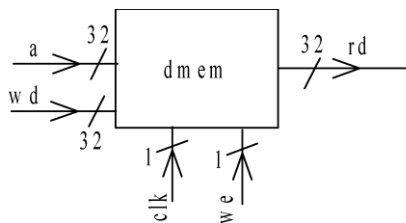


Arquitectura de Computadoras 2015

Práctico N° 2

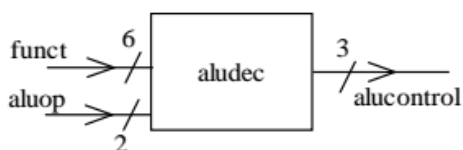
Para todos los ejercicios se pide que respeten los nombres de las entidades, entradas y salidas. Representación de datos siempre en *std_logic* o *std_logic_vector*. Internamente, se les da la libertad para que utilicen el diseño que consideren más adecuado siempre y cuando la salida sea la esperada.

Ejercicio 8) Diseñe una memoria RAM según el diagrama. (Internamente descarte los *bits* 32..8 y 1..0; es decir, solo direcciona 64 palabras según los *bits* 7 a 2 de la dirección.



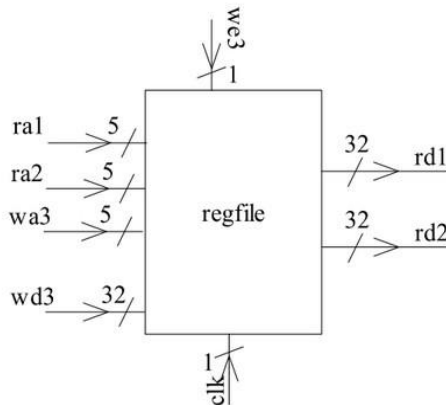
Nota: el dato contenido en 'wd' va a guardarse en la dirección 'a' siempre que 'we' sea uno.

Ejercicio 9) Diseñe un módulo decodificador según el diagrama y la tabla.



aluop	funct	alucontrol
00	-----	010
01	-----	110
1-	100000	010
1-	100010	110
1-	100100	000
1-	100101	001
1-	101010	111

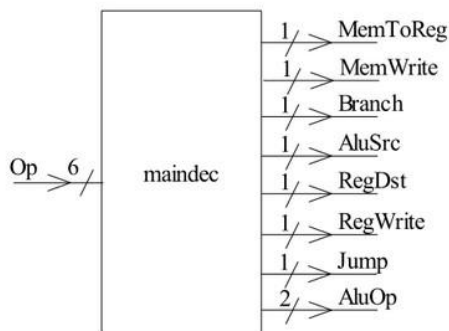
Ejercicio 10) Diseñe una memoria de doble puerto de salida y un puerto de escritura según el diagrama.



Nota: el contenido de la memoria en la dirección 0 podría no estar definido, en caso de que la dirección (*ra1* o *ra2*) sea 0, debe retornarse 0.

Implemente en VHDL, diseñe un *testbench* adecuado, simule con GHDL y GTKWave.

Ejercicio 11) Diseñe un decodificador según la tabla y el diagrama.



Op	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemToReg	Jump	AluOp
000000	1	1	0	0	0	0	0	10
100011	1	0	1	0	0	1	0	00
101011	0	0	1	0	1	0	0	00
000100	0	0	0	1	0	0	0	01
001000	1	0	1	0	0	0	0	00
000010	0	0	0	0	0	0	1	00
Others	-	-	-	-	-	-	-	--

Implemente en VHDL, diseñe un *testbench* adecuado, simule con GHDL y GTKWave.