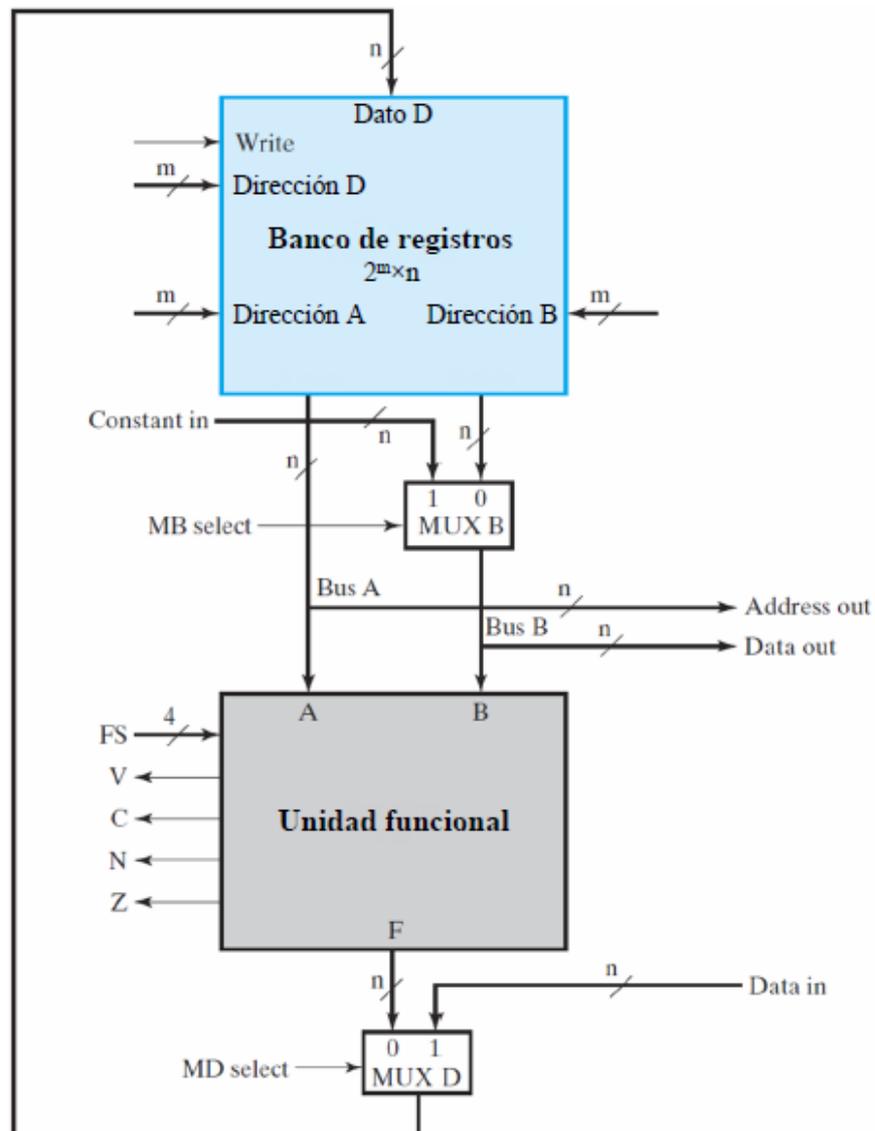


Práctica 3 - Diseño de una ruta de datos completa

Práctica

El objetivo de esta práctica es el diseño de la ruta de datos correspondiente a la máquina estudiada en clases de teoría. Es la siguiente:



El funcionamiento de la unidad funcional es el estudiado en clase, que se resume en la siguiente tabla:

FS(3:0)	MF Select	G Select(3:0)	H Select(1:0)	Microoperación
0000	0	0000	XX	$F = A$
0001	0	0001	XX	$F = A + 1$
0010	0	0010	XX	$F = A + B$
0011	0	0011	XX	$F = A + \overline{B} + 1$
0100	0	0100	XX	$F = A + \overline{B}$
0101	0	0101	XX	$F = A + \overline{B} + 1$
0110	0	0110	XX	$F = A - 1$
0111	0	0111	XX	$F = A$
1000	0	1X00	XX	$F = A \wedge B$
1001	0	1X01	XX	$F = A \vee B$
1010	0	1X10	XX	$F = A \oplus B$
1011	0	1X11	XX	$F = \overline{A}$
1100	1	XXXX	00	$F = B$
1101	1	XXXX	01	$F = sr B$
1110	1	XXXX	10	$F = sl B$

La unidad funcional ya ha sido diseñada en la anterior práctica.

PRE-INFORME

- 1- Realizar un módulo Verilog que describa el comportamiento tanto del MUX_B como del MUX_D (ambos son idénticos): ¡¡Cuidado!! Ambos tienen dos entradas de 16 bits cada una, y el bit para seleccionar grupo. Asignar un retardo de 2 ns a este dispositivo.
- 2- Realizar otro módulo en Verilog (no ha de contener submódulos) que describa el funcionamiento del banco de registros. ¡¡Cuidado!! El funcionamiento del banco de registros es síncrono, por lo que necesitamos una señal de reloj. El número de registros ha de ser 8, y la longitud de cada uno de los registros ha de ser de 16 bits, de acuerdo a la longitud de palabra con la que trabaja la Unidad Funcional. Este módulo ha de contener únicamente los dos siguientes retardos:
 - a. Desde que se da un flanco activo del reloj (positivo) hasta que el registro es escrito (la salida del registro seleccionado cambia), han de pasar 3 ns.

- b. Desde un cambio en la salida de un registro o direcciones de salida, hasta el cambio en las salidas del banco de registros han de pasar 2 ns.

3- Realizar un módulo en Verilog que simule la ruta de datos completa. Este módulo ha de ser muy sencillo, y contendrá como módulos hijos al módulo unidad_funcional ya diseñado en la anterior práctica, al módulo que describe el banco de registros, y al módulo que describe los multiplexores.

4- Supondremos ahora que en t=0, cada uno de los 8 registros tiene almacenada como valor su número, es decir R[0]=0, R[1]=1, ..., R[7]=7.

En cada ciclo de reloj llegan los siguientes datos a la ruta de datos (escritos aquí en hexadecimal):

Dirección_	Dirección	Writ	Dirección	Constant_I	MB_s	F	Data_	MD_sel
A	_B	e	_D	n	el	S	In	
7	0	1	7	*	0	1	*	0
1	2	1	6	*	0	2	*	0
3	7	1	4	*	0	3	*	0
1	0	1	2	*	0	4	*	0
5	7	1	5	*	0	5	*	0
5	7	1	5	*	0	6	*	0
5	6	1	7	*	0	8	*	0
5	2	1	2	*	0	9	*	0
5	6	1	5	*	0	A	*	0
5	6	1	2	*	0	B	*	0
1	5	1	5	*	0	D	*	0
1	4	1	4	*	0	E	*	0
4	5	1	4	*	0	2	*	0
5	0	1	5	OFF1	1	5	*	0
0	5	1	0	*	0	C	*	0

Para cada ciclo de reloj, escribir una tabla con el contenido de cada uno de los 8 registros (en hexadecimal), así como el contenido de los bits de estado ("flags").

EN EL LABORATORIO

Realizar el diseño completo de la ruta de datos y un testbench para la comprobación, siguiendo las instrucciones de la tabla que ha de haber sido previamente rellena.