FUNDAMENTOS DE COMPUTADORES
18 de junio de 2014. Examen parcial del 2º cuatrimestre.

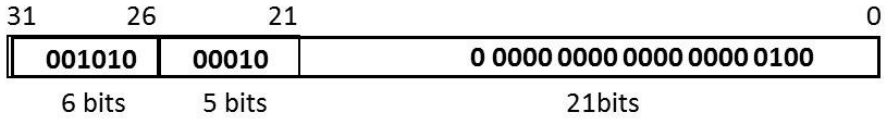
Nombre_____DNI_____
__

Apellidos_____Grupo_____

Ejercicio 1 (2.5 puntos) Para el computador MIPS estudiado en clase, responder a las siguientes preguntas:

a) (0.5 puntos) Qué instrucción/es, del repertorio de instrucciones del procesador MIPS estudiado en clase, se ven afectada/s y no se podrían ejecutar si se elimina de la ruta de datos **el registro MDR, la entrada 1 del Mux que selecciona el dato que se escribe en el banco de registros, y por lo tanto dicho Multiplexor.**

b) (0.75 puntos) Partiendo de la ruta de datos completa del MIPS, se desea añadir una nueva instrucción Store con Direccionamiento Absoluto (SMDA). Se añade para ello un nuevo formato de instrucción con los siguientes campos (Op, Rs, dirección):



El comportamiento de la instrucción SMDA sería el siguiente:

SMDA Rs, dirección **Mem[ExtCeros(dirección)] ← BR[rs]**

donde **ExtCeros(dirección)** extiende la dirección de 21 bits a 32 bits añadiendo ceros por la izquierda. Añade a la ruta de datos los cambios que tendrían que realizarse para poder ejecutar esta instrucción.

c) (0.5 puntos) Añade los cambios necesarios en el diagrama de transición de estados del controlador para poder ejecutar correctamente esta nueva instrucción.

d) (0.75 puntos) Indicar los cambios necesarios en la tabla de verdad del controlador (añadir las filas y columnas necesarias).

Ejercicio 2 (3 puntos) Sea un computador con un procesador ARMv4. EL procesador cuenta con una memoria cache de emplazamiento directo de 256 bytes con bloques de 64 bytes, unificada para datos e instrucciones (se almacenan en la misma cache los bloques de instrucciones accedidos en la etapa FETCH y los bloques de datos accedidos durante las etapas que hacen acceso a memoria en las instrucciones ldr y str).

a) (0.5 puntos) En dicho sistema se quiere ejecutar el siguiente programa. Explique razonadamente lo que hace el código.

```
.equ  N, 16

.data
A: .word N valores enteros separados
por comas
B: .word N valores enteros separados
por comas

.text
start:   ldr r0, =A
         ldr r1, =B
         mov r4, #N
L1:      ldr r2, [r0]
         ldr r3, [r1]
         add r2, r2, r3
         and r2, r2, #0xF
         str r2, [r0]
         add r0, r0, #4
         add r1, r1, #4
         sub r4, #1
         cmp r4, #0
         bne L1
end:     b .
```

b) (0.25 puntos) Con direcciones de 32 bits, indicar el formato de la dirección para MP y para la MC.

c) (0.5 puntos) El programa se enlaza ubicando la sección .data a partir de la dirección 0x0C000000, y la sección .text se coloca a continuación de la sección .data. Obtenga los rangos de direcciones que ocupan el array A, el array B y las instrucciones, indicando para cada rango el/los bloques de memoria correspondientes, y el marco de bloque (bloque de cache) y la etiqueta asociados.

d) (1 punto) Determine el número de aciertos y fallos de cache que se producirían al ejecutar el código anterior, hasta la primera vez que se ejecuta la instrucción que está en la etiqueta end.

e) (0.75 puntos) Suponga que el programador cambia el valor N por 32. Obtenga de nuevo los bloques de memoria en que se ubicarían los datos y las instrucciones. Explique cualitativa y razonadamente cómo afectaría este cambio a los fallos de cache (No es necesario calcular el nuevo número de fallos, sólo explicar lo que sucedería).

Ejercicio 3 (3 puntos) Dado un vector V de N componentes se dice que es Melchoriforme si posee al menos un elemento Rubio. Un elemento V[i] es Rubio si satisface la siguiente expresión:

$$\sum_{j=0}^{N-1} v[j] = 2 * v[i]$$

Se pide:

a) (1 punto) Una subrutina SumaVector(V, N) que sume los N elementos del vector V, respetando el convenio de llamadas a subrutinas visto en clase.

b) (2 puntos) Un programa que dado un vector V y su dimensión N decida si es Melchoriforme, utilizando la subrutina SumaVector.

