



instrucciones. Y finalmente el último módulo hijo debería ser la memoria de datos.

Para la implementación de las memorias, Verilog permite definir estructuras bidimensionales, como por ejemplo:

```
reg [31:0] nombre [1023:0]
```

Creamos una estructura tipo registro bidimensional, de 1024 direcciones y 32 bits cada una de ellas. Sólo se puede acceder a una dirección de la estructura de cada vez. Por ejemplo:

```
nombre [2]
```

nos daría como resultado el vector contenido en la dirección 2, de 32 bits de longitud.

## **1º) Crear un módulo que implemente la memoria de datos**

La memoria de datos debería actuar como una memoria RAM, ya que durante la ejecución del programa tenemos que ser capaces de leer su contenido y de escribir datos en ella.

Su escritura será síncrona, para evitar que se active el bit de escritura (MW) antes que los cambios de dirección. Es decir, ha de contener una entrada para el reloj. El tiempo de escritura ha de ser de 10 ns (tiempo desde que se da el flanco activo del reloj (positivo) hasta que el dato queda grabado en la memoria).

Su lectura será asíncrona, y su tiempo de lectura ha de ser de otros 10 ns (tiempo desde que cambia la dirección o cambia el dato contenido hasta el cambio en la salida).

## **2º) Crear un módulo que implemente el controlador.**

Lo primero, ha de crearse un módulo que implemente el contador de programa. El retardo del contador ha de ser de 5 ns (desde el flanco activo del reloj hasta el cambio en la salida del contador).

A continuación ha de implementarse otro módulo que implemente la memoria de instrucciones. La memoria de instrucciones ha de funcionar como una memoria ROM, ya que suponemos que previamente se ha grabado en ella las instrucciones de las que se compone el programa, y durante la ejecución del programa se ha de acceder a ella para la lectura de instrucciones pero no para la escritura de datos. Al igual que la memoria de datos, la lectura de la memoria de instrucciones ha de ser asíncrona. El retardo de lectura de la memoria de instrucciones ha de ser de 10 ns.

Una vez hecho esto, se han de juntar en un módulo que implemente el controlador. Suponer que el decodificador de instrucciones no tiene ningún retardo.

### **3º) Crear el módulo padre que implemente la máquina completa.**

A partir de los módulos de la ruta de datos, la memoria de instrucciones y el controlador.

### **4º) Comprobación de la máquina con un programa específico**

La máquina ha de comprobarse del siguiente modo. Inicialmente, la dirección 0 de la memoria de datos ha de contener un número entero positivo. Después de la ejecución del programa, la dirección 1 de la memoria ha de contener el número inicial, y la dirección 0 de la memoria la raíz cuadrada por defecto del número que inicialmente estaba en esa posición (si por ejemplo se almacena el número 70, la raíz por defecto ha de ser 8). En el anexo se puede ver un algoritmo para poder hallar la raíz cuadrada de un número entero positivo.

Crear una instrucción que será la última para finalizar el programa. Esta instrucción consistirá en saltar a sí misma, de modo que una vez en ella sea imposible salir de ella.

Escribir el programa de dos formas:

1º) En lenguaje simbólico (lenguaje ensamblador) para cada una de las direcciones de memoria.

1º) En código máquina, poniendo claramente los bits que se almacenan en cada dirección de la memoria de instrucciones.

### **ANEXO: ALGORITMO PARA EL CÁLULO DE LA RAÍZ CUADRADA**

Para calcular la raíz cuadrada por defecto de un número natural que llamaremos X, el programa irá calculando sucesivamente los cuadrados de los números naturales empezando por 1. Cada vez que calcule un nuevo cuadrado lo comparará con el número X. Si el nuevo cuadrado es menor que X calculará el siguiente cuadrado hasta que el resultado de la comparación nos indique que el cuadrado que acaba de hallar es mayor que X. En ese momento el programa dejará de calcular cuadrados y almacenará el resultado.

Para el cálculo del cuadrado de un número natural  $X=(X-1)+1$  se utilizará el siguiente algoritmo:

$$X^2=((X-1)+1)^2=(X-1)^2+2*(X-1)*1+1^2=(X-1)^2+2(X-1)+1$$

Consiste en hacer una serie de iteraciones. La primera es muy sencilla, cuando  $X = 1$ , es decir  $X-1 = 0$ . Tenemos que conseguir sacar a partir de ahí  $X^2$ . Comprobamos si es mayor que nuestro número o no.

Si no lo es, sumamos un 1 a  $X$  y repetimos lo mismo. Ya tenemos la  $(X-1)^2$  nueva, ya que es el resultado de  $X^2$  de la anterior iteración. Y hay que conseguir el cuadrado del nuevo número (que es 2 y su cuadrado es 4). Lo comparamos con el número original para comprobar si es mayor o no, y si no lo es, repetir las iteraciones necesarias hasta que lo sea.