

Apellidos y Nombre

DNI

Grupo

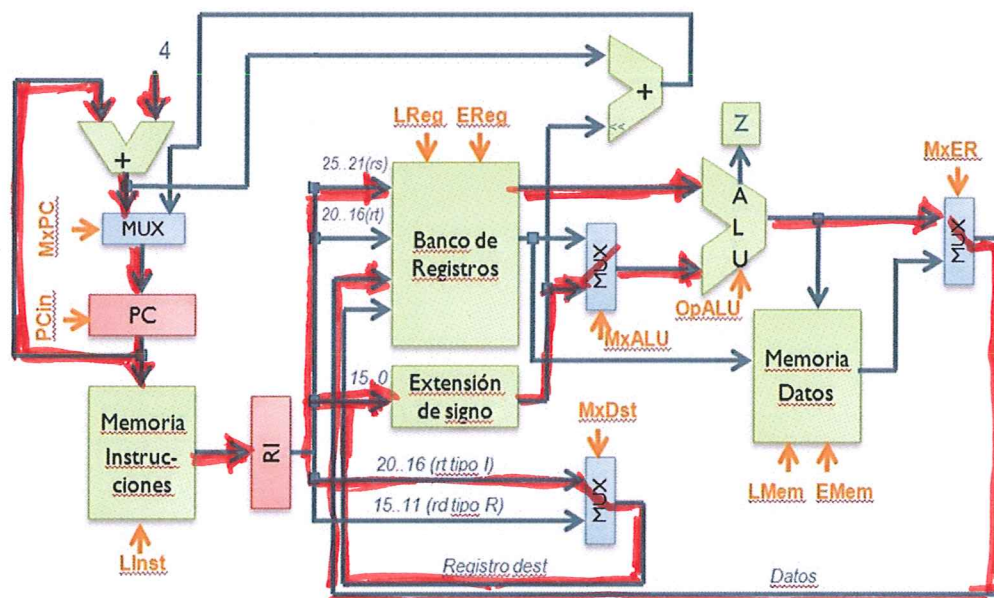
1. (3 puntos) Se desea incluir la pseudoinstrucción `li rt,inm` como una instrucción más del juego de instrucciones básico del MIPS visto en clase. La instrucción carga el valor `inm` en `rt`.



Figura 1

Teniendo en cuenta que esta instrucción va a seguir el formato I indicado en la Figura 1 y que la codificación del registro `Rs` será siempre 0:

- a) (1.5 puntos) Marque sobre la ruta de datos de la Figura 2, los caminos activos para ejecutar esta instrucción. Considérese también la etapa de búsqueda de la instrucción y las acciones correspondientes.



- b) (1.5 puntos) Complete la tabla correspondiente a las señales de control para poder ejecutar las instrucciones indicadas. Las señales de operación en la ALU tiene la siguiente codificación

OpUAl	Operación
000	$a \wedge b$ (and)
001	$a \vee b$ (or)
010	$a + b$ (suma aritmética)
110	$a - b$ (resta)

Instrucción	Form	EReg	OpALU	LMem	EMem	MxPC	MxALU	MxDst	MxER
sub rd, rs, rt	R	1	110	0	0	0	0	1	0
lw rt, desp(rs)	I	1	010	1	0	0	1	0	1
sw rt, desp(rs)	I	0	010	0	1	0	1	X	X
beq rs, rt, etiq	I	0	110	0	0	1	0	X	X
li rt, inm	I	1	001 010	0	0	0	1	0	0

2. (1 punto) La ruta de datos anterior correspondiente a un procesador monociclo con una frecuencia de reloj de 8 MHz se rediseña segmentándola en 5 etapas (LI, DI, EX, M, ER). La duración de las etapas con acceso a memoria es de 35 ns y las de decodificación, ejecución y escritura en registros 20 ns. Asuma que los registros de segmentación introducen un retardo de 5 ns. Indique:

a) (0,4 puntos) Frecuencia de reloj del procesador segmentado

$$C = C_{max} + t_{req} = 35 + 5 = 40 \text{ ns} \rightarrow f_{req} = \frac{1}{40 \text{ ns}} = 25 \text{ MHz}$$

b) (0,3 puntos) Productividad máxima que puede llegar a ofrecer

$$P_{max} = \frac{1 \text{ inst.}}{\text{ciclo}} = \frac{1}{40 \text{ ns}} = 25 \text{ MIPS}$$

c) (0,3 puntos) Aceleración máxima respecto del procesador original monociclo

$$S_{max} = \frac{T_{monociclo}}{T_{seg}} = \frac{125 \text{ ns}}{40 \text{ ns}} = 3,125$$