Ejercicio 4 (1.5 puntos) El código de un determinado programa está constituido por instrucciones máquina con la siguiente frecuencia de ejecución: aritméticas 50%, carga 20%, almacenamiento 10% y salto condicional 20% (donde la mitad de los saltos se toman). En cierto procesador las instrucciones consumen los siguientes ciclos: aritméticas 4, carga 5, almacenamiento 4, salto no tomado 3, salto tomado 4.

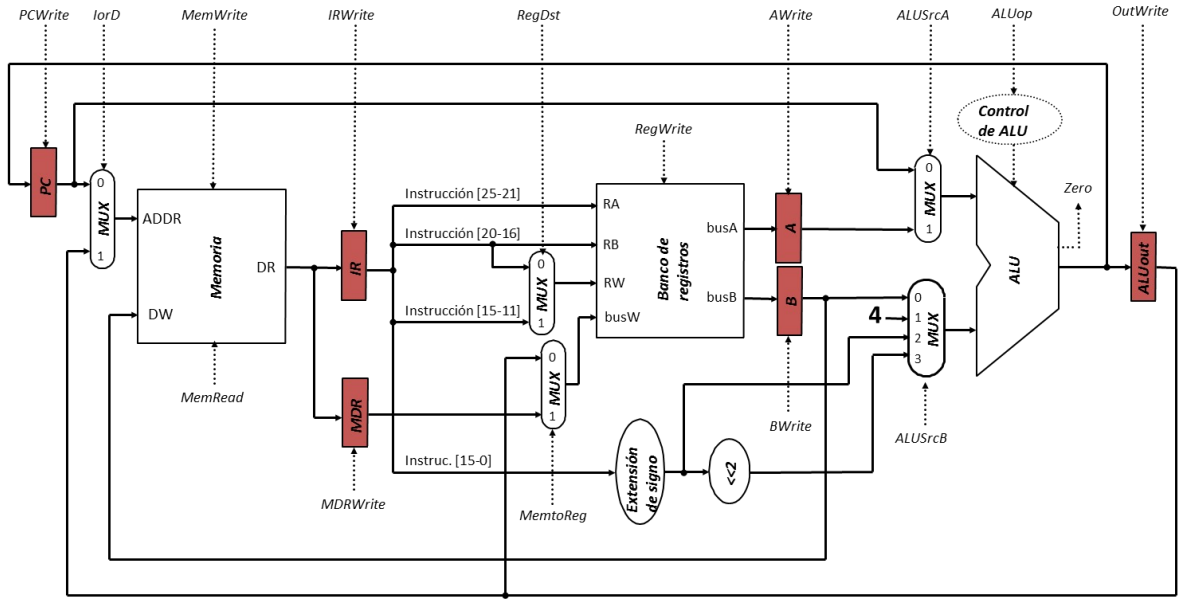
a) (0.5 puntos) Calcula el CPI del procesador al ejecutar este programa.

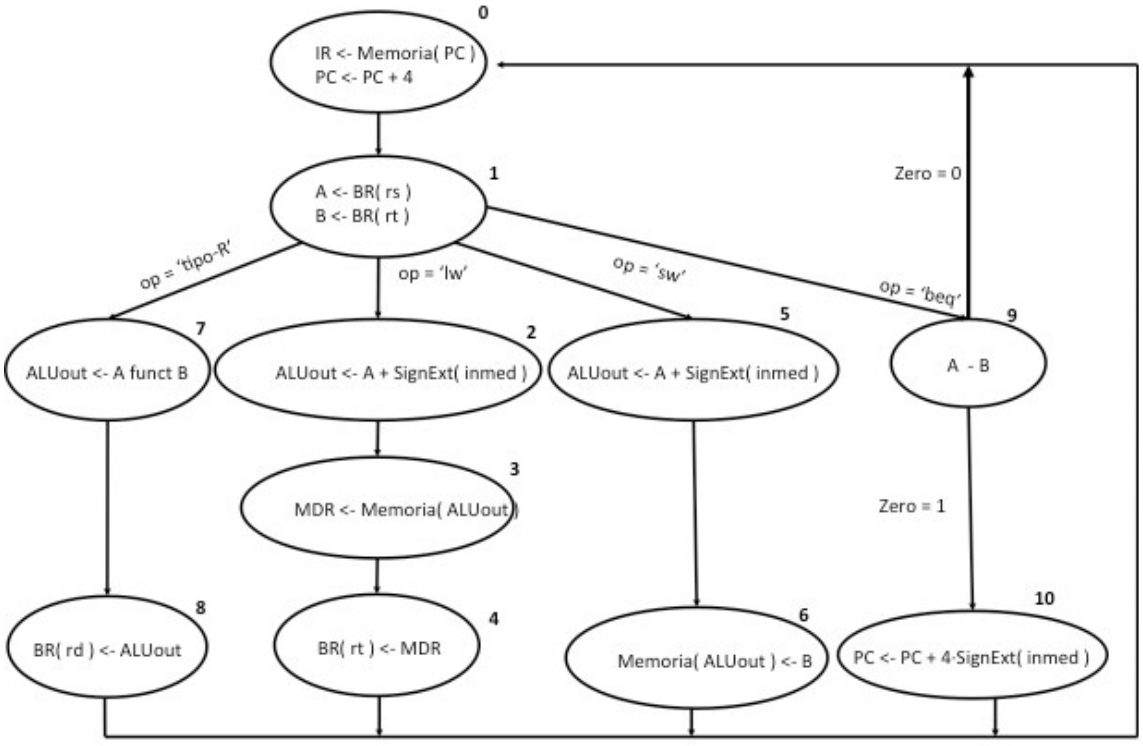
b) (0.5 puntos) Sabiendo que el programa se ha ejecutado en 8.2ms, y que la frecuencia del procesador es de 1GHz, calcula el número de instrucciones que se han ejecutado.
c) (0.5 puntos) Suponiendo que por cada 100 instrucciones ejecutadas se produce 1 fallo de cache, que implica una penalización de 300 ciclos de reloj, calcula el nuevo tiempo de ejecución del programa. ¿Cuál sería entonces el nuevo CPI?

