

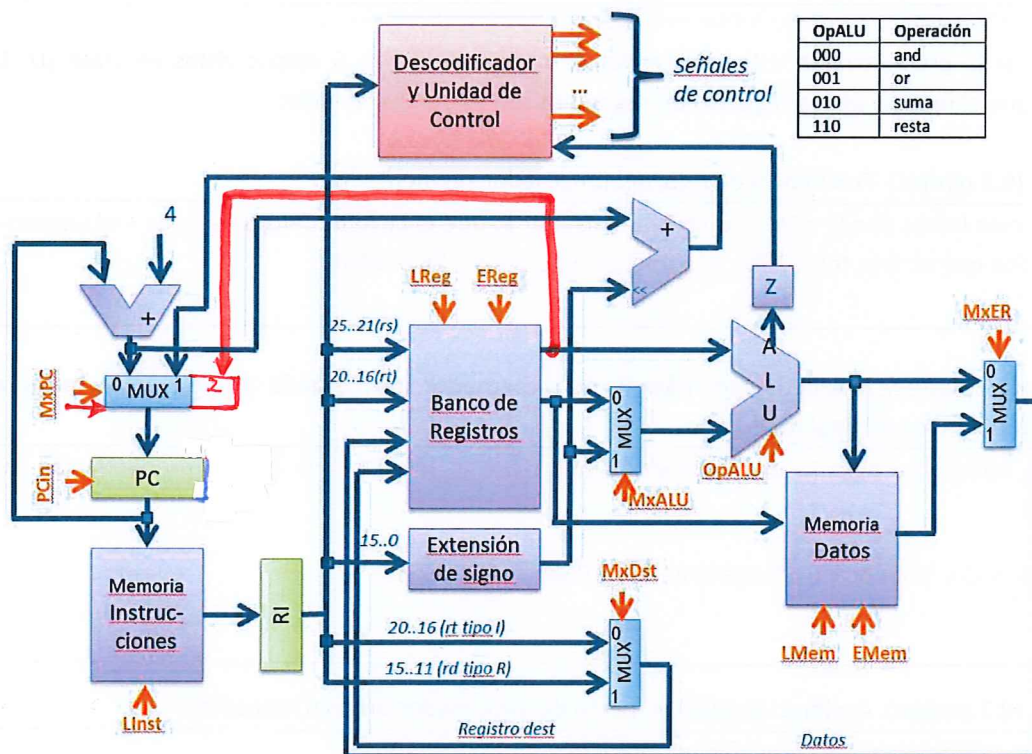
Apellidos y Nombre

DNI

Grupo

**SOLUCIÓN**

- 1 (1 punto) Dada la ruta de datos monociclo del procesador MIPS R2000, que se muestra en la figura adjunta:



- a) (0.4 puntos) Dibuje en la ruta anterior los elementos y conexiones necesarias para ejecutar la instrucción:

**jr \$rs** # PC  $\leftarrow$  \$rs, salta a la instrucción apuntada por el contenido del registro rs

COP	RS	RT	RD	Función	
000000	rs	00000	00000	00000	001000

Nota: en caso de que algún multiplexor aumente de tamaño, indique en el dibujo las nuevas señales de selección.

- b) (0.6 puntos) Rellene la siguiente tabla de señales de control, correspondiente a la unidad de control del procesador:

Instrucción	Form	EReg	OpALU	LMem	EMem	MxPC	MxALU	MxDst	MxER
sub rd, rs, rt	R	1	110	0	0	00	0	1	0
andi rt, rs, inm	I	1	000	0	0	00	1	0	0
lw rt, desp(rs)	I	1	010	1	0	00	1	0	1
sw rt, desp(rs)	I	0	010	0	1	00	1	X	X
beq rs, rt, eti	I	0	110	0	0	0Z	0	X	X
jr \$rs	R	0	XXX	0	0	10	X	X	X

**2** (1.5 puntos) En la ruta de datos anterior, asúmase que las operaciones en memoria conllevan 30 ns, leer y escribir en el banco de registros 10 ns y operar en la ALU 20 ns. Indíquese, **justificando SIEMPRE la respuesta**:

a) (0.3 puntos) La máxima frecuencia de reloj a la que puede trabajar este procesador.

La instrucción más larga es la lw que hace  $M+R+ALU+M+R = 30+10+20+30+10 = 100\text{ns}$

La frecuencia será  $1/100\text{ns} = 10\text{MHz}$

Para aumentar la productividad este procesador se segmenta en las 5 etapas vistas en clase (LI, DI, EX, M, ER). Asumiendo que el retardo de los registros de segmentación es de 5ns, se pide:

b) (0.3 puntos) Frecuencia de reloj del procesador segmentado.

Las etapas más lentas son LI y M, que tardan 30ns, por lo que el ciclo de reloj será 30ns + el tiempo del registro de segmentación que es 5ns, total 35ns, y la frecuencia  $1/35\text{ns} = 28,57\text{MHz}$

c) (0.3 puntos) Productividad máxima del procesador segmentado y Aceleración máxima respecto del procesador original monociclo

En el caso óptimo se ejecuta una instrucción por ciclo de reloj, por lo que la productividad máxima será de 28,57 MIPS.

$\text{Speedup } (n \rightarrow \infty) = T_{\text{monociclo}} / T_{\text{segmentado}} = 100\text{ns} / 35\text{ns} = 2.86$

d) (0.2 puntos) Aceleración ideal respecto del procesador original monociclo

El speedup ideal de un procesador es el número de etapas, por tanto 5.

e) (0.4 puntos) Suponga que se decide supersegmentar el procesador, dividiendo las etapas LI y M en dos subetapas de 15 ns (LI1, LI2, DI, EX, M1, M2, ER). ¿qué tiempo se requeriría para ejecutar 4000 instrucciones en dicho procesador? Incluya en el cálculo los ciclos de llenado inicial del pipeline. Suponga que no hay conflictos.

Al dividir las etapas más lentas, la nueva etapa más lenta es la EX que son 20ns, sumando el tiempo del registro de segmentación sería un tiempo de ciclo de reloj de  $20+5=25\text{ns}$

4000 instrucción en el caso óptimo serían 4006 ciclos (6 por el llenado inicial), por tanto tardaría

$T = 4006 * 25 = 100150 \text{ ns} = 100.15 \mu\text{s}$

**3** (1 punto) En el procesador segmentado del ejercicio anterior (sin supersegmentar) se va a ejecutar el siguiente fragmento de código en ensamblador del MIPS R2000.

```

(1)          lw $t2, 0($t1)
(2)          add $t3, $t2, $t3
(3)          bne $t3, $zero, no
(4)          beq $t2, $zero, si
(5)  no:      addi $t1, $t1, 4
(6)  si:      sw $t3, 0($t1)
(7)          add $t3, $t2, $t3

```

a) (0.7 puntos) Asuma que se utiliza inserción de ciclos de parada tanto para solucionar los conflictos por dependencias de datos como los riesgos de control. Teniendo en cuenta que para este procesador la latencia de salto es 2 y que el salto de (3) NO se produce y el salto de (4) SI se produce, complete el diagrama instrucciones/ciclo

