



**No se considerarán como válidas las respuestas en las que no se justifiquen los cálculos realizados**

**No se permite el uso de ningún tipo de documentación, pero si de calculadora**

**Sólo existe una única opción correcta por pregunta de test**

**Puntuación:**

Respuesta correcta  $\Rightarrow$  **0,25**      Respuesta incorrecta  $\Rightarrow$  **-0,15**      Pregunta no contestada  $\Rightarrow$  **0**

**TEORÍA (5 puntos)**

1.- Indica cuál de los siguientes sumadores no es un sumador que acelere la suma entera.

- a) Sumador con salto de acarreo
- b) Sumador con selección de acarreo
- c) Sumador con propagación de acarreo**
- d) Sumador con anticipación de acarreo

2.- Al sumar dos números representados en exceso M

- a) Se debe corregir siempre restando el exceso M al resultado**
- b) Se debe corregir **siempre** sumando el exceso M al resultado
- c) Solamente se debe corregir si los dos números son negativos o si siendo de diferente signo el resultado en valor absoluto de la suma es positivo
- d) La corrección consiste en invertir el bit más significativo del resultado

3.- Empleando el algoritmo de división sin restauración dividir los dos números siguientes representados en complemento a 1:    A = 111011 y el número B = 010011

**No se puede realizar puesto que el algoritmo opera solamente con números en binario puro**

4.- Sea una ALU que emplea dos bits de guarda y un bit retenedor. Sea el número 0100 0011 **100** (los tres bits resaltados en negrita corresponde con los bits de guarda y el bit retenedor) Calcular cual sería el valor correcto de redondear al más próximo siguiendo el estándar IEEE 754

- a) 0100 0011
- b) 0100 0010
- c) 0100 0100**
- d) Ninguna de las anteriores



5.- Señale la afirmación **incorrecta**:

- a) En las máquinas de acumulador las instrucciones son cortas
- b) En las máquinas de acumulador se minimizan los estados internos de la máquina.
- c) En las máquinas de acumulador existe una gran cantidad de tráfico entre la memoria y el acumulador
- d) En una máquina de acumulador el código generado es muy eficiente**

6.- Señale la afirmación **incorrecta** con respecto a de qué depende el número de registros de las máquinas de registros de propósito general:

- a) A los que requiere el compilador para ubicar variables
- b) A los que requiere el compilador para evaluar expresiones
- c) A los que requiere el compilador para pasar parámetros a los procedimientos
- d) A los que requiere el compilador para direcciones de las instrucciones**

7.- Señale la afirmación **incorrecta** con respecto a las tecnología VLIW

- a) El compilador agrupa varias instrucciones sencillas e independientes en la misma palabra de instrucción
- b) Cada compilador trabaja con una arquitectura
- c) Detecta y ordena el código en tiempo de ejecución**
- d) Se mejora la velocidad de reloj al tener unidades funcionales mejores

8.- Sea un computador de 8 bits. Para las referencias a memoria 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9. Indique cuáles de ellas se encuentran alineadas a byte.

- a) Todas ellas**
- b) Las pares
- c) Las impares
- d) Otras: \_\_\_\_\_

9.- Señale la opción **incorrecta**.

- a) Las optimizaciones de alto nivel de los compiladores dependen del lenguaje y son totalmente independientes de la máquina**
- b) Las optimizaciones de alto nivel de los compiladores son algo dependientes del lenguaje y son en su mayor parte independientes de la máquina
- c) El optimizador global presenta poca dependencia del lenguaje y ligera dependencia de la máquina
- d) El optimizador global incluye optimizaciones locales, globales y asignaciones de registros



10.- En memoria caché, las políticas de extracción se refieren a

- a) Correspondencia entre bloques de MP y MCa
- b) Qué y cuándo se envía información de MP a MCa**
- c) Qué bloque abandona MCa para dejar espacio
- d) Ninguna de las anteriores

11.- Señale la opción **incorrecta** con respecto a la memoria caché en multiprocesadores.