FUNDAMENTOS DE COMPUTADORES
18 de junio de 2014. Examen parcial del 2º cuatrimestre.

Nombre_____DNI_____

Apellidos_____Grupo_____





Estado actual	op	Zero	Estado siguiente	IRWrite	PCWrite	AWrite	BWrite	ALUSrcA	ALUScrB	ALUOp	OutWrite	MemWrite	MemRead	IorD	MDRWrite	MemtoReg	RegDest	RegWrite
0000	XXXXXX	X	0001	1	1			0	01	00 (add)		0	1	0				0
0001	100011 (lw)	X	0010															
0001	101011 (sw)	X	0101															
0001	000000 (tipo-R)	X	0111	0	0	1	1					0	0					0
0001	000100 (beq)	X	1001															
0010	XXXXXX	X	0011	0	0			1	10	00 (add)	1	0	0					0
0011	XXXXXX	X	0100	0	0							0	1	1	1			0
0100	XXXXXX	X	0000	0	0							0	0			1	0	1
0101	XXXXXX	X	0110	0	0		0	1	10	00 (add)	1	0	0					0
0110	XXXXXX	X	0000	0	0							1	0	1				0
0111	XXXXXX	X	1000	0	0			1	00	10 (funct)	1	0	0					0
1000	XXXXXX	X	0000	0	0							0	0			0	1	1
1001	XXXXXX	0	0000	0	0			1	00	01 (sub)		0	0					0
1001	XXXXXX	1	1010															
1010	XXXXXX	X	0000	0	1			0	11	00 (add)		0	0					0

FUNDAMENTOS DE COMPUTADORES
18 de junio de 2014. Examen final .

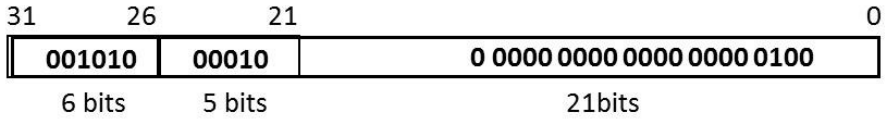
Nombre_____DNI_____
_

Apellidos_____Grupo_____

Ejercicio 4 (2.5 puntos) Para el computador MIPS estudiado en clase, responder a las siguientes preguntas:

c) (0.5 puntos) Qué instrucción/es, del repertorio de instrucciones del procesador MIPS estudiado en clase, se ven afectada/s y no se podrían ejecutar si se elimina de la ruta de datos **el registro MDR, la entrada 1 del Mux que selecciona el dato que se escribe en el banco de registros, y por lo tanto dicho Multiplexor.**

d) (0.75 puntos) Partiendo de la ruta de datos completa del MIPS, se desea añadir una nueva instrucción Store con Direccionamiento Absoluto (SMDA). Se añade para ello un nuevo formato de instrucción con los siguientes campos (Op, Rs, dirección):



El comportamiento de la instrucción SMDA sería el siguiente:

SMDA Rs, dirección Mem[ExtCeros(dirección)] ← BR[rs]

donde **ExtCeros(dirección)** extiende la dirección de 21 bits a 32 bits añadiendo ceros por la izquierda. Añade a la ruta de datos los cambios que tendrían que realizarse para poder ejecutar esta instrucción.

e) (0.5 puntos) Añade los cambios necesarios en el diagrama de transición de estados del controlador para poder ejecutar correctamente esta nueva instrucción.

f) (0.75 puntos) Indicar los cambios necesarios en la tabla de verdad del controlador (añadir las filas y columnas necesarias).

