

Tarea 5

Interfaces de memoria

Integrantes:	Diego Hurtado
Profesor:	Luis Mateu
Auxiliar:	Jose Astorga
Ayudantes:	Francisco Muñoz
	Gabriel Montañana
	Pablo Polanco

1. Pregunta 1

1.1. a

i) ¿De cuantos kilobytes es el chip de memoria ROM?:

$$2^{15} * 2B = 64kB \text{ (aprox.)}$$

ii) ¿En qué rango de direcciones se ubica la ROM?:

Memoria	A19	A18	A17	A16	A15-A1	A0
Rom	1	1	1	1	X	0

Figura 1: Rango de direcciones de la ROM

$$[2^{19} + 2^{18} + 2^{17} + 2^{16}, 2^{20}[$$

$$[512k + 256k + 128k + 64k, 1024k[$$

$$[960k, 1024k[$$

iii) ¿De cuantos kilobytes es cada chip de memoria SRAM?

Ambos tienen la misma cantidad de memoria, la diferencia es que están conectados a distintos intervalos del bus de datos, pero tienen el mismo ancho. Por lo que, cada uno tiene:

$$2^{17} * 1B = 128kB \text{ (aprox.)}$$

iv) ¿En qué rango de direcciones se ubica la SRAM? (El conteo de RAM va de izquierda a derecha)

Ambas comparten el Chip Selector, por lo que trabajan en el mismo rango.

Memoria	A19	A18	A17-A1	A0
RAM1	0	0	X	0
RAM2	0	0	X	0

Figura 2: Rango de direcciones de las SRAM

$$[0, 2^{18}[$$

$$[0, 256k[$$

Cada SRAM es de 128k, y como son 2 llegan a los 256k, por lo que pueden manejar ese intervalo.

v) ¿Cuanta es la máxima cantidad de memoria, en kilobytes, que puede direccionar el procesador?

$$2^{19} * 2B = 1024kB \text{ (aprox.)}$$

vi) ¿Por qué el procesador no tiene la línea de dirección A0?

- Por alineamiento de memoria

vii) ¿Después de encenderse el procesador, qué puede decir acerca de la dirección de la primera instrucción que ejecuta este procesador? (Si se que se elimino, pero ya la tenia escrita :S)

- La primera instrucción que ejecuta un procesador lo lleva a algo llamado el "Reset Vector", que es un puntero o una dirección almacenado en memoria no volátil y que va a contener las instrucciones de la CPU. La dirección en la que esta depende del procesador que se tiene, por ejemplo, en el 8086 este se encuentra en la dirección física FFFF0h, que son 16 bytes abajo de 1 MB.

1.2. b

Agregue a este computador 384 kilobytes de memoria usando 2 chips de memoria SRAM de 256Kx8 en las direcciones [256KB, 640KB]. Evite a toda costa que esta memoria se seleccione más arriba de los 640 KB. En su dibujo no incluya la parte dada en la figura de más arriba. Incluya solamente la memoria que está agregando y su interfaz con el bus del procesador.

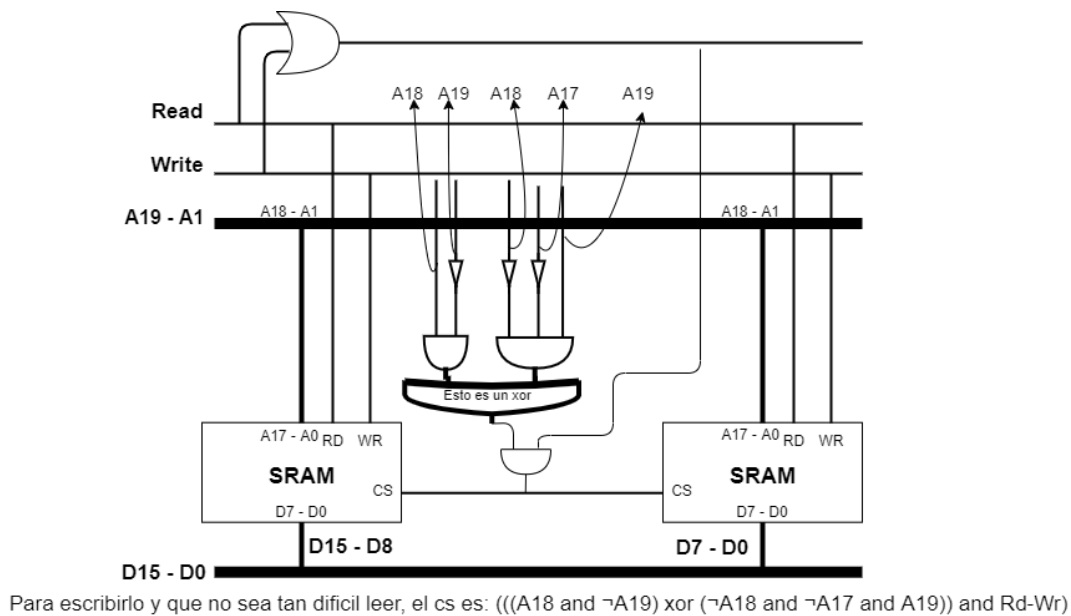


Figura 3: Extensión con SRAM

En vez de un xor podría ser un or, ya que nunca se van a poder activar los dos "and" al mismo tiempo (porque uno tiene A18 y el otro $\neg A18$), pero no influye en el resultado.