- a) Caches locales: cada caché contiene únicamente información de solo lectura y datos de escritura locales al procesador asociado. Las referencias a datos compartidos directamente sobre MP
- b) Directorio compartido: se emplea una zona de memoria compartida para almacenar de forma centralizada el estado de cada uno de los bloques de información presentes en las cachés
- c) Protocolo de escucha: dotar a las cachés de la capacidad para permanecer a la escucha de las peticiones de acceso a memoria que aparecen en el bus
- d) La caché privada aumenta el problema de la latencia y el tráfico de información entre la memoria principal y los procesadores**

12.- Señale la opción **incorrecta**.

- a) En un sistema de memoria caché con política de actualización de escritura inmediata se escribe a la vez en memoria principal y en memoria caché
- b) En un sistema de memoria caché con política de actualización de escritura inmediata se asegura la consistencia entre la memoria caché y la memoria principal
- c) La implementación hardware de un sistema de memoria caché con política de actualización de escritura inmediata no es sencilla de realizar**
- d) En un sistema de memoria caché con política de actualización de escritura inmediata el procesador debe esperar a que se complete la escritura.

13.- Señale la opción **incorrecta** con respecto a la memoria entrelazada

- a) Aumenta el ancho de banda de la memoria
- b) Permitir el acceso concurrente a módulos de memoria de tecnología no demasiado rápida
- c) El entrelazado de orden superior facilita la expansión de la memoria
- d) El entrelazado de orden superior reduce los conflictos de acceso si las referencias a memoria son consecutivas**

14.-¿Cuál de los siguientes elementos presenta localidad espacial?

- a) Los datos generados en las operaciones siguientes
- b) La estructura secuencial de un programa**
- c) Las mismas instrucciones dentro de un bucle
- d) Ninguna de las anteriores



15.- Las operaciones elementales

a) Pueden ser de transferencia o de proceso

b) Pueden terminar en un elemento de memoria o almacenamiento o en los buses

c) a) y b) son correctas

d) Ninguna de las anteriores

16.- Señale la opción **correcta** con respecto al dispositivo del periférico:

a) Puede ser mecánico, electromecánico, electromagnético con su electrónica de control o transductor en el caso de sensores

b) Se encarga de la comunicación con la CPU

c) Realiza la transferencia de datos

d) Define el protocolo de la comunicación

17.- Señale la opción **incorrecta** con respecto a la entrada-salida programada

a) Puede ser condicional o incondicional

b) La CPU realiza un sondeo para ver la disponibilidad del periférico

c) **Solamente se puede conectar un periférico**

d) La implementación hardware es sencilla

18.- Señale la opción **incorrecta** con respecto a la entrada-salida mediante interrupciones con una única línea de interrupción:

a) Todos los periféricos hacen la petición por la misma línea.

b) El vector de interrupción es común para todos ellos

c) La rutina de interrupción identifica mediante polling el periférico que interrumpió a la CPU y desactiva la petición

d) **La resolución de prioridades se puede realizar por daisy-chain**

19.- Señale la opción **incorrecta** con respecto a un gestor programable de interrupciones:

a) Un **PIC** permite la conexión de distintos periféricos a la línea de petición de la CPU expandiendo en varios niveles dicha línea

b) **Resuelve la gestión de prioridades de forma rígida**

c) Admite controladores esclavos para expandir más niveles de interrupción

d) Permite enmascarar aquellas líneas que se desee mediante un registro de máscara programable

20.- Señale la opción **incorrecta** con respecto a lo que requiere una rutina de tratamiento de interrupción para ser reentrante

a) No debe almacenar nada en sí misma

b) No debe automodificarse

c) Debe hacer los accesos de forma relativa

c) **Cada vez que se llame se le asignará el mismo área de trabajo**



## PROBLEMAS (5 puntos)

### EJERCICIO 1:

(3 puntos)

Sea un computador capaz de ejecutar 100 MIPS. Se desea conectar al computador, únicamente un periférico con una velocidad de transferencia de 150.000 bytes/sg. y sobre el que se realizan operaciones de lectura de bloques de 512 bytes.

Se pretende ver el comportamiento de la pareja computador-periférico ante las diferentes técnicas de entrada-salida (programada, mediante interrupciones y por DMA)

Se sabe que:

- La rutina de transferencia de E/S programada consta de 8 instrucciones.
- La rutina de tratamiento de interrupción en la E/S mediante interrupciones consta de 25 instrucciones.

