

Apellidos ..... Nombre .....

Valor de cada: Respuesta correcta + 0.1  
Respuesta incorrecta - 0.05  
No respuesta - 0.05

TIEMPO REALIZACIÓN EXAMEN: 45 minutos

TEST →	
P1 →	
P2 →	
P3 →	
NOTA →	

1	Para mejorar el rendimiento de un computador para una aplicación se tienen dos opciones: 1.-Cambiar el procesador gráfico que se utiliza un 20% del tiempo, consiguiendo un factor de mejora de 2.- Incrementar la memoria obteniendo un factor de mejora de 1,5 el 80% del tiempo. ¿Cuál es la mejor opción?:	A
	A) La opción 2 B) Ambas opciones dan igual aceleración C) La opción 1 D) Ninguna de las afirmaciones anteriores es correcta	
2	Si una máquina A ejecuta un programa en diez segundos, ¿en cuánto tiempo ejecuta el mismo programa la máquina B si la máquina A es el 50% más rápida que la B?:	A
	A) En 15 seg B) En 10,5 seg C) En 20 seg D) Se necesitan más datos para conocer la respuesta	
3	El rendimiento de la CPU depende:	A
	A) Del recuento de instrucciones B) De la tensión de alimentación de la CPU C) Del tamaño de memoria del disco duro D) Ninguna de las afirmaciones anteriores es correcta	
4	El Principio de Localidad de Referencia, se aplica:	D
	A) Sólo a los datos tipo vector B) Únicamente a los accesos de datos C) Únicamente a los accesos de instrucciones D) Ninguna de las afirmaciones anteriores es correcta	
5	Los MIPS (Millones de Instrucciones Por Segundo):	A
	A) Pueden variar entre programas en el mismo computador B) Son independientes del repertorio de instrucciones de la máquina C) No varía entre programas en el mismo computador D) Nos permiten calcular exactamente el tiempo de ejecución de un programa real	
6	En relación a la CPU, M. Principal y M. Caché, indique la afirmación que NO es cierta:	B
	A) La transferencia de información entre M. Caché y MP se hace a nivel de bloque B) La transferencia de información entre M. Caché y MP se hace a nivel de palabra C) M. Principal y M. Caché tienen diferente velocidad D) La transferencia de información entre CPU y M. Caché se hace a nivel de palabra	
7	Para un determinado tamaño de Memoria Caché, a menor tamaño de bloque corresponde:	B
	A) Mayor tasa de acierto por localidad espacial B) Mayor tasa de acierto por localidad temporal C) Siempre menor tasa de aciertos D) Ninguna de las afirmaciones anteriores es correcta	
8	De las funciones de correspondencia de la M. Caché, la que requiere de un único bloque comparador para localizar el bloque de memoria principal es la:	B
	A) Asociativa por conjuntos B) Directa C) Totalmente asociativa D) En todas se necesita más de un comparador	
9	Atendiendo a los tipos de fallos de caché respecto a las funciones de correspondencia:	B
	A) Los de capacidad no existen en la totalmente asociativa B) Los forzosos y de capacidad existen en las tres funciones de correspondencia C) Los forzosos no existen en la directa D) Los de conflicto sólo existen en la asociativa por conjuntos	
10	La frecuencia o tasa de fallos en memoria se define como:	D
	A) La fracción de accesos a memoria encontrados en el nivel inferior de la jerarquía de memoria B) La fracción de accesos a memoria no encontrados en el nivel inferior de la jerarquía de memoria C) La fracción de accesos a memoria no encontrados en cualquier nivel de la jerarquía de memoria D) La fracción de accesos a memoria no encontrados en el nivel superior de la jerarquía de memoria	
11	En el computador serie Von Neumann estudiado, indica cuál de las siguientes afirmaciones es cierta:	D
	A) No pueden realizarse operaciones elementales a la vez B) Cada instrucción tiene una duración de 1 ciclo C) Pueden estar ejecutándose varias instrucciones a la vez D) Cada microinstrucción tiene una duración de 1 ciclo	
12	Sea la arquitectura Von Neumann estudiada, para realizar una operación de acceso a memoria:	D
	A) Se define el valor del registro RM siempre B) Se necesita definir el valor del registro PC C) Se necesita definir el valor del registro SP D) Ninguna de las afirmaciones anteriores es correcta	
13	Sea un computador con unidad de control microprogramada con secuenciamiento explícito y 512 instrucciones:	C
	A) La etapa traductora ROM es de 9 pos x 512 bits B) Faltan datos para definir la etapa traductora ROM C) No necesita etapa traductora D) Ninguna de las afirmaciones anteriores es correcta	
14	De las siguientes instrucciones, indica en cuál/es de ella/s se recupera/n el contenido del registro SR de memoria:	C
	A) DIV .1, .2 con trap por división por cero B) RET y RETI C) Sólo RETI D) Ninguna de las afirmaciones anteriores es correcta	
15	Sea un sistema con arquitectura Von Neumann, indica las instrucciones que implican bifurcación condicional:	C
	A) CALL y RET B) ADD .2, .4 con trap por desbordamiento C) BZ D) RET y RETI	

