

Apellidos y Nombre

DNI

Grupo

--	--	--

- 1 (1.75 puntos)** La figura 1 muestra la ruta de los datos monociclo básica del procesador MIPS R2000, vista en clase. Suponga que se desea añadir la instrucción **slt rd, rs, rt (set on less than)** como una instrucción más del juego de instrucciones. El cometido de la instrucción es el siguiente:

if (rs < rt), rd ← 1
else rd ← 0

Nota: se sugiere verificar la desigualdad $rs < rt$ a partir de una resta, $(rs - rt < 0)$, y comprobando los flags adecuados

El formato de la instrucción es de tipo R, con los siguientes campos:

000000	RS	RT	RD	00000	011010
--------	----	----	----	-------	--------

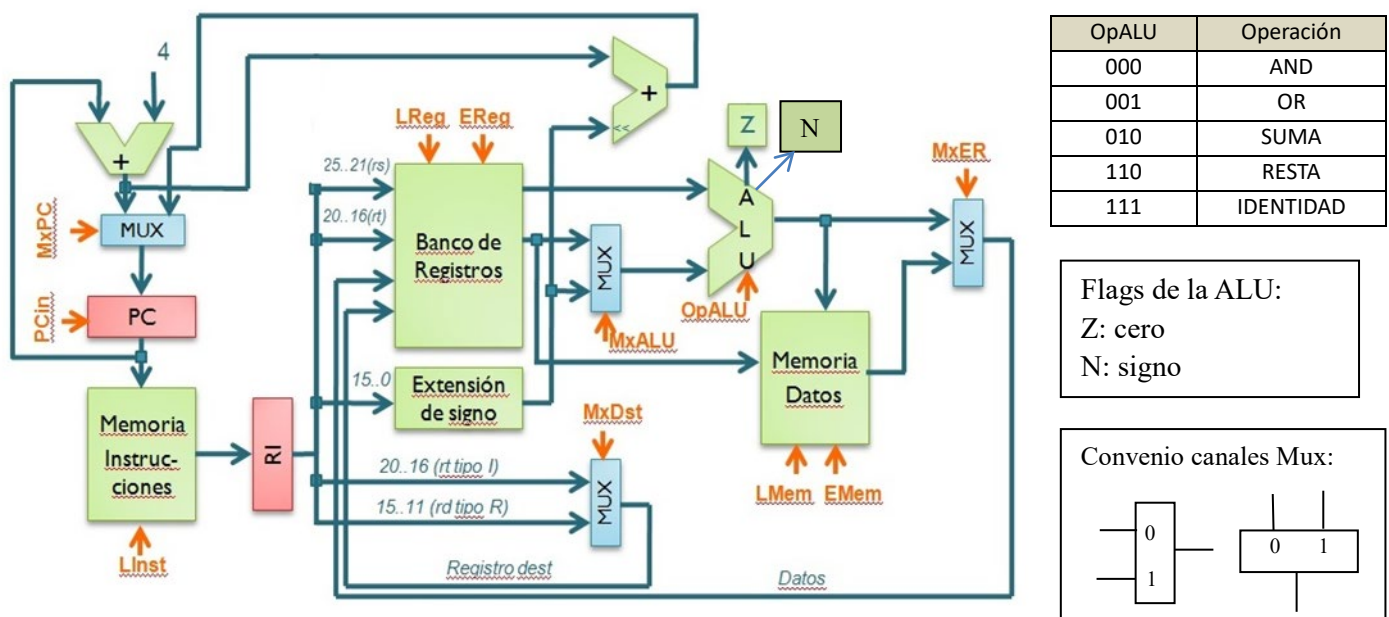


Figura 1

- a) **(0.5 puntos)** Indique los cambios que habría que introducir en la ruta de datos (elementos nuevos, o modificación/extensión de elementos ya existentes)

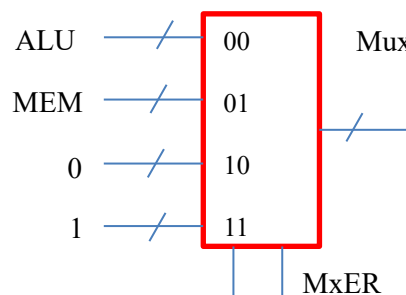
El multiplexor MxER se extiende con 2 canales más y dos señales de selección:

00 → salida de la ALU

01 → salida de la memoria de datos

10 → 0 (0.....00)

11 → 1 (0.....01)



- b) **(1 punto)** Complete la tabla correspondiente a las señales de control para ejecutar sobre la misma las instrucciones que se detallan. En binario.