Ejercicio 5 (3 puntos) Sea un computador con un procesador ARMv4. EL procesador cuenta con una memoria cache de emplazamiento directo de 256 bytes con bloques de 64 bytes, unificada para datos e instrucciones (se almacenan en la misma cache los bloques de instrucciones accedidos en la etapa FETCH y los bloques de datos accedidos durante las etapas que hacen acceso a memoria en las instrucciones ldr y str).

f) (0.5 puntos) En dicho sistema se quiere ejecutar el siguiente programa. Explique razonadamente lo que hace el código.

```
.equ  N, 16

.data
A: .word N valores enteros separados
por comas
B: .word N valores enteros separados
por comas

.text
start:   ldr r0, =A
         ldr r1, =B
         mov r4, #N
L1:      ldr r2, [r0]
         ldr r3, [r1]
         add r2, r2, r3
         and r2, r2, #0xF
         str r2, [r0]
         add r0, r0, #4
         add r1, r1, #4
         sub r4, #1
         cmp r4, #0
         bne L1
end:     b .
```

g) (0.25 puntos) Con direcciones de 32 bits, indicar el formato de la dirección para MP y para la MC.

h) (0.5 puntos) El programa se enlaza ubicando la sección .data a partir de la dirección 0x0C000000, y la sección .text se coloca a continuación de la sección .data. Obtenga los rangos de direcciones que ocupan el array A, el array B y las instrucciones, indicando para cada rango el/los bloques de memoria correspondientes, y el marco de bloque (bloque de cache) y la etiqueta asociados.

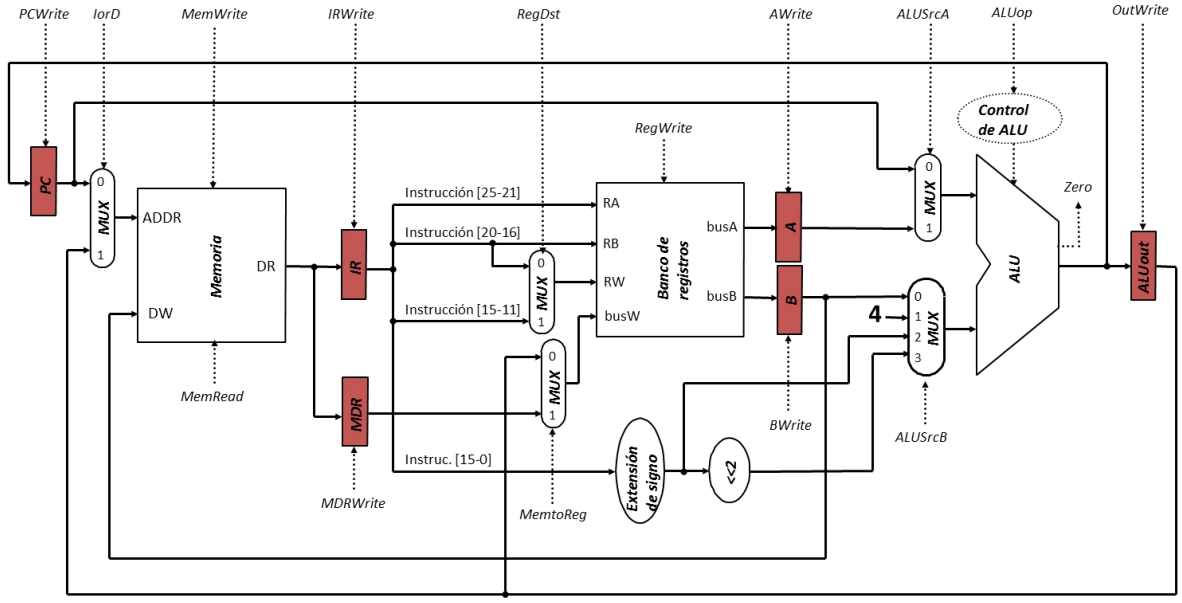
i) (1 punto) Determine el número de aciertos y fallos de cache que se producirían al ejecutar el código anterior, hasta la primera vez que se ejecuta la instrucción que está en la etiqueta end.