<b>Comparando la arquitectura Von Neumann con la arquitectura DLX:</b>				<b>D</b>
16	A) Ambas son arquitecturas segmentadas		C) La segunda es arquitectura serie	
	B) La primera aumenta el paralelismo a nivel de instrucción		D) La primera es arquitectura serie	
<b>En base a la arquitectura DLX estudiada, qué registro emplearías para representar el valor <math>+\infty</math>?:</b>				<b>D</b>
17	A) El registro R0		C) No es posible representar ese valor	
	B) Dos registros de coma fija ( $R_i-R_{i+1}$ )		D) Un registro de coma flotante de simple precisión ( $F_i$ )	
<b>Sea la secuencia de instrucciones: ADD R1, R2, R3 y SUB R2, R2, R4 que se ejecuta en la arquitectura DLX:</b>				<b>D</b>
18	A) No puede ejecutarse en una arquitectura súper escalar		C) Implica una detención por riesgo de tipo WAR	
	B) Debe ejecutarse en una arquitectura vectorial		D) Puede ejecutarse en una arquitectura súper escalar	
<b>Sea la secuencia de instrucciones LF F0, 10(R1) SF 10(R1), F0:</b>				<b>D</b>
19	A) Implica cargar en memoria el valor 0		C) Se beneficia si existe adelantamiento ALU-ALU	
	B) No es posible ejecutar esta secuencia de instrucciones		D) Se beneficia si existe adelantamiento MEM-MEM	
20	<b>Sea la secuencia de instrucciones:</b>	<b>DIVF F2, F0, F1 ADDF F3, F0, F2</b>	<b>ejecutadas sobre una arquitectura DLX con una unidades funcionales de coma flotante segmentadas:</b>	<b>B</b>
	A) Las fases de $EX_{DIV}$ y $EX_{ADD}$ pueden solaparse		C) Las fases de ejecución durarán 2 ciclos en total	
	B) Presenta un riesgo de datos de tipo RAW		D) Ninguna de las afirmaciones anteriores es correcta	
<b>Sea una arquitectura DLXV con longitud máxima de vector <math>MVL = 64</math> y sea X un vector de 124 elementos, sobre el que se hace la operación <math>aX+b</math>, siendo a y b escalares:</b>				<b>A</b>
21	A) Implica la ejecución de un bucle vectorial dos veces		C) Implica la ejecución de un bucle vectorial 64 veces	
	B) Implica la ejecución de un bucle vectorial 124 veces		D) Ninguna de las afirmaciones anteriores es correcta	
<b>Sean las sentencias de un bucle S1: <math>B(i) = A(i)/K</math> y S2: <math>C(i) = A(i) + A(i-1)</math>, con <math>i = 1...64</math>:</b>				<b>A</b>
22	A) No existe dependencia entre las sentencias		C) Existe riesgo de tipo WAW en S2 respecto a S1	
	B) Existe riesgo de tipo RAR en S2 respecto a S1		D) Ninguna de las afirmaciones anteriores es correcta	
<b>Sea un bucle con sentencias S1: <math>B(i) = A(i-1)*K</math> y S2: <math>C(i) = A(i-1) + B(i)</math>, con <math>i = 1...64</math>:</b>				<b>A</b>
23	A) Ninguna sentencia es vectorizable		C) Sólo la sentencia S1 es vectorizable	