Instrucción	Form	EReg	OpALU	LMem	EMem	MxALU	MxDst	MxER	MxPC
add rd,rs,rt	R	1	010	0	0	0	1	00	0
lw rt, desp(rs)	I	1	010	1	0	1	0	01	0
beq rs, rt, etiq	I	0	110	0	0	0	X	XX	Z
slt rd,rs,rt	R	1	110	0	0	0	1	1N	0

Para ejecutar la instrucción slt, se hace la resta en la ALU, y dependiendo del flag N:

Si N=0 (resultado positivo, $R_s \geq R_t$), MxER = 10, Rd \leftarrow 0

Si N=1 (resultado negativo, $R_s < R_t$), MxER = 11, Rd \leftarrow 1

- c) **(0.25 puntos)** Se pretende incluir también la versión inmediata de slt, **slti rt, rs, inm**

if (rs < inm), rt \leftarrow 1

else rt \leftarrow 0

001010	RS	RT	Inm
--------	----	----	-----

Rellene la tabla con las señales de control necesarias. En binario.

Instrucción	Form	EReg	OpALU	LMem	EMem	MxALU	MxDst	MxER	MxPC
slti rt,rs,inm	I	1	110	0	0	1	0	1N	0

- 2 (0.75 puntos)** Un cierto procesador se ha segmentado en 5 etapas (LI, DI, EX, M, ER), cuyos retardos son los siguientes: 25ns, 15ns, 30ns, 25ns y 15ns. Suponiendo que el retardo de los registros de segmentación es de 10ns, se pide:

- a) **(0,25 puntos)** Frecuencia de reloj a la que trabajará el procesador segmentado y productividad máxima que podrá alcanzar.

$$\tau = \max\{25, 15, 30\} + 10 = 40 \text{ ns} \rightarrow f_{\max} = 1/\tau = 1000/40 \text{ MHz} = 25 \text{ MHz}$$

$$X(\infty) = 1 \text{ instrucción por ciclo} = 1/\tau = 25 \text{ MIPS}$$

- b) **(0,25 puntos)** Frecuencia de reloj a la que trabajará el procesador NO-segmentado y productividad máxima que podrá alcanzar.

$$T_{NS} = 25 + 15 + 30 + 25 + 15 = 110 \text{ ns} \rightarrow f_{\max} = 1/T_{NS} = 1000/110 \text{ MHz} = 9.09 \text{ MHz}$$

$$X = 1 \text{ instrucción por ciclo} = 1/T_{NS} = 9.09 \text{ MIPS}$$

- c) **(0,25 puntos)** Suponiendo que el procesador segmentado se rediseña como un procesador SUPERSEGMENTADO subdividiendo la etapa EX en dos etapas de 15ns, ¿qué productividad máxima se podría alcanzar?

6 etapas, con retardos: 25, 15, 15, 15, 25, 15

$$\tau = \max\{\tau_i\} + 10 = 35 \text{ ns}$$

$$X(\infty) = 1 \text{ instrucción por ciclo} = 1/\tau = 28.6 \text{ MIPS}$$

- 3 (1.5 puntos)** Considere la siguiente secuencia de programa en ensamblador de MIPS R2000, el cual se ejecuta en cierta ruta de datos que se halla segmentada en cinco etapas (LI, DI, EX, M, ER):

```
(1)      lw $2,0($4)
(2)      beq $2,$3, etiqueta
(3)      add $1,$2,$3
(4)      sub $4,$1,$3
(5)      lw $5,100($4)
(6)      sw $5,50($4)
        ...
etiqueta : ...
```

- a) **(0.5 puntos)** Rellene el diagrama instrucciones/tiempo desde la **instrucción 1 a la 4** suponiendo que el salto NO tiene lugar. Asuma que los conflictos por dependencias de datos y control se solucionan mediante la inserción de instrucciones NOP y que la latencia de salto es 2.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	LI	DI	EX	M	ER															
nop		LI	DI	EX	M	ER														
nop			LI	DI	EX	M	ER													
beq				LI	DI	EX	M	ER												
nop					LI	DI	EX	M	ER											
nop						LI	DI	EX	M	ER										
add							LI	DI	EX	M	ER									
nop								LI	DI	EX	M	ER								
nop									LI	DI	EX	M	ER							
sub										LI	DI	EX	M	ER						

