

Ejercicio 1:

a) <u>MOV R0, R1:</u> R0 := R1	b) <u>ADD R0, R1:</u> ALU_IN1 := R0 ALU_IN2 := R1 ALU_add R0 := ALU_OUT (*)	c) <u>AND R0, R1:</u> ALU_IN1 := R0 ALU_IN2 := R1 ALU_and R0 := ALU_OUT (*)	d) <u>NOT R0:</u> ALU_IN1 := R0 ALU_dot R0 := ALU_OUT (*)	e) <u>NEG R0:</u> ALU_IN1 := R0 ALU_neg R0 := ALU_OUT (*)	(*) <u>MOV Z, ALU_Z</u> MOV C, ALU_C MOV N, ALU_N MOV V, ALU_V
-----------------------------------	---	---	--	--	---

Ejercicio 2:

2a) Para indicar el contenido de los registros IRO, IRI e IR2 luego de decodificar cada instrucción, debemos analizarlas en lenguaje máquina y luego asignar en base al ancho de palabra (16b). Usaré xxxx para decir que el ancho de dicho registro no se modifica luego de la decodificación (puede ser menor anterior por la longitud de la instrucción):

Instrucción	Lenguaje máquina	IRO	IR1	IR2
I) MOV R0, R1	0x1821	0x1821	xxxx	xxxx
II) MOV R0, 0xFF00	0x1800 0xFF00	0x1800	0xFF00	xxxx
III) MOV [0xAA00], 0xFF00	0x1200 0xAA00 0xFF00	0x1200	0xAA00	0xFF00
IV) MOV [0xAA00], [0xFF00]	0x1218 0xAA00 0xFF00	0x1218	0xAA00	0xFF00
V) MOV [0xAA00], [R0]	0x1230 0xAA00	0x1230	0xAA00	xxxx
VI) MOV [0xAA00], [R0+0xFF00]	0x1238 0xAA00 0xFF00	0x1238	0xAA00	0xFF00

b) I) <u>MOV R0, R1:</u> R0 := R1	II) <u>MOV R0, 0xFF00:</u> R0 := IRL
--------------------------------------	---

III) <u>MOV [0xAA00], 0xFF00:</u> MAR := IRI MDR := IR2 MEM_WRITE
--

IV) <u>MOV [0xAA00], [0xFF00]:</u> MAR := IR2 MEM_READ MAR := MDR MEM_READ MAR := IRL MEM_WRITE
---

V) <u>MOV [0xAA00], [R0]:</u> MAR := R0 MEM_READ MAR := IRL MEM_WRITE	VI) <u>MOV [0xAA00], [R0+0xFF00]:</u> ALU_IN1 := R0 ALU_IN2 := IR2 ALU_add MAR := ALU_OUT MEM_READ MAR := IRL MEM_WRITE
---	--