La rutina de inicialización del DMA consta de 12 instrucciones. Y en cada operación de escritura de un dato en memoria el controlador ocupa los buses durante 30 ns.

Indicar el número de instrucciones de otros procesos que puede realizar el computador durante cada uno de los tipos de E/S previstos.

### EJERCICIO 2:

(2 puntos)

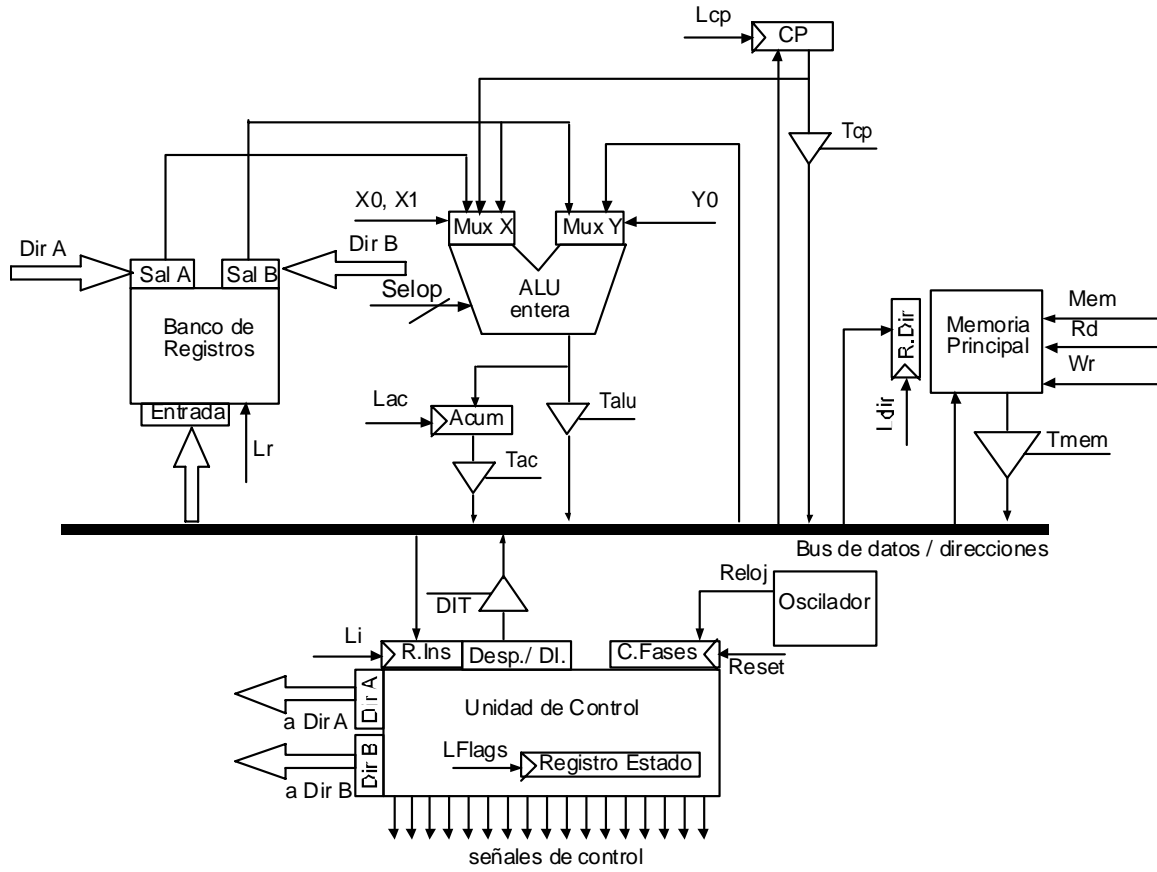
Se dispone de un computador que consta, entre otros, de los siguientes elementos:

- **ALU** que se alimenta de dos entradas y permite realizar 16 operaciones.  
Contiene un registro acumulador para almacenar resultados temporales.
- **Banco de registros** de 32 registros, con una puerta de entrada y dos puertas de salida.
- **Contador de programa** conectado al registro de datos/direcciones.
- **Memoria principal** de 128 Mbytes, organizada en palabras de 32 bits.  
Se considerará que las lecturas y escrituras en memoria se realizan en dos periodos de reloj.
- El **bus de datos / direcciones** es de 32 bits.

