



FUNDAMENTOS DE COMPUTADORES
18 de junio de 2013. Examen parcial del 2º cuatrimestre.

Nombre _____ DNI _____
Apellidos _____ Grupo _____

Ejercicio 1 (3 puntos). Supongamos que definimos que un número natural es “bonito” si es menor que cien mil y además su valor puede obtenerse como una suma de números naturales de la forma $1+2+3+4+5+\dots$

Se pide:

- (1,5 puntos) Escribir un programa en lenguaje ensamblador del ARM tal que dado un número natural N decida si es o no bonito. El programa escribirá en la variable B un 1 si el número es bonito y un 0 en caso contrario.
- (1,5 puntos) Convertir el código anterior en una subrutina que reciba como entrada un número natural N y devuelva como salida un 1 si el número N es bonito y un 0 si no lo es. Escribir un programa en lenguaje ensamblador del ARM que llame a la subrutina, y tal que dado un vector A de M números naturales sea capaz de hallar cuántos números bonitos hay en el vector. El programa debe almacenar la cantidad de números bonitos hallada en la variable “cuenta_bonitos”.

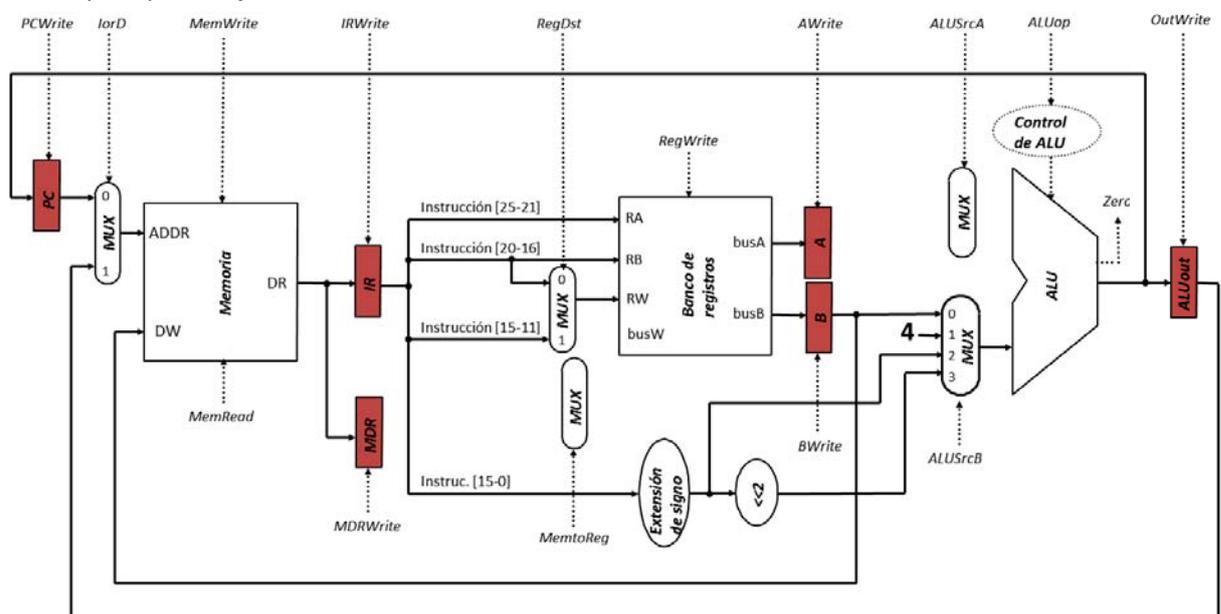
Nota: Se debe respetar el convenio del ARM visto en clase para llamadas a subrutinas. Además, en ambos apartados se deben incluir las directivas para reservar memoria y declarar las secciones (.data, .bss y .text) correspondientes.

Ejercicio 2 (3 puntos). Se desea añadir al procesador multiciclo la instrucción *movz*, de tipo R:

movz rd, rs, rt # si $(BR[rt]==0)$ entonces $BR[rd] \leftarrow BR[rs]$

Se pide:

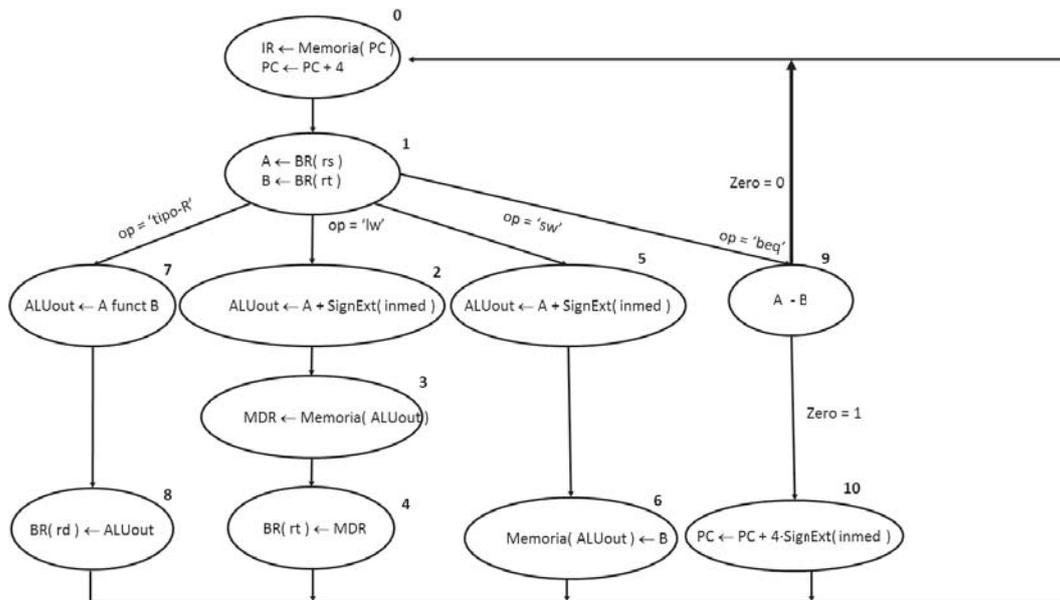
- (1 punto) Completar la ruta de datos de la figura con los elementos que faltan para que se puedan ejecutar las instrucciones del procesador MIPS estudiadas en clase, incluyendo además los cambios que tendrían que realizarse para poder ejecutar esta instrucción.



- (1 punto) Añadir los estados necesarios a la máquina de estados (ver figura siguiente) para poder implementar el control de esta instrucción.

- (1 punto) Indicar los valores que van tomando todos los registros implicados en la ejecución de la instrucción MOVZ R1, R2, R3. El código de operación de la función MOVZ es 000000 y el valor de los campos SHAMT y FUNCT son 00000 y 100110, respectivamente. El estado actual de la máquina es:

- R1=0x00000017
- R2=0x00110011
- R3=0x00000000
- PC=0x00003400



Ejercicio 3 (3 puntos). Sea un computador con una memoria principal de 1MB, con una memoria cache de emplazamiento directo de 1KB con bloques de 128 bytes. La memoria es direccionable por bytes y el tamaño de palabra es de 4 bytes. Se pide:

- (0,5 puntos) Indicar el formato de la dirección para MP y para MC.
- (1 punto) En un momento dado sólo cuatro de las entradas de la cache contienen datos válidos, los contenidos del array de etiquetas de la cache (en hexadecimal) son los indicados en tabla inferior. Expresar en hexadecimal el rango de direcciones de memoria principal ubicadas en la memoria cache.

Bloque de MC	Etiqueta
0	
1	
2	0x3A7
3	
4	0x100
5	0x100
6	
7	0x0D2

- (1,5 puntos) Partiendo de la situación indicada en la tabla anterior, supongamos que un programa realiza la siguiente cadena de referencias en lectura (en hexadecimal): todas las palabras entre E9C38 y E9D7C ambos incluidos, todas las palabras entre las direcciones 4019C y 402AC, ambas incluidas, y todas las palabras entre las direcciones 34A84 y 34BC0, ambas incluidas. Indicar en cada caso los fallos que se producen e indicar cómo se queda el array de etiquetas de la memoria cache después de acceder a la última dirección.

Ejercicio 4 (1 punto). El siguiente código en lenguaje ensamblador del ARM puede usarse para generar un retardo que permita la eliminación de rebotes en un pulsador, tal como hemos hecho por ejemplo en el laboratorio.

```

LDR R2,=200000
MOV R3,#0
bReb:  CMP R3,R2
      BEQ finR
      MUL R1,R0,R1
      ADD R3,R3,#1
      B   bReb

```

finR: (salida del bucle, continua el programa)

- (0,75 puntos) Si sabemos que la frecuencia de reloj del ARM es de 66 MHz y hemos medido que desde que se produce la entrada en el bucle hasta que se produce la salida transcurren 2 décimas de segundo ¿cuál ha sido el valor de CPI en la ejecución del bucle?
- (0,25 puntos) Si en las mismas condiciones aumentamos la frecuencia del reloj hasta 100 MHz ¿cuánto tiempo transcurrirá desde que se entra en el bucle hasta que se sale?