

Apellidos y Nombre

DNI

Grupo

--	--	--

1. (3 puntos) La figura 1 muestra la ruta de datos básica del procesador monociclo visto en clase. Esta ruta permite ejecutar, entre otras, las instrucciones **lui rt,inm₁₆** (Load Upper Immediate) y **j etiqueta** (salto incondicional). Estas instrucciones tienen el siguiente formato:

- **lui rt,inm₁₆** Pone en la parte alta del registro 'rt' la constante inmediata de 16 bits 'inm₁₆', dejando la parte baja a ceros.



- **j etiqueta** La instrucción 'jump' produce un salto incondicional a la dirección indicada por 'etiqueta'. La dirección se crea mediante el campo 'inm₂₆' de 26 bits que incluye la instrucción, multiplicada por 4 (desplazamiento 2 bits izquierda) y los 4 bits de más peso del contador de programa [PC+4]:

$$\text{Dirección (etiqueta)} = [\text{PC}+4]_{31..28} \parallel \text{inm}_{26} \parallel 00$$



Para dar soporte a las instrucciones, la ruta de datos incluye: un operador de extensión de signo cuya funcionalidad se gestiona mediante los 2 bits (OpExtens) como muestra la Tabla 1; la ALU que se gestiona mediante las señales de control (OpALU) de 3 bits como muestra la Tabla 2. La operación IDENTIDAD significa que la salida de la ALU es igual a la entrada 2, pero en bit Z se activará (Z=1) si la salida es cero. Observe también que el multiplexor del PC (MxPC) se gestiona mediante 2 bits.

OpExtens	Función	Salida (32 bits)
00	Extiende signo (s)	sssss....sssss [Entrada]
01	Extiende 1's	1111...1111 [Entrada]
10	Extiende 0's	0000....0000 [Entrada]
11	Pone en parte alta	[Entrada] 0000000....0

Tabla 1

OpALU	Operación
000	AND
001	OR
010	SUMA
110	RESTA
111	IDENTIDAD

Tabla 2

- a) (1 punto) Marque sobre esta ruta de datos los caminos activos para ejecutar estas instrucciones. Considérese también la etapa de búsqueda de la instrucción y las acciones correspondientes.