Se quiere que la CPU ejecute la siguiente instrucción de una palabra:

**XOR C, 80h**

- a. Describir las operaciones elementales, y las señales de control asociadas, que se realizan en cada una de las fases de ejecución de la instrucción. **(1 punto)**
- b. Realizar el formato de microinstrucción para el computador dado. **(1 punto)**





## SOLUCIÓN EJERCICIO 1 (3 PUNTOS)

El periférico suministra 150.000 bytes/sg con lo que enviará un byte cada 6,66µsg.

Como los bloques son de 512 bytes, una operación de E/S durará:

$$6,66\mu\text{sg}/\text{byte} \cdot x 512 \text{ bytes} = 3.409,92\mu\text{sg}.$$

### E/S programada

El periférico proporcionará un dato cada 6,66µsg. La CPU en ese tiempo es capaz de ejecutar 6.660 instrucciones (ya que el computador trabaja a 100 MIPS). Dado que en transmitir los 512 bytes se emplean 3.409,92µsg las instrucciones que podría haber realizado la CPU serían 340.992. Al ser entrada-salida programada, la CPU está plenamente dedicada a ejecutar las instrucciones de la rutina de entrada-salida, con lo que no es capaz de ejecutar ninguna instrucción de otro proceso.

### E/S mediante interrupciones

La rutina de interrupción consta de 25 instrucciones, y se ejecutará cada vez que está lista para entregarnos un byte nuevo. Al ser los bloques de 512 bytes, ejecutaremos un total de  $25 \times 512 = 12.800$  instrucciones dedicadas a la transferencia de datos, con lo que la CPU podría dedicar tiempo a otros procesos.

El número de instrucciones de otros procesos que puede ejecutar la CPU es de:

$$340.992 \text{ instrucciones} - 12.800 \text{ instrucciones} = 328.192$$

### E/S mediante DMA

La CPU dedicará el tiempo necesario para poder programar el controlador de DMA. El DMA emplea robo de ciclo, con lo que el periférico ocupará los buses 30 ns. por byte. Con lo que para leer 512 bytes, tardará:  $30 \text{ ns} \cdot x 512 \text{ bytes} = 15.360 \text{ ns} = 15,36\mu\text{sg}$ . Dado que la rutina de inicialización del controlador del DMA consta de 8 instrucciones, podemos aproximar el cálculo a los 15,36µsg. que se corresponden con los robos de ciclo.

En ese tiempo, la CPU habrá ejecutado 1.536 instrucciones, que se corresponderán con el hecho de que durante el robo de ciclo por cada dato, la CPU no puede acceder a los buses.

La CPU podrá ejecutar  $340.992 - 1.536 = 339.456$  instrucciones correspondientes a otros procesos.

Resumiendo:

Tipo de E/S	Instrucciones totales	Instrucciones para otros procesos
E/S programada	340.992	0
E/S mediante interrupciones	12.800	328.192
E/S mediante DMA	1.536	339.456