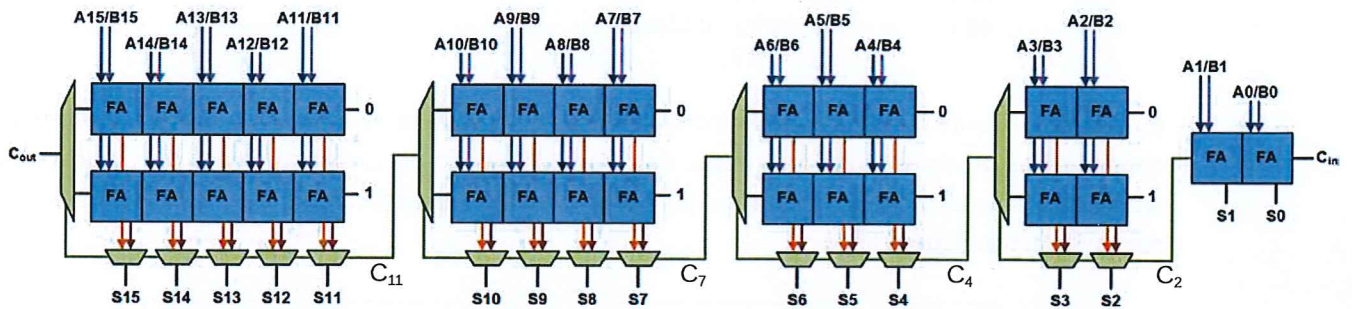
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
(1) lw	LI	DI	EX	M	ER														
(2) add		LI	DI	DI	DI	EX	M	ER											
(3) bne			LI	LI	LI	DI	DI	DI	EX	M	ER								
(4) beq						LI	LI	LI	LI	LI	DI	EX	M	ER					
(6) sw											LI	LI	LI	DI	EX	M	ER		
(7) add														LI	DI	EX	M	ER	

b) (0.3 puntos) Indique para dicho código:

Número de ciclos de parada (P)	4 (datos) + 4 (control) = 8
Número de ciclos totales de ejecución (T)	$I + (K-1) + P = 6 + 4 + 8 = 18$
CPI (indique las operaciones realizadas para el cálculo)	$(18-4)/6 = 2,33$



- 4 (1.2 puntos) Se pretende diseñar un sumador de enteros de 16 bits que reduzca el tiempo de respuesta respecto al sumador con propagación de acarreo (CPA) clásico. Se opta por un **sumador con selección de acarreo (Carry Select Adder, CSA)**, pero con el **tamaño de los bloques variable**. En concreto, se trata de un CSA con tamaño de bloques 5-4-3-2-2 (de izquierda a derecha), tal como muestra la figura adjunta. Suponga los FA (Full Adder) diseñados en base a sus funciones lógicas, como los que se han visto en el aula. Suponga que el retardo de una puerta es 1ns, y el retardo de un multiplexor es de 2ns.



- a) (0.1 puntos) Indique el retardo de un FA:

$t_s(\text{suma}) = 3 \text{ ns}$

$t_c(\text{carry}) = 2 \text{ ns}$

- b) (0.6 puntos) Indique el retardo en las señales siguientes:

$C_2 = 4 \text{ ns}$

$C_4 = 6 \text{ ns}$

$C_7 = 8 \text{ ns}$

$C_{11} = 10 \text{ ns}$

$S_{15} = 13 \text{ ns}$

$C_{out} = 12 \text{ ns}$

- c) (0.2 puntos) Indique la productividad del sumador en unidades MOPS

$X = 1/13\text{ns} = 76.92 \text{ MOPS}$

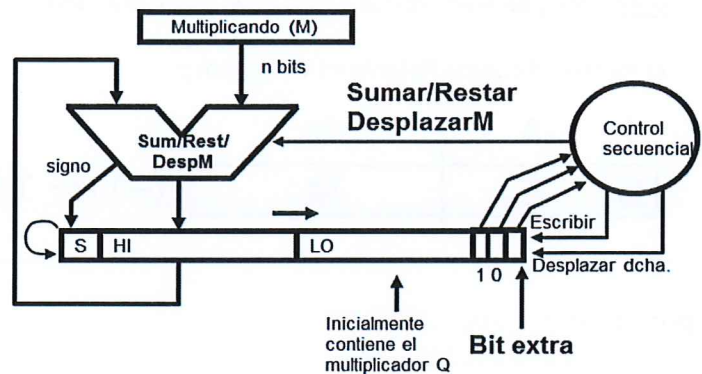
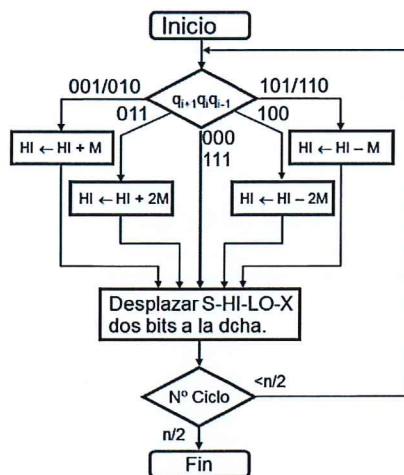
- d) (0.3 puntos) Compare con la productividad de un sumador CPA de 16 bits.