	B) Las dos sentencias son vectorizables		D) Sólo la sentencia S2 no es vectorizable	
<b>Sea la arquitectura vectorial DLXV con vector de tamaño 64 elementos y cada elemento de 8 bytes. Si dispone de una memoria organiza en 8 bancos de memoria, cuando la CPU solicita la dirección 1 y siguientes:</b>				<b>A</b>
24	A) Ésta se ubicará en el Banco 0		C) Ésta se ubicará en el Banco 1	
	B) Ésta se ubicará en el Banco 7		D) Ésta se ubicará en el Banco 128	
<b>Sea la operación vectorial MULTV V2, V1, V0 (vectores de 128 elementos) con tiempo de arranque de 6 ciclos y tasa de iniciación de 1 ciclo/resultado, el tiempo de operación vectorial es de:</b>				<b>D</b>
25	A) 6 ciclos		C) $(3 \times 128 \times 6)$ ciclos	
	B) $(128 \times 6)$ ciclos		D) $(6 + 128 \times 1)$ ciclos	
<b>En base a la clasificación de Flynn sobre los sistemas computadores, en una arquitectura de tipo MISD:</b>				<b>B</b>
26	A) El flujo de instrucciones es compartido		C) Existen tantos flujos de datos como procesadores	
	B) La unidad de memoria es compartida		D) Ninguna de las afirmaciones anteriores es correcta	
<b>Comparando la estructura de un array sistólico respecto a un array de frente de onda:</b>				<b>B</b>
27	A) En el primero los procesadores comparten la memoria principal		C) En los dos casos, todos los procesadores comparten la Unidad de Control	
	B) En el primero los procesadores comparten la señal de reloj		D) En los dos casos, todos los procesadores comparten el mismo registro PC	
<b>Comparando la arquitectura escalar multihebra con entrelazado respecto a la arquitectura con bloqueo:</b>				<b>B</b>
28	A) La primera conmuta a otra hebra cuando finaliza con la anterior		C) La segunda conmuta a otra hebra cuando finaliza con la anterior	
	B) La primera conmuta a otra hebra después de cada ciclo		D) La primera ofrece varios cauces de trabajo	
<b>Sea una arquitectura DLX que emplea 60 ns en cada etapa (IF, ID, EX, MEM, WB). ¿Cuánto tiempo dedicará a la ejecución de 5 instrucciones consecutivas sin dependencia y sin ningún tipo de riesgo entre ellas?</b>				<b>B</b>
29	A) 300 ns		C) 1500 ns	
	B) 540 ns		D) Ninguna de las afirmaciones anteriores es correcta	
<b>En el diseño de una arquitectura de computador, indique la afirmación que es cierta:</b>				<b>D</b>
30	A) La arquitectura Von Neumann presenta más ventajas que el resto de arquitecturas estudiadas		C) El registro RI es independiente del repertorio de instrucciones	
	B) El tiempo de ejecución es independiente de la frecuencia de procesamiento		D) En la arquitectura DLX, las etapas para la ejecución de una instrucción pueden necesitar más de un ciclo	