- b) **(0.5 puntos)** Rellene el diagrama instrucciones/tiempo desde **la instrucción 1 a la 4** suponiendo que el salto NO tiene lugar. Asuma que los conflictos por dependencias de datos se solucionan mediante la inserción de instrucciones NOP y ahora los conflictos de control se resuelven mediante predicción de salto NO efectivo.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	LI	DI	EX	M	ER															
nop		LI	DI	EX	M	ER														
nop			LI	DI	EX	M	ER													
beq				LI	DI	EX	M	ER												
add					LI	DI	EX	M	ER											
nop						LI	DI	EX	M	ER										
nop							LI	DI	EX	M	ER									
sub								LI	DI	EX	M	ER								

- c) **(0.5 puntos)** Compare razonadamente ambas soluciones empleadas en la resolución de los conflictos de control desde el punto de vista de las prestaciones. Incluya en la comparación los casos en el que el salto SI es efectivo.

Inserción de instrucciones NOP: 14 ciclos siempre

Predicción de salto NO efectivo: 12 ciclos en caso de acierto (el salto no tiene lugar), y 14 ciclos en caso de no acierto (el salto es efectivo). En caso de no acierto en la predicción, el pipeline debe vaciarse de las 2 instrucciones que se han introducido detrás de beq. Asumimos que esta penalización conlleva dos ciclos, igual a la latencia de salto.

Por tanto, la técnica de predicción siempre mejora las prestaciones respecto a la inserción sistemática de instrucciones NOP. El nivel de mejora depende de la frecuencia con la que se cumple la predicción (en este caso, que el salto no tenga lugar, depende del valor de los registros \$2 y \$3).

4. (1 punto) La figura 2 muestra un sumador para números enteros de 64 bits, basado en la técnica CSA (Carry Select Adder). Para ello, utiliza sumadores CPA (Carry Propagated Adder) como los vistos en clase de 8 y 16 bits. Asumiendo que cada CPA está compuesto por celdas sumadoras tipo FA (Full Adder) que implementan la Suma y el Acarreo a partir de sus funciones lógicas y el retardo de puerta es de 1ns (para todo tipo de puerta) y que cada uno de los multiplexores tiene un retardo correspondiente a 3 puertas lógicas, responda:

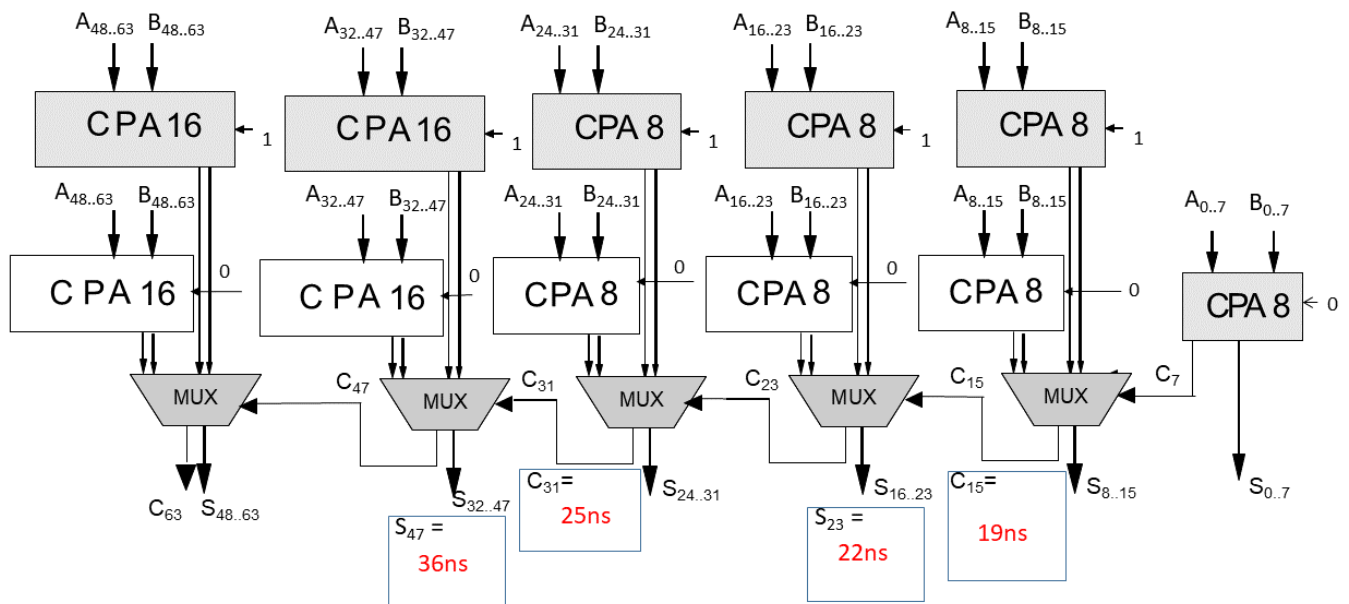


Figura 2

- a) **(0.6 puntos)** Tiempo necesario para realizar la suma completa. Justifíquelo completando en la figura los tiempos intermedios en el recuadro a propósito.

T = 38 ns

- b) **(0.4 puntos)** ¿Cuál es la aceleración de este sumador CSA respecto de un único sumador CPA de 64 bits?

$$S = T_{CSA} / T_{CPA} = 129/38 = 3,34$$

5. (1 punto) La Figura 3 muestra tres operadores de multiplicación para enteros con signo de 32 bits. El operador 1 es un multiplicador combinacional cuyo retardo es de 100 ns. El operador 2 es secuencial como el visto en clase e implementa el algoritmo de Booth, utilizando una señal de reloj cuyo periodo es 6 ns; el operador 3 también es secuencial y, en este caso implementa el algoritmo de recodificación por parejas, trabajando con un reloj de 125 MHz de frecuencia. Especifique para cada uno de los circuitos cuál es su productividad máxima. Asuma que los circuitos 2 y 3 requieren un ciclo específico para la inicialización.

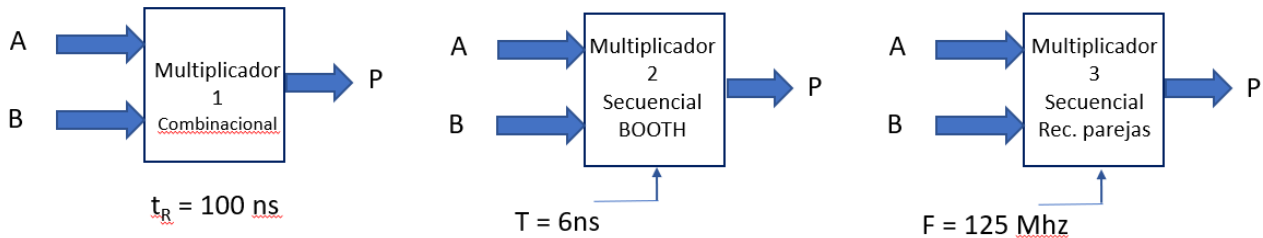


Figura 3

Operador 1	Productividad = $1/t_R = 10 \text{ MOPS}$
Operador 2	Productividad = $1/33 \text{ ciclos} * 6 \text{ ns} = 5,05 \text{ MOPS}$
Operador 3	Productividad = $1/17 \text{ ciclos} * 8 \text{ ns} = 7,35 \text{ MOPS}$

6. (1 punto) La Figura 4 se corresponde con el esquema de bloques de un circuito para soporte de la instrucción de coma flotante cvt.d.s fd, fs. Inserta en los huecos A, B, C, D y E los valores que faltan y justifícalos.

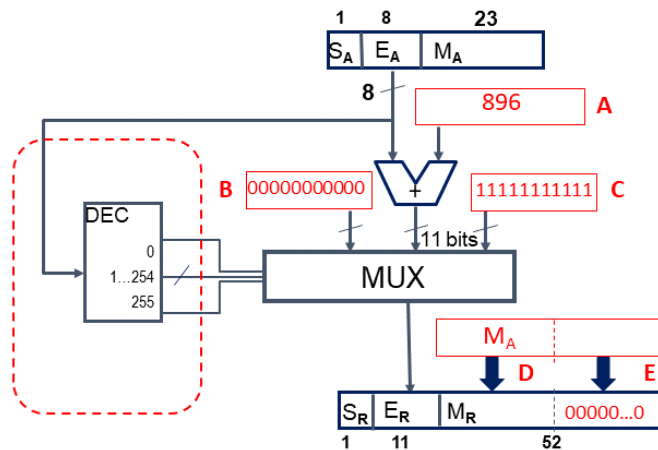


Figura 4

c) Justificación **(0.8 puntos)**:

A	Lo que falta para llegar al exceso a 1023 desde exceso 127
B	Valores especiales con exponente 0
C	Valores especiales con exponente 1s
D	Mantisa de A
E	Completamos con 0 los bits de menor peso

d) **(0.2 puntos)** Explica el cometido del decodificador de 8 a 256 remarcado con las líneas punteadas

El decodificador sirve para identificar y tratar a parte el exponente de los valores especiales (valores con exponente todo 0s ó todo 1s) y no sumarles 896

7. (1.5 puntos) Se desea lanzar al mercado un nuevo modelo de placa “Blueberry Phi” basada en un procesador ARM de 64 bits, capaz de direccionar hasta 4Gbytes, aunque la primera versión comercial incluirá solo 2 Gbytes de SDRAM de tipo DDR. El diseño del mapa de memoria de esta primera versión se va a hacer mediante la inclusión de dos módulos idénticos de 1 Gbyte cada uno, respondiendo a las siguientes especificaciones:

- a) **(0.8 puntos)** El primer módulo (M1) será DIMM de 184c formado internamente por chips de 128 Mbytes y se emplazará en la dirección más baja (0x00000000). Responda a las siguientes cuestiones en relación a este módulo M1:

Número de chips de memoria que contiene el módulo	8
Número y nombre (denomínelas Ai) de líneas de dirección que toma el controlador de memoria para las direcciones del módulo	A3 A29 , 27 líneas
Número de líneas de habilitación de byte necesarias. ¿A qué pines del módulo deberán conectarse?	8, a los pines DMQ*(0...7)
¿Cuál es la función de selección del módulo <i>utilizando lógica negativa</i> ? ¿Cuál es el nombre que normalmente recibe el pin donde se ha de conectar?	A31 + A30, conectadas a CS*

- b) **(0.4 puntos)** El segundo módulo (M2) es idéntico al primero, pero se situará en las direcciones más altas del mapa de memoria. Indíquese para este módulo:

Dirección inicial y final (exprésela en hexadecimal)	Inicial :0xC0000000 final: 0xFFFFFFFF
Función de selección para este módulo <i>utilizando lógica negativa</i> .	A31* + A30*, conectadas a CS*

- c) **(0.3 puntos)** Este mapa de memoria dispone de una zona libre para futuras versiones en las que se incluirá más memoria. Indíquese:

Dirección inicial y final del espacio libre	Inicial :0x40000000 final: 0xBFFFFFFF
---	---------------------------------------

8. (1.5 puntos) Los chips utilizados en los módulos M1 y M2 del ejercicio anterior son de tipo DDR con los siguientes parámetros temporales t_{CL} - t_{RCD} - t_{RP} - t_{RAS} 2-2-2-6 y funcionan a una frecuencia de 200MHz. Se organizan internamente en un único banco de 1024 filas, siendo el tamaño de bloque de 8 palabras. Responda a las siguientes cuestiones justificando la respuesta:

a) **(0.4 puntos)** ¿Cómo se interpretan de manera física y lógica los bits de direcciones?

Dir. Física	Fila	Columna	Dir. Lógica	Bloque	Desp.
	10 bits	17 bits		24 bits	3 bits

b) **(0.3 puntos)** ¿Cuál es el tiempo de acceso?

$T_{acceso} = TRCD + CL = 4 \text{ ciclos} = 4 * 5ns = 20 \text{ ns}$

a) **(0.4 puntos)** ¿Cuántos ciclos se necesitan para leer un bloque de datos? Justifíquelo en el cronograma asociado.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27
Orden	ACT		RD																								
Dirección	fila		col																								
Datos						W0W1	W2W3	W4W5	W6W7																		

8 ciclos

b) **(0.4 puntos)** ¿Cuál es el ancho de banda de los módulos M1 y M2?

$B = 8bytes * 200MHz * 2 = 3.2 \text{ GB/s}$