$t_{cout} = 2nt_p = 32 \text{ ns}$

$t_s = (2n+1) t_p = 33 \text{ ns}$

$t_p = \text{retardo de puerta básica, } 1 \text{ ns en este caso}$

- 5 (1.3 puntos) Considere un operador secuencial de multiplicación mediante **recodificación por parejas de bits**. Los operandos son enteros de 8 bits con signo. A continuación se muestran los detalles del algoritmo y del operador.



- a) (0.2 puntos) Considere la multiplicación del multiplicando  $M = 3$  y el multiplicador  $Q = -8$  en este operador. ¿Cuáles serán los valores de HI y LO tras la inicialización del operador?

HI = 00000000  
LO = Q = 11111000

- b) (0.2 puntos) Indique la codificación Booth de  $Q (-8)$ :

Q Booth: 0 0 0 0 -1 0 0 0

- c) (0.2 puntos) Indique la recodificación por parejas de  $Q$ :

Q Parejas: 0 0 -2 0

- d) (0.2 puntos) Indique las acciones a realizar en el ciclo 2 del algoritmo (sin tener en cuenta el ciclo de inicialización de los registros):

HI ← HI - 2M  
Desplazar S-HI-LO-X dos bits a la derecha

- e) (0.5 puntos) Calcule la productividad del operador teniendo en cuenta los retardos de las operaciones. Se asume que en cada ciclo de reloj se completa una iteración. La inicialización se realiza en un ciclo de reloj adicional.

Desplazar M 1 bit	Insp. $q_{i+1}, q_i, q_{i-1}$	suma	resta	escritura HI	Desplazamiento S-HI-LO-X 1 bit	Eval. Nº ciclo
2 ns	6 ns	20ns	25ns	2 ns	4 ns	5 ns

$$t_{\text{ciclo}} = t_{\text{insp}} + t_{\text{despM}} + t_{\text{resta}} + t_{\text{escrituraHI}} + 2 \times t_{\text{desplazartodo}} + t_{\text{evaluar}} = 6 + 2 + 25 + 2 + 8 + 5 = 48\text{ns}$$

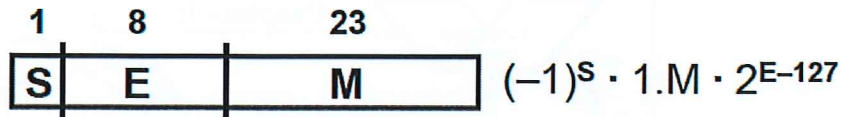
número de ciclos =  $(n/2) + 1 = 5$  (se incluye el de inicialización)

retardo multiplicación =  $48 \times 5 = 240\text{ns}$

$X = 1/240\text{ns} = 4.17 \text{ MOPS}$

- 6 (1 punto) Escriba el código de una subrutina **pot2** que calcule  $2^n$  ( $-126 \leq n \leq 127$ ). La subrutina recibe como parámetros el valor de  $n$  en el registro  $\$a0$  y devuelve el resultado en  $\$f0$ . El cálculo se realizará utilizando exclusivamente operaciones de enteros para componer el resultado en formato de coma flotante de simple precisión, y se asumirá que  $n$  está en el rango indicado.

El formato de coma flotante es el siguiente:



```
pot2: addi $t0,$a0,127
      sll $t0,$t0,23
      mtcl $t0,$f0
      jr $ra

 $2^n = (-1)^0 \cdot 1.0 \cdot 2^n \rightarrow E=n+127, S=0, M=0$ 
```

- 7 (1.4 puntos) Se está diseñando un mapa de memoria para un procesador MIPS de 32 bits de acuerdo con los siguientes requisitos:

- A partir de la dirección 0x40000000 se va a ubicar un módulo de memoria M1 de 512 MB, un módulo M2 de 256MB y un módulo M3 de 64MB, sin dejar ninguna dirección libre entre ellos. Indique para este caso:

- a. (0.6 puntos) ¿Cuál será el rango de direcciones de cada módulo? Expréselo en hexadecimal.

M1 de 0x40000000 a 0x5FFFFFFF  
 M2 de 0x60000000 a 0x6FFFFFFF  
 M3 de 0x70000000 a 0x73FFFFFF

- b. (0.6 puntos) ¿Cuál será la función de selección para cada módulo utilizando lógica negativa?

$M1^* = A_{31} + A_{30}^* + A_{29}$   
 $M2^* = A_{31} + A_{30}^* + A_{29}^* + A_{28}$   
 $M3^* = A_{31} + A_{30}^* + A_{29}^* + A_{28}^* + A_{27} + A_{26}$

- c. (0.2 puntos) ¿Cuántas líneas de direcciones y cuantas líneas de selección de octeto hay en el bus del procesador? Indique cuáles son.

Bus de direcciones:  $A_{31} \dots A_2$ , 30 líneas físicas de dirección  
 $/BE_3, /BE_2, /BE_1, /BE_0$ , 4 líneas de selección de octeto, al haber 4 octetos por palabra (32 bits)

8 (1.6 punto) Se dispone de un chip de memoria SDRAM DDR con un ancho de palabra de 64 bits. En la tabla siguiente pueden verse los valores de los principales parámetros temporales de esta memoria expresados en ns. Dicha memoria va a conectarse a una CPU mediante un bus de frecuencia 500 MHz ( $t_{\text{ciclo}}=2$  ns).

- a. (0.3 puntos) Complete la tabla de los parámetros de la memoria expresando su valor en ciclos para que puedan cumplirse las restricciones temporales.

Parámetro	ns	ciclos
$T_{\text{RCD}}$	7	4
$T_{\text{RAS}}$	15	8
$T_{\text{RC}}$	22	11
$T_{\text{RP}}$	6	3
CL		3

- b. (0.4 puntos) Rellene el cronograma para realizar la lectura de un bloque de 8 palabras. Considere que la orden de precarga NO es automática.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16
Orden	act				read				pre							
Dirección	Fila				col.											
Datos								w1w2	w3w4	w5w6	w7w8					

- c. (0.3 puntos) ¿Cuál es el ancho de banda de esta memoria? Inclúyanse los cálculos realizados.

$$B = W \times f \times 2 = 8B \times 500\text{MHz} \times 2 = 8000 \text{ MB/s} = 8\text{GB/s}$$

- d. (0.3 puntos) ¿Cuál es su tiempo de acceso? Inclúyanse los cálculos realizados.

$$T_A = T_{\text{RCD}} + T_{\text{CL}} = (4+3) t_{\text{ciclo}} = 7 t_{\text{ciclo}} = 14\text{ns}$$

- e. (0.3 puntos) ¿Cómo se verían afectados el ancho de banda y el tiempo de acceso si la memoria fuera de tipo DDR3 y se mantuvieran los mismos valores de los parámetros temporales? Inclúyanse los cálculos realizados.

TA el mismo y B el mismo, para la misma frecuencia y parámetros temporales.

No obstante, DDR3 podría llegar a funcionar hasta una frecuencia cuádruple que DDR, en ese caso B sería 4x