EXAMEN DE ARQUITECTURA DE COMPUTADORES  
2º Curso Grado en Ingeniería Informática

La Rábida, 6 de Septiembre de 2019

**PROBLEMA 1.** (2,5 ptos.). Un sistema computador (sin memoria virtual) tiene una memoria principal con organización de 1 Gpalabras x 2 bytes cada palabra, dividida en 1 Mbloques y una memoria caché de 64 bloques y función de correspondencia directa.

1. Se pide:

- Justificar el tamaño de cada bloque (en bytes);
- Deducir la capacidad en bits de la Memoria Principal.

Respecto a la **Memoria Caché**, justificar:

- su número de palabras.
2. Mostrar el **formato de la dirección** que define la unidad central de proceso en base a la función de correspondencia empleada, definiendo cada uno de los campos en los que se divide.

Se supone que en el instante inicial la memoria caché está “vacía”, y a continuación, en el *instante 1* la CPU solicita los bloques de memoria principal desde el B64 al B126, leídos en ese orden y todas sus direcciones ordenadamente una vez.

- Mostrar el contenido de la memoria caché en el *instante 1*.
- Identificar cuántos fallos de caché se han producido y de qué tipo hasta ese *instante 1*.

Indicar, en binario las tres primeras direcciones que produjeron los tres primeros fallos.

A continuación, la CPU lee la secuencia de direcciones de memoria: 0, 1023, 1024, 1025 y 65536 (*instante 2*), según el orden marcado en la misma.

- Mostrar el contenido de la memoria caché en el *instante 2*.
- En ese *instante 2*, ¿cuántos fallos y de qué tipo se han producido en total?.
- Justificar la frecuencia de uso de cada bloque que se encuentra en caché en el instante 2, considerando el total de referencias desde el instante inicial.

**PROBLEMA 2.** (2,5 ptos.). Para el sistema computador representado en la figura, y para la secuencia de instrucciones siguiente:

<b>Dirección de Memoria</b> <b>(en hexadecimal)</b>	<b>Instrucción</b> <b>(en ensamblador)</b>
A0000h	DIV .2, 120(.3)
A0001h	BZ A0003h
A0002h	INC .1
A0003h	ADD 50(.4), .1

1. Mostrar los formatos de las instrucciones.
2. Definir la secuencia de operaciones elementales y el solapamiento posible de las mismas. Especificar el cronograma según el flujo marcado por el programa propuesto.
3. Definir el valor de las señales de control correspondiente a la secuencia ordenada de operaciones elementales **RD ← Desplazamiento (en RI) + R3** y **R2 ← R2/M(RD)**, realizadas en el menor tiempo posible. Mostrar únicamente el valor de las señales que intervienen directamente en dicha operación.

Teniendo en cuenta:

- a) PRIMERA INSTRUCCIÓN: La instrucción de división, con direccionamiento directo absoluto a registro para el primer operando y directo relativo para el segundo. El contenido de memoria  $M(120 + R3) = 0$ . La división por cero genera TRAP, no siendo necesario guardar el resultado erróneo de la operación. La dirección donde se encuentra la Rutina de Servicio a la Interrupción proporcionada por la propia CPU es A0002h
- b) SEGUNDA INSTRUCCIÓN: La instrucción de bifurcación condicional BZ, con modo de direccionamiento directo absoluto y considerando el biestable  $Z = 0$ .
- c) TERCERA INSTRUCCIÓN: La instrucción aritmética INC, con modo de direccionamiento directo absoluto a registro.
- d) CUARTA INSTRUCCIÓN: La instrucción aritmética ADD, con modo de direccionamiento directo relativo a registro para el primer operando y directo absoluto a registro para el segundo operando.
- e) Las operaciones con la unidad aritmético-lógica se realizan en un periodo.
- f) Los cálculos de las direcciones de los operandos en memoria especificados con direccionamiento relativo, se realizarán de manera automática en el periodo de decodificación.
- g) El resultado de una operación o el operando destino va a ser siempre el especificado en primer lugar en la instrucción de ensamblador.
- h) Las operaciones de L/E en memoria se realizan durante dos periodos.