2. Pregunta 2

2.1. a

Modifique e implemente el módulo Y-SEL para que esta instrucción sea implementable. Su modificación debe ser tal que las actuales instrucciones de M32 continúen siendo implementables.

- Hay que modificar el modulo Y-SEL para que la instruccion LDRPC sea implementable, por lo que de alguna manera hay que pasarle a este modulo la instruccion. Actualmente tiene 4 posibles instrucciones en OP-Y-SEL, por lo que el multiplexor asociado a OP-Y-SEL es de 2 bits. Al agregarle la nueva instruccion @LDRPC a OP-Y-SEL, el multiplexor va a pasar a ser de 3 bits, con 8 posibles operaciones, lo que no quiere decir que tengamos que usarlas todas. Las instrucciones que se hacian antes van a seguir siendo compatibles porque los 32 bits no se tocan, y el unico cambio que conlleva agregarle una cifra al multiplexor es que ahora se le agrega un 0 antes a cada instruccion, pero viene por default. Para la nueva instruccion @LDRPC, se asocia el codigo binario de OP-Y-SEL 100. A esta entrada, para respetar que es un desplazamiento de 19 bits el que se quiere lograr, se conectan los 18 bits menos significativos de INST. En la figura vemos como quedaria el nuevo modulo Y-SEL:

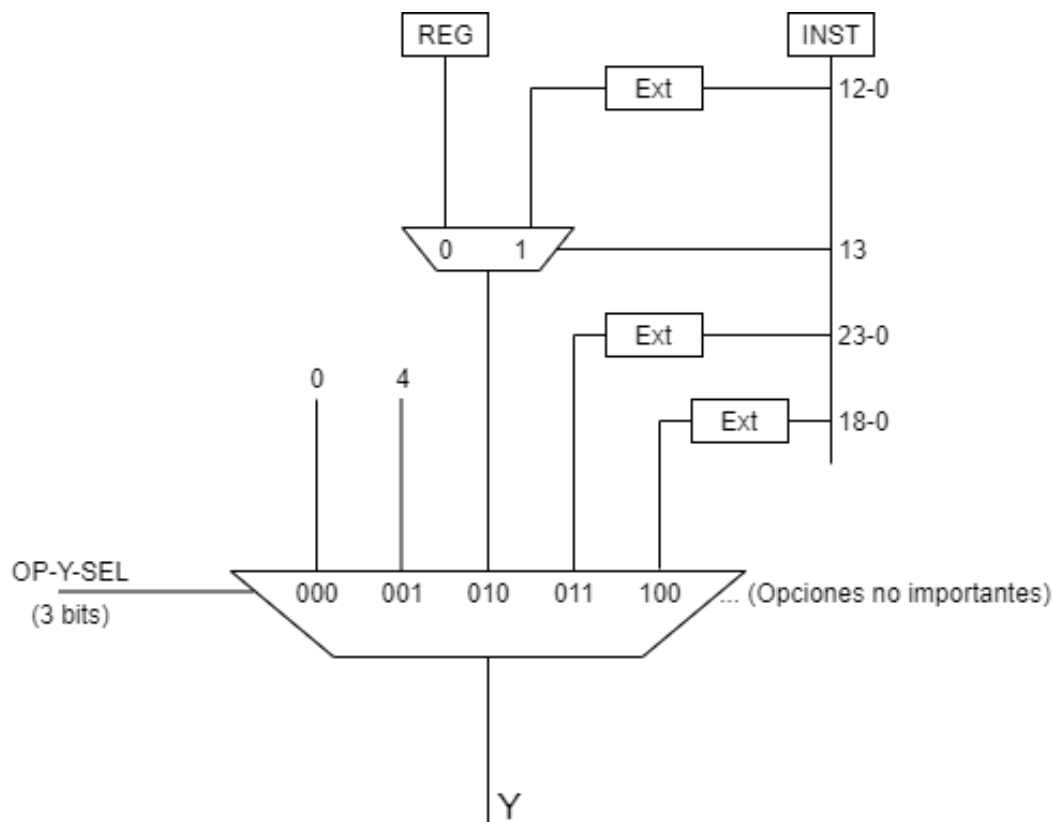


Figura 4: Nuevo Modulo Y-SEL, considerando la nueva opcion en OP-Y-SEL

2.2. b

Indique ciclo por ciclo las transferencias entre registros y las señales de control necesarias para ejecutar LDRPC (no incluya la fase de fetch ni la fase de decodificación).

Van a estar repartidas en 2 ejecuciones (Las que no se ponen, se asumen como 0):

1. Ex1: OP-ABI = @W, EN-A = 1, OP-Y-SEL = @LDRPC, OP-ALU = @ADD, WR-SR = 1, WR-AR = 1
2. Ex2: OP-ABI = @W, EN-A = 1, OP-DBI = @LDW, SEL-D = 1, WR-Rd = 1