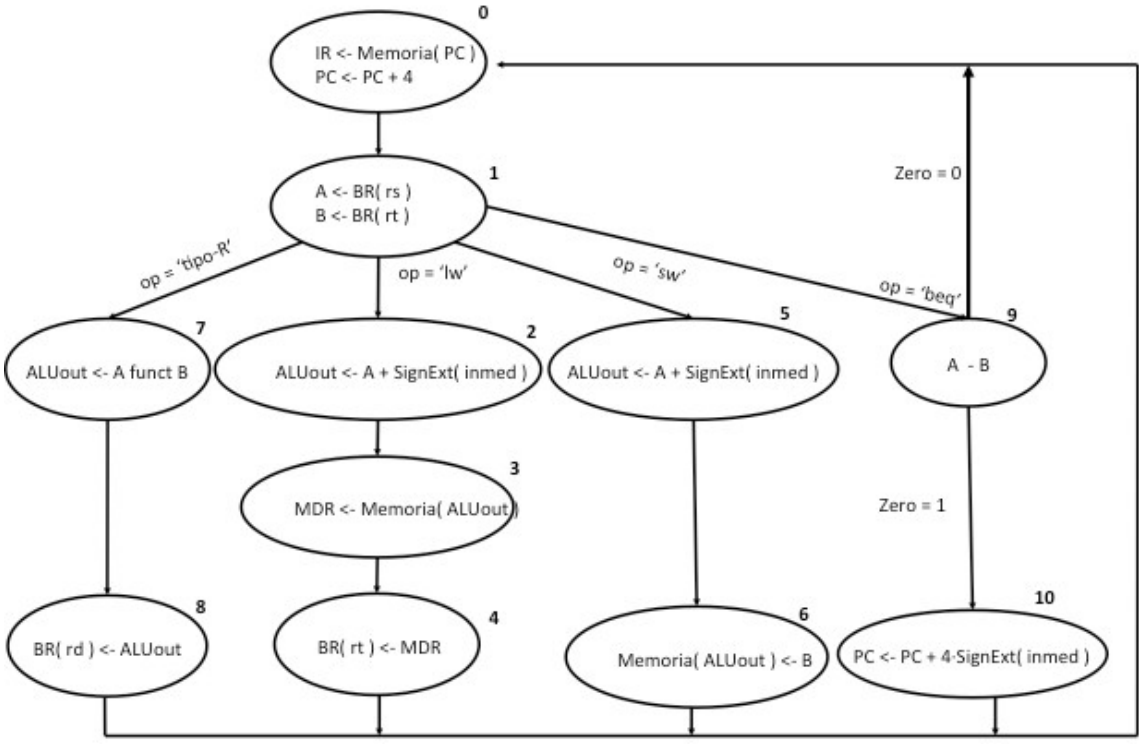
j) (0.75 puntos) Suponga que el programador cambia el valor N por 32. Obtenga de nuevo los bloques de memoria en que se ubicarían los datos y las instrucciones. Explique cualitativa y razonadamente cómo afectaría este cambio a los fallos de cache (No es necesario calcular el nuevo número de fallos, sólo explicar lo que sucedería).

FUNDAMENTOS DE COMPUTADORES
18 de junio de 2014. Examen final.

Nombre_____DNI_____

Apellidos_____Grupo_____





Estado actual	op	Zero	Estado siguiente	IRWrite	PCWrite	AWrite	BWrite	ALUSrcA	ALUScrB	ALUOp	OutWrite	MemWrite	MemRead	IorD	MDRWrite	MemtoReg	RegDest	RegWrite
0000	XXXXXX	X	0001	1	1			0	01	00 (add)		0	1	0				0
0001	100011 (lw)	X	0010															
0001	101011 (sw)	X	0101	0	0	1	1					0	0					0
0001	000000 (tipo-R)	X	0111															
0001	000100 (beq)	X	1001															
0010	XXXXXX	X	0011	0	0			1	10	00 (add)	1	0	0					0
0011	XXXXXX	X	0100	0	0							0	1	1	1			0
0100	XXXXXX	X	0000	0	0							0	0			1	0	1
0101	XXXXXX	X	0110	0	0		0	1	10	00 (add)	1	0	0					0
0110	XXXXXX	X	0000	0	0							1	0	1				0
0111	XXXXXX	X	1000	0	0			1	00	10 (funct)	1	0	0					0
1000	XXXXXX	X	0000	0	0							0	0			0	1	1
1001	XXXXXX	0	0000	0	0			1	00	01 (sub)		0	0					0
1001	XXXXXX	1	1010															
1010	XXXXXX	X	0000	0	1			0	11	00 (add)		0	0					0