- i) La pila, que se implementa en el sistema con el registro SP (puntero de pila), funciona de la siguiente forma: 1) La pila crece hacia direcciones decrecientes. 2) El registro puntero de pila SP indica la dirección de memoria principal en la que se introdujo el último dato en la pila. Antes de empezar a ejecutar la secuencia de instrucciones dadas, el registro SP contiene la dirección 0FFFh.
- j) Para cuando hay que salvar el contenido de SR y el de PC, siempre se almacenará primero el de PC.

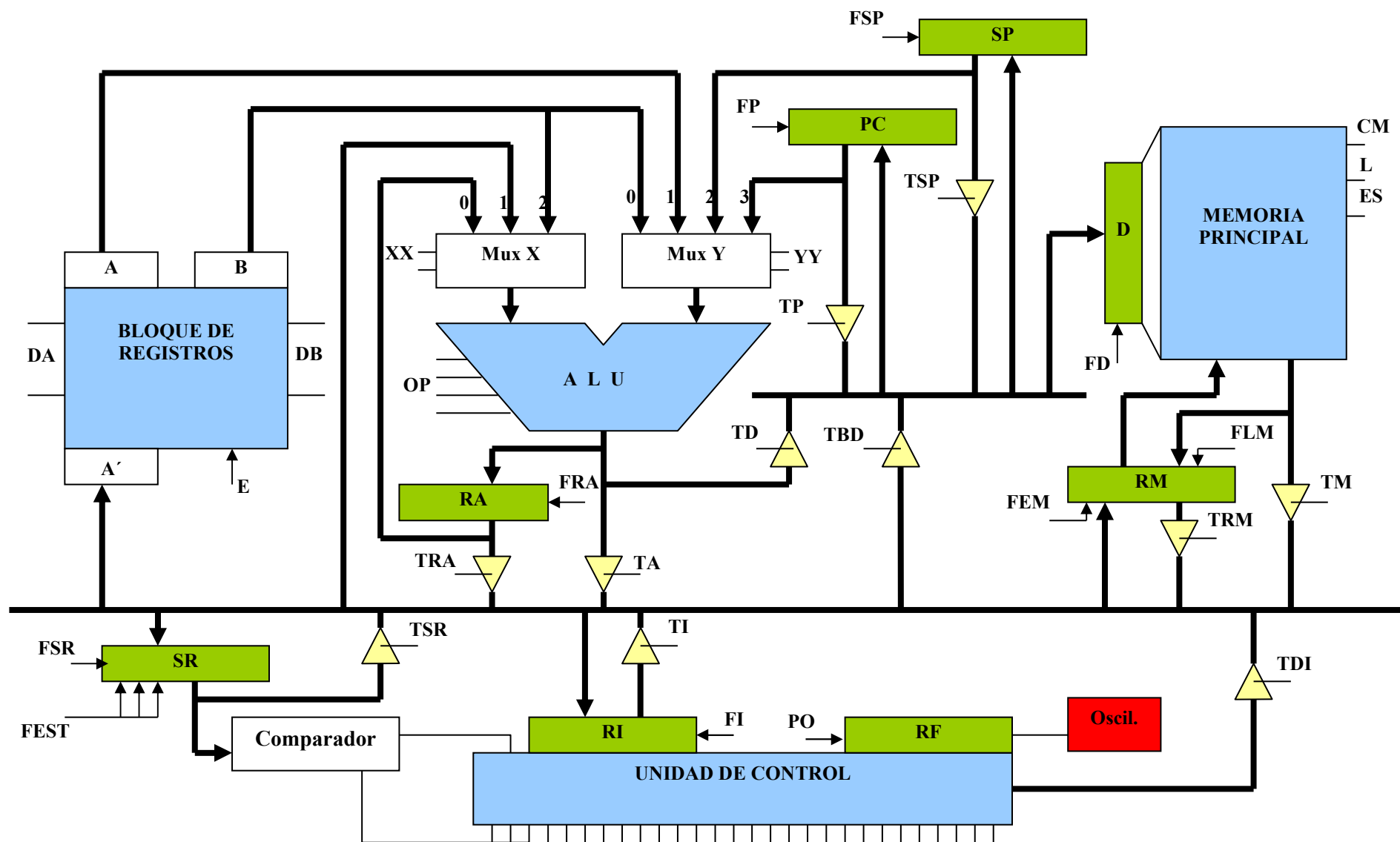
**PROBLEMA 3.** (2 ptos.). El siguiente fragmento de código se ejecuta en un procesador con arquitectura DLX. Las latencias (en ciclos de reloj) de las unidades funcionales son las siguientes: Sumador/restador entero: 1; Sumador/restador flotante: 2; y Multiplicador/Divisor: 4

```

ADDI R1, R2, #22
LW R1, 20(R4)
LW R2, 10(R5)
ADD R1, R1, R2
LF F2, 30(R1)
FADD F2, F4, F3
SF 30(R5), F2
SW 40(R6), R2

