |
| $e(+,0)(V^+)$ | 160, 188 | | |
| $e(-,--)(V^{--})$ | 160, 188 | | |

A

| | |
|-----------------------------|-----------------------------|
| ab-initio | 81 |
| actualización | |
| dinámica | 163, 337, 338 |
| estática | 163, 164, 337 |
| acumulación de dañado | 126 |
| ad-atom | 26 |
| ambient | 149 |
| amorfización | 131 |
| amorfo | <i>véase</i> defecto amorfo |
| amorphous pocket | <i>véase</i> pocket amorfo |
| anneal | <i>véase</i> recocido |
| aproximación | |
| ortogonal | 110 |
| arsénico | 196 |
| cluster | 200 |
| defecto puntual | 196, 197 |
| energías | 197, 198 |

- fuelle de 196
interacción 197
intercara 198
pockets amorfos 200
recristalización 200
As..... *véase* arsénico
- ## B
- B..... *véase* boro
barreras de migración 166
boro 189
clusters 195
defecto puntual..... 189
difusión..... 170, 339
difusividad 330
energías 190, 191
fuente de 189
interacción 190
intercara 191
pocket amorfo 194
recristalización 194
salto largo 194, 342
- ## C
- C..... *véase* carbono
campo eléctrico..... 165, 179
carbono 204
cluster..... 206
defecto puntual..... 205
energías 205
fuente de 205
interacción 205
pockets amorfos 205
carga..... 157
actualización... *véase* actualización
barreras de migración 166
compensación 167
dependencia con la temperatura 160
deriva 165
difusividad 168
energías de formación..... 159
formación 170
hipótesis 158
implementación..... 166
nivel de Fermi 85, 160, 168
probabilidad..... 162
ruptura..... 170
suavizado 167
casi-equilibrio 328
celda 14
amorfa 133
de simulación..... 105
cluster 86, 114
captura..... 116
de impurezas 139, 143
captura..... 145
emisión..... 146
Frank-Turnbull 147
histograma 180
recombinación 146
emisión 116
forma 115
histograma 179
impurezas 98
intersticiales..... 86
vacantes 90
comandos 175
concentración 104
electrónica 179
condición
de contorno..... 14
extrínseca 173
intrínseca..... 173, 287
periódica 14
- ## D
- dañado..... 126
DADOS..... 82
defecto
{311}..... 87, 118
captura..... 120

- mínima 108
 función de Fermi-Dirac 168
- ## G
- gap 160
 estrechamiento 174
- ## H
- histéresis 120
 histograma
 cluster 179
 cluster de impurezas 180
 pocket amorfo 180
- ## I
- I *véase* intersticial
 imagen 179
 impurezas 97, 139
 In *véase* indio
 in-diffusion 242
 indio 213
 defecto puntual 213
 energía 214
 fuente de 214
 interacción 214
 intercara 215
 pocket amorfo 215
 información
 atómica 179
 de perfiles 179
 integrador 14
 Verlet 15
 interacciones 158
 intercara 95, 149
 captura 151, 156
 defectos puntuales I, V 149
 emisión 153, 154
 estado estacionario 156
 impurezas 153
- interfaz *véase* intercara
 intersticial 83, 185
 capacidad de transporte 333
 clusters 114
 concentración 333
 concentración en equilibrio 333
 defecto puntual 185
 difusividad 85
 energías 186
 fuente de 185
 interacción 110, 185
 intercara 186
 migración 109
 pocket amorfo 187
- ## K
- kick-out 97, 140, 303
 kMC 81
- ## L
- Lennard-Jones 42
 ley
 de continuidad 325
 de Fick 325
 longitud de recombinación 95, 149, 332
- ## M
- material 105
 MD 11, 92
 microcanónica 19
 migración 330
 Monte Carlo Cinético 81
- ## N
- nitridación 95, 157
 nivel
 de Fermi *véase* carga nivel de Fermi
 electrónico 160

intrínseco.....161
 NMOSFET.....98

O

O..... *véase* oxígeno
 Ostwald ripening..... 87
 oxígeno.....211
 cluster.....213
 defecto puntual.....211
 energías..... 212
 fuente de.....212
 interacción..... 212
 intercara..... 212
 pockets amorfos..... 212
 oxidación..... 95, 157

P

P..... *véase* fósforo
 partícula
 oculta..... 129, 133
 tipo..... 105
 PDE..... 81
 PDL..... 88
 perfiles
 una dimensión.....181
 pocket amorfo.... *véase* defecto pocket amorfo
 portadores mayoritarios..... 168
 potencial..... 16
 Lennard–Jones..... 42
 Stillinger–Weber.....17
 prefactor..... 329
 probabilidad de formación.....170

R

radio de corte..... 14
 rate..... 107
 reacción
 de emparejado.....170

de ruptura..... 170
 recocido
 dinámico..... 127
 recristalización.....94, 134
 región amorfa..... 133
 reversibilidad microscópica..... 170

S

salto..... 104
 largo..... 171, 342
 silicidación..... 95
 Silicio
 defectos..... 83
 sistema conservativo.....19
 snapshot..... 179
 solubilidad..... 173, 287
 Stillinger–Weber.....16
 suavizado electrónico..... 167
 superficie..... *véase* intercara
 supersaturación.....179, 336
 forma de cálculo.....181

T

tiempo..... 104
 timer.....107, 153

U

UVAS-DADOS..... 105

V

V..... *véase* vacante
 vacante..... 83, 187
 cluster.....90
 clusters.....114
 defecto puntual.....187
 difusividad..... 85
 energías..... 188
 fuente de..... 187

| | |
|--------------------------|---------------------------|
| interacción..... | 110, 188 |
| intercara | 189 |
| migración | 109 |
| pocket amorfo | 189 |
| Verlet | 16 |
| void | <i>véase</i> defecto void |
| volumen de captura | 114 |