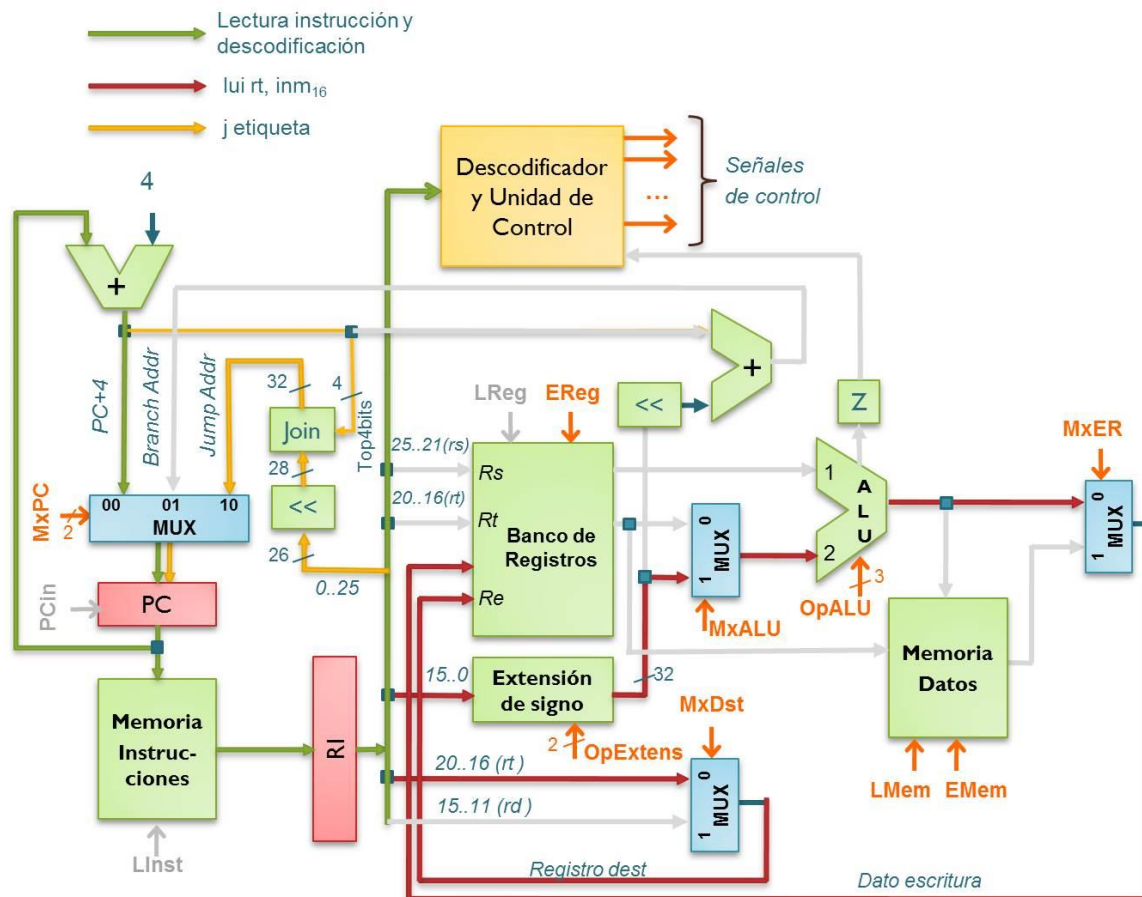


Figura 1. Ruta de datos básica.

b) (1 punto) Complete la tabla correspondiente a las señales de control para poder ejecutar las instrucciones que se indican.

Instrucción	Form	EReg	OpALU	OpExtens	LMem	EMem	MxPC	MxALU	MxDst	MxER
<code>add rd, rs, rt</code>	R	1	010	XX	0	0	00	0	1	0
<code>sw rt, desp(rs)</code>	I	0	010	00	0	1	00	1	X	X
<code>beqz rt, etiq</code>	I	0	111	00	0	0	0Z	0	X	X
<code>lui rt, inm₁₆</code>	I	1	111	11	0	0	00	1	0	0
<code>j etiqueta</code>	J	0	XXX	XX	0	0	10	X	X	X

c) (0.75 puntos) Asumiendo los retardos de los distintos elementos de la ruta de datos que se indican a continuación, indique los tiempos de ejecución de cada una de las instrucciones siguientes:

Lectura Memoria Instrucciones y carga RI	30 ns
Escritura de PC	5 ns
Decodificación de la instrucción	25 ns
Lectura / Escritura del banco de registros	25 ns
Sumador simple (+); desplazador (<<) y 'Join'	5 ns
Operación extensión signo	10 ns
Operación en la ALU	25 ns
Lectura / Escritura Memoria de datos	30 ns
Multiplexores	0 ns

Recuerde que las señales de control no son válidas hasta después de la decodificación de la instrucción.

Instrucción	Tiempo de ejecución (ns)
add rd, rs, rt	105
lw rt, desp(rs)	145
lui rt, inm ₁₆	115

ver esquema de cálculo de tiempos en la última página

- d) (0.25 puntos) Asumiendo que la instrucción más lenta que se puede ejecutar en la ruta de datos básica es alguna de las indicadas en el apartado anterior, calcule la productividad de este procesador monociclo (en MIPS)

Instrucción más lenta (lw) = 145 ns
Productividad = $1 / 145 \times 10^{-9} = 6.897$ MIPS

2. (1 punto) El procesador anterior se rediseña segmentándolo en 5 etapas (LI, DI, EX, M, ER). La duración de las etapas es la siguiente LI: 35ns; DI: 28ns; EX: 30ns; M: 35ns y ER: 30ns. Los registros de segmentación introducen un retardo adicional de 5 ns. Indique, justificando en todos los casos el resultado indicado:

- a) (0,25 puntos) Frecuencia de reloj del procesador segmentado

$T = \max \{ \tau_i \} + T_{\text{reg_segm}} = 35 + 5 = 40$ ns
 $F = 1/\tau = 1 / 40 \times 10^{-9} = 25$ MHz

- b) (0,25 puntos) Productividad máxima que puede llegar a ofrecer

$X(\infty) = 1/\tau = 25$ MIPS

- c) (0,25 puntos) Aceleración máxima respecto del procesador original monociclo del ejercicio anterior

$S = T / \tau = 145 / 40 = 3.625$

- d) (0,25 puntos) ¿Qué debería ocurrir para que el procesador segmentado tuviera una aceleración máxima de 5?