3. (2 puntos) Sobre el anterior procesador segmentado se ejecuta el código siguiente:

```

.data 0x10000000
A:    .word 10, 20, 30, 40    # Vector A
N:    .word 4                # Num. de elementos del vector
suma: .word 0
.text 0x00400000
1 start: lui $4, 0x1000
2        addi $5, $4, 16
3        lw $8, 0($5)
4        ori $6, $0, 0
5 do:    lw $7, 0($4)
6        add $6, $6, $7
7        addi $4, $4, 4
8        addi $8, $8, -1
9        bgtz $8, do
10       addi $5, $5, 4
11 fin:  sw $6, 0($5)

```

Handwritten annotations:

- Instructions 1-4 are grouped with a bracket and labeled "4 inst".
- Instruction 5 is labeled "1P".
- Instructions 6-9 are grouped with a bracket and labeled "4 veces - 5 inst.".
- Instructions 10-11 are grouped with a bracket and labeled "3P" and "2 INST.".

- a) (1 punto) Indique utilizando la tabla los **conflictos en los datos** que se producen así como la solución más adecuada para los mismos. Asuma que el procesador dispone de los cortocircuitos estudiados en clase (MaEX, ERaEX, ERaM) y especifique el cortocircuito elegido para solucionar cada riesgo detectado. En el caso de necesitar ciclos de parada indique el número de ellos.

	Registro	Número de instrucción en que se escribe	Número de instrucción en que se lee	Solución
Riesgo 1	\$4	1	2	MaEX
Riesgo 2	\$5	2	3	MaEX
Riesgo 3	\$7	5	6	Parada + ERaEX
Riesgo 4	\$8	8	9	MaEX
Riesgo 5	\$5	10	11	MaEX
Riesgo 6	\$6	4	6	

hay riesgo pero no genera conflicto
El ciclo de parada entre 5) y 6) lo soluciona.

b) (1 punto) Indique para dicho código:

Asumimos latencia salto = 3 ciclos

Número de Instrucciones ejecutadas (I)	$4 + 4 * 5 + 2 = 26 \text{ inst.}$
Número de ciclos de parada (P)	$4 * (1 + 3) = 16$
Número de ciclos totales de ejecución (T)	$C = n + k - 1 + P = 26 + 5 - 1 + 16 = 46$
CPI	$CPI = \frac{46 - 4}{26} = 1,6$

4. (1,5 puntos) El circuito de la figura 3 es un sumador de enteros de 16 bits que pretende reducir el tiempo de respuesta respecto al sumador con propagación de acarreo (CPA) clásico. Se trata de dos sumadores con selección de acarreo (Carry Select Adder, CSA) conectados en serie a través del acarreo de peso 8. Un CSA se encarga de la suma de los 8 bits de menor peso y otro de los 8 bits de mayor peso.

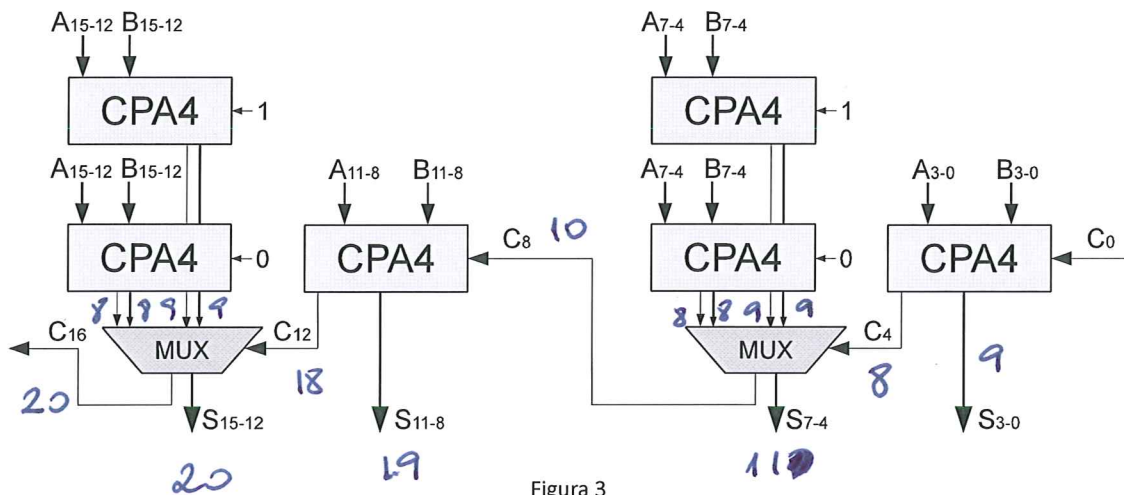


Figura 3

Los CPAs de 4 bits que emplea (CPA4) utilizan sumadores completos con un tiempo de respuesta de 2 ns para el acarreo y 3 ns para la suma. El retardo introducido por los multiplexores (MUX) es de 2 ns. Indique:

- a) (0,5 puntos)Cuál sería el tiempo de respuesta de un sumador CPA clásico de 16 bits que utilizara los mismos sumadores completos ($t_s=3 \text{ ns}$, $t_c=2 \text{ ns}$).

$$T = 33 \mu s$$

- b) (0,5 puntos) El tiempo de respuesta del sumador propuesto en los siguientes puntos:

S_{3-0} 9 ns
 S_{7-4} 11 ns
 S_{11-8} 19 ns
 S_{15-12} 20 ns
 C_{16} 20 ns

- c) (0,5 puntos) La productividad del CPA clásico y la del sumador propuesto. Expresé los resultados en MOPS.

$$P_{CPA} = \frac{1}{33 \mu s} = 30,3 \text{ MOPS}$$

$$P_{CSA} = \frac{1}{20 \mu s} = 50 \text{ MOPS}$$

5. (1 punto) Se dispone de un circuito multiplicador secuencial con **recodificación por parejas de bits** para números enteros de 32 bits tal y como se muestra en la figura 4. Supóngase que el retardo de las diferentes operaciones involucradas en la multiplicación es el que se indica en la tabla posterior:

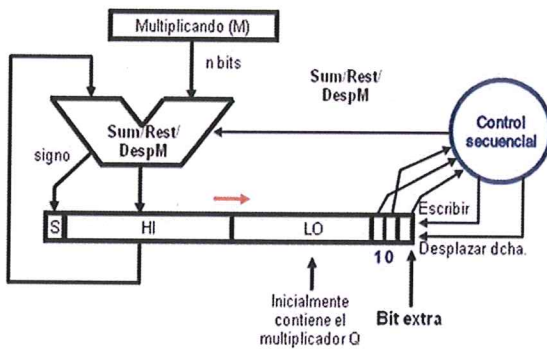


Figura 4

Inicializar registros y circuito de control	4 ns
Inspeccionar q_{i+1} q_i y q_{i-1}	1 ns
Sumar	10 ns
Restar	12 ns
Desplazar M 1 bit	2 ns
Desplazar S-HI-LO-X 2 bits	4 ns
Escribir registro HI	2 ns
Evaluar el número de ciclo actual	4 ns

- a) (0,4 puntos) Indique cuál sería el periodo mínimo de la señal de reloj y cuántos ciclos son necesarios para realizar una operación de multiplicación. Asuma que la inicialización de los registros se hace en un ciclo

$$T = 25 \mu s$$

$$n^{\circ} \text{ ciclos} = 1 + 16 = 17$$

- b) (0,2 puntos) ¿Cuál sería la productividad que podría alcanzar dicho multiplicador?

$$P = \frac{1 \text{ op}}{17 * 25} = 2,35 \text{ MOPS}$$

- c) (0,4 puntos) Para el siguiente operando de 16 bits indique cuál sería la recodificación por parejas de bits correspondiente 1 0 0 0 1 1 0 0 0 1 1 1 0 1 0 1

$$\text{Booth} = \begin{array}{cccccccccccccccc} - & 1 & 0 & 0 & + & 1 & 0 & - & 1 & 0 & 0 & + & 1 & 0 & 0 & - & 1 & + & 1 & - & 1 & + & 1 & - & 1 \end{array}$$

$$\text{Rec. parejas} = \begin{array}{cccccccccccccccc} \text{---} & & \text{---} & & \text{---} & & \text{---} & & \text{---} & & \text{---} & & \text{---} & & \text{---} & & \text{---} & & \text{---} & & \text{---} & & \text{---} & & \text{---} \end{array}$$

$$\text{Rec. parejas} = \begin{array}{cccccccccccccccc} -2 & & +1 & & -1 & & 0 & & +2 & & -1 & & +1 & & +1 & & \end{array}$$

6. (1.5 punto) Escriba el código de una subrutina **área_esfera** que calcule el área de una esfera de radio r mediante la fórmula $4\pi r^2$. La subrutina recibe el radio como un entero en ca2 en el registro \$a0 y devuelve el resultado en \$f0. Utilice números reales en simple precisión. Como registros temporales puede utilizar \$f4...\$f10.

```

área-esfera : mtc1 $a0, $f0
               cvt.s.w $f0, $f0
               li.s $f4, 4.0
               li.s $f6, 3.14
               mul.s $f0, $f0, $f0
               mul.s $f0, $f0, $f4
               mul.s $f0, $f0, $f6
               jr ra

```