```

- a) Suponiendo que la **memoria caché está partida, no existe adelantamiento** y las **unidades funcionales para operaciones en coma flotante no están segmentadas**, indicar el estado de cada instrucción durante los ciclos de ejecución del código; calcular el número de ciclos necesarios para ejecutarlo; e indicar si existen bloqueos en la cadena y a qué se deben.
- b) Repetir del apartado anterior considerando que la **memoria caché está unificada, existe adelantamiento generalizado** y las **unidades funcionales para operaciones en coma flotante están segmentadas**. Indicar además los adelantamientos que se producen y entre qué salidas y entradas. En cuanto a los bloqueos, indicar de qué tipos son, aclarando por qué se producen (qué dato falta o qué elemento hardware es el que lo provoca).



Etapa	Instrucción ALU	Carga/Almacenamiento	Salto/Bifurcación
<b>IF</b>	$IR \leftarrow M[PC]$ $PC \leftarrow PC+4$	$IR \leftarrow M[PC]$ $PC \leftarrow PC+4$	$IR \leftarrow M[PC]$ $PC \leftarrow PC+4$
<b>ID</b>	$A \leftarrow Rs1; B \leftarrow Rs2$ $PC1 \leftarrow PC$ $IR1 \leftarrow IR$	$A \leftarrow Rs1; B \leftarrow Rs2$ $PC1 \leftarrow PC$ $IR1 \leftarrow IR$	$A \leftarrow Rs1; B \leftarrow Rs2$ $PC1 \leftarrow PC$ $IR1 \leftarrow IR$
<b>EX</b>	$ALU_{output} \leftarrow A \text{ op } (B \text{ or } (IR1_{16})^{16} \text{###} IR1_{16..31})$	$MAR \leftarrow A + (IR1_{16})^{16} \text{###} IR1_{16..31}$ $SMDR \leftarrow B$	<b>Salto</b> $ALU_{output} \leftarrow PC1 + IR1_{16})^{16} \text{###} IR1_{16..31})$ $Cond \leftarrow (Rs1 \text{ op } 0);$ <b>Bifurcación</b> $ALU_{output} \leftarrow PC1 + IR1_{26})^{6} \text{###} IR1_{6..31})$
<b>MEM</b>	$ALU_{output1} \leftarrow ALU_{output}$	$LMDR \leftarrow M[MAR]$ o $M[MAR] \leftarrow SMDR$	If (cond): $PC \leftarrow ALU_{output}$
<b>WB</b>	$Rd \leftarrow ALU_{output1}$	$Rd \leftarrow LMDR$	

**Tabla.** Eventos en cada etapa de la segmentación de DLX diferenciados según el tipo de instrucción.

**Apellidos:** ..... **Nombre:** .....

[illegible]

## 1) FORMATOS


### 3) SEÑALES DE CONTROL

[illegible]



Apellidos:

Nombre:

Apartado a)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
ADDI R1,R2,#22																														
LW R1,20(R4)																														
LW R2,10(R5)																														
ADD R1,R1,R2																														
LF F2,30(R1)																														
FADD F2,F4,F3																														
SF 30(R5),F2