## **SOLUCIÓN EJERCICIO 1 (3 PUNTOS)**





## SOLUCIÓN EJERCICIO 2 (2 PUNTOS)

### Apartado a)

ACCIÓN	OPERACIÓN ELEMENTAL	SEÑAL DE CONTROL
<b>Fase de fetch</b>		
Direccionar	CP → bus de datos	(TCP)
	Bus de datos → Reg. Direcciones	(LDIR)
Traer la instrucción	Inicio ciclo de memoria	(MEM)
	Leer	(RD) dos períodos
	Contenido memoria → B. Datos	(TMEM)
	Cargar registro de instrucción	(LI)
<b>Decodificación</b>		1 período
<b>Actualización de CP</b>		
Sumar CP + 1	Tamaño → Bus de datos	(DIT)
	Bus de datos → Multiplexor X	< X0, X1 = b. datos >
	CP → Multiplexor Y	< Y0 = CP >
	Sumar	< Selop = + >
	Cargar el acumulador	(LAC)
Cargar nuevo valor en CP	Acumulador → Bus de datos	(TAC)
	Cargar el CP	(LCP)
<b>Ejecución de la instrucción</b>		
Realizar el desplazamiento	Poner C en salida A	< DIR A = C >
	Salida A → Multiplexor X	< X0, X1 = SAL A >
	80h → B. Datos	(DIT)
	B. datos → Multiplexor Y	< Y0 = B. Datos >
	Realizar la Xor	< Selop = XOR >
	Cargar el acumulador	(LAC)
	Modificar los flags de estado	(LFLAGS)
Cargar nuevo valor en F	Acumulador → Bus de datos	(TAC)
	Elegir C como registro destino	< DIR A = C >
	Cargar el banco de registros	(LR)
<b>Poner el contador de fases a 0</b>		
	Activar el Reset	(RESET)



## SOLUCIÓN EJERCICIO 2 (2 PUNTOS)

### Apartado b)

Señales de carga de registros y Reset	
B0	Reset
B1	Lflags
B2	LI
B3	LR
B4	Lac
B5	Ldir

Codificación de las señales al bus de datos / direcciones

Acceso al bus de datos / direcciones			
B8	B7	B6	
0	0	0	Tac
0	0	1	Tmem
0	1	0	Tcp
0	1	1	Dit
1	0	0	Talu
1	0	1	nop
1	1	X	

Codificación de la ALU: multiplexores Y y X y Selop

Y0	
B9	
0	Salida B
1	Bus datos / direcciones

X1	X0	
B11	B10	
0	0	Salida A
0	1	Salida B
1	0	CP
1	1	nop



Señales de selección de operación				
B15	B14	B13	B12	
0	0	0	0	ADD
0	0	0	1	SUB
0	0	1	0	MUL
0	0	1	1	DIV
0	1	0	0	ADC
0	1	0	1	SBB
0	1	1	0	SHL
...	...	...	...	
1	1	1	1	ROR

#### Señales de acceso a la memoria

Mem	R/W	
B17	B16	
0	X	nop
1	0	E
1	1	L

#### Señales de la unidad de direccionamiento

Lcp	
B18	
0	nop
1	Cargar valor en CP

#### Banco de registros

Banco de registros (salida A)					
B23	B22	B21	B20	B19	
0	0	0	0	0	A
0	0	0	0	1	B
0	0	0	1	0	C
0	0	0	1	1	D
0	0	1	0	0	E
0	0	1	0	1	F
...	...	...	...	...	...
1	1	1	1	1	...



Banco de registros (salida B)					
B28	B27	B26	B25	B24	
0	0	0	0	0	A
0	0	0	0	1	B
0	0	0	1	0	C
0	0	0	1	1	D
0	0	1	0	0	E
0	0	1	0	1	F
...	...	...	...	...	...
1	1	1	1	1	...

Cumple la condición de microsalto

Cond	
B29	
0	No cumple condición
1	Si cumple condición

La instrucción lleva microsalto

$\mu$ Salto	
B30	
0	No lleva microsalto
1	Si lleva microsalot

Bit de secuenciamiento: final de microprograma

Fin $\mu$ prg	
B31	
0	No es la última
1	Última microinstrucción

La dirección de microsalto, suponiendo una memoria de control de 64k requeriría un total de 16 bits para especificarla. Tomamos entonces los bits b15-b0 solapando los campos de la ALU, del bus de datos / direcciones y las señales de carga de registros y reset.



Apellidos, Nombre: \_\_\_\_\_

**Formato de microinstrucción sin  $\mu$ salto**

Fin $\mu$ prog	$\mu$ Salto	Cond	Rango de registros salidas A y B										CP	Memoria	Alu: multiplexores X, Y y selección de operación						Bus datos/direc						Carga de Registros y Reset					
b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	

**Formato de microinstrucción con  $\mu$ salto**

Fin $\mu$ prog	$\mu$ Salto	Cond	Dirección de $\mu$ salto condicional o incondicional según valor de b29 o b30																												
b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0