Para que $S = 5$ todas las etapas deberían tener el mismo retardo y los registros de segmentación un retardo nulo.

3. (1.5 puntos) Sobre un procesador segmentado de 5 etapas se ejecuta el siguiente código:

```
(1)      ori  $8, $0, 20
(2)  do:  lui  $7, 0x0005
(3)      ori  $7, $7, 0xFFFF
(4)      lw   $4, 0($8)
(5)      add  $6, $6, $4
(6)      sw   $7, 0($8)
(7)      addi $8, $8, -4
(8)      bgtz $8, do
```

Este procesador tiene latencia de salto 2 y los conflictos de control se resuelven mediante la inserción de ciclos de espera. Para la resolución de los conflictos debidos a dependencias de datos, el procesador puede hacer uso de técnicas de anticipación implementando los cortocircuitos vistos en clase (MaEX, ERaEX, ERaM). En caso de no poder resolver enteramente el conflicto mediante estas técnicas el compilador insertará instrucciones NOP.

- a) (0.5 puntos) Indique utilizando la tabla los **conflictos en los datos** que se producen así como la solución más eficiente para los mismos. (En caso de insertar alguna NOP mantenga la numeración original de las instrucciones tal y como indica el enunciado del ejercicio 2).

	Registro	Número de instrucción en que se escribe	Número de instrucción en que se lee	Solución
Riesgo 1	\$7	2	3	MaEX
Riesgo 2	\$4	4	5	NOP + EraEX
Riesgo 3	\$8	7	8	MaEX

...				

- b) (1 punto) Indique para dicho código y especifique cómo obtiene el resultado:

Número total de Instrucciones ejecutadas (I)	1 antes del bucle + (7 + 1 nop)*5 iteraciones = 41 instr
Número de ciclos totales de ejecución (T)	ciclos=4 iniciales + 1 instr antes del bucle + (7 intrs +1 nop + 2 stalls)*5 iteraciones = 55 ciclos
CPI	(ciclos-4)/instrucciones=51/41=1,24 CPI

4. (1, 5 puntos) Se dispone de un circuito multiplicador secuencial para números enteros de 32 bits que admite tanto el algoritmo de Booth como el de recodificación por parejas de bits tal y como se muestra en la figura 2. Supóngase que el retardo de las diferentes operaciones involucradas en la multiplicación es el que se indica en la tabla adjunta.

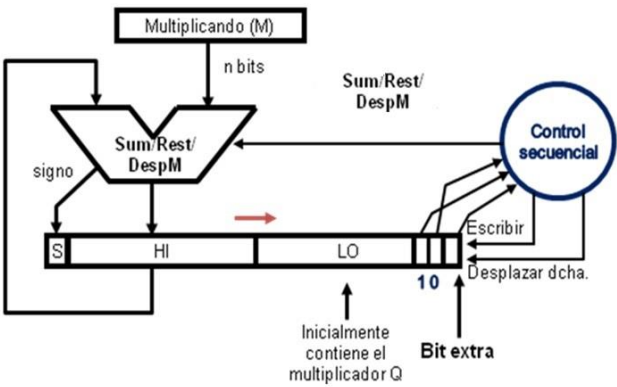


Figura 2

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

- a) (0,75 puntos) Indique cuál sería el periodo mínimo de la señal de reloj y cuántos ciclos son necesarios para realizar una operación de multiplicación con cada uno de los algoritmos propuestos. Asuma que la inicialización de los registros se hace en un ciclo. Justifique las respuestas.

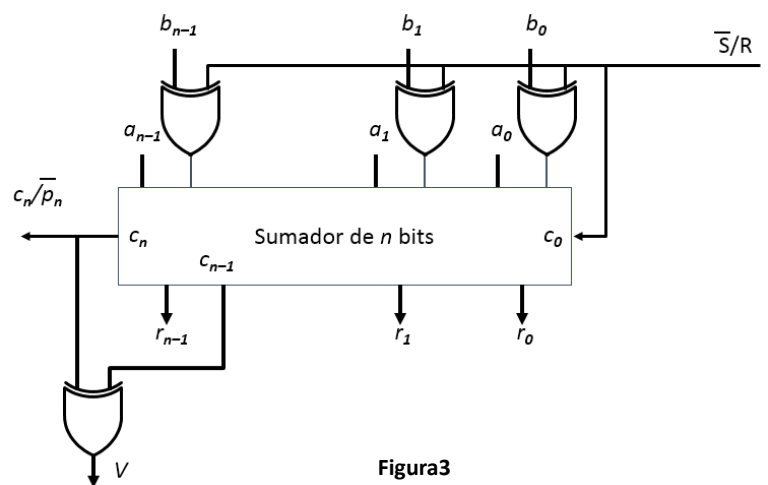
Multiplicación por Booth	$\tau = 1$ (inspecc. qi) + 12 (restar M) + 2 (escribir HI) + 2 (desplazar S_HI-LO-X 1 bit) + 4 (evaluar ciclo) = 21 ns	Nº de ciclos = 1(inicialización) + 32 = 33 ciclos
Multiplicación por recodificación por parejas	$\tau = 1$ (inspecc. qi) + 2x1 (2M) + 12 (restar 2M) + 2 (escribir HI) + 2x2 (desplazar S_HI-LO-X 2 bits) + 4 (evaluar ciclo) = 25 ns	Nº de ciclos = 1(inicialización) + 16 = 17 ciclos

- b) (0,75 puntos) ¿Cuál sería la productividad que podría alcanzar dicho multiplicador? Justifíquelo

Multiplicación por Booth	Productividad = $1 \text{ op} / (33 \times 21) \text{ ns} = 1.443 \text{ MOPS}$
Multiplicación por recodificación por parejas	Productividad = $1 \text{ op} / (17 \times 25) \text{ ns} = 2.353 \text{ MOPS}$

5. (1 punto) El esquema de la figura 3 corresponde con un sumador/restador de n-bits

Teniendo en cuenta que las puertas XOR tiene un retardo equivalente a 3 niveles de puertas, y que el sumador de n bits es un sumador serie (formado exclusivamente por sumadores completos de 1 bit conectados en cascada).



- a) (0,75 puntos) Asumiendo que el tiempo equivalente a 1 nivel de puertas es 1ns, calcula la expresión del tiempo total de circuito en función del número de bits (n).

En un sumador normal, el tiempo de C_n es $2n$, y el de S_{n-1} es $2n+1$, en esta caso al tener una puerta XOR antes de las entradas b, el tiempo se incrementa en 3, por lo que C_n es $2n+3$ y r_{n-1} es $2n+4$, en este caso la señal más lenta de todas sería la de V que viene dada por el tiempo de C_n+3 y por tanto será $2n+6$ ns.

- b) (0,25 puntos) ¿Cuál sería la productividad del circuito para $n=8$?

Para $n=8$, el tiempo será $(2 \times 8 + 6) = 22 \text{ ns}$ y la productividad será $1/22 \text{ ns} = 45,45 \text{ Mops}$

6. (2 puntos) Se desea escribir en ensamblador del MIPS el código de una subrutina **double volumen(float *radio)** que calcule el volumen de una esfera de radio 'r' de acuerdo con la expresión estándar:

$$volumen = \frac{4.0}{3.0} \cdot \pi \cdot r^3$$

- a) (1.5 puntos) Escriba el código de dicha subrutina asumiendo que el registro \$a0 se emplea para pasar la dirección de la variable float en memoria que contiene el radio y el resultado se devolverá en \$f0. Los cálculos deben hacerse en doble precisión. Puede utilizar pseudoinstrucciones **li.d** para inicializar con valores reales en doble precisión.

```
volumen:    lwcl $f0, 0($a0)      # Leo el radio 'float'
            cvt.d.s $f0, $f0     # convertir a 'double'
            mul.d $f2, $f0, $f0  # r^2
            mul.d $f0, $f2, $f0  # r ^3
            li.d $f2, 4.0
            li.d $f4, 3.0
            li.d $f6, 3.1416
            mul.d $f2, $f2, $f6  # 4.0 x PI
            mul.d $f0, $f2, $f0  # 4.0 x PI x r^3
            div.d $f0, $f0, $f4  # 4.0 x PI x r^3 / 3.0
            jr $ra
```

- b) (0.5 puntos) Suponga que el procesador MIPS tarda 1.2 segundos en ejecutar un millón de veces seguidas la subrutina **volumen** del apartado anterior. Indique la potencia de cálculo de dicho procesador en MFLOPS.

Nº instrucciones de coma flotante= 6

Productividad CPU = $6 \times 10^6 / 1.2 = 5 \times 10^6 = 5 \text{ MFLOPS}$

Esquema de cálculo de tiempos del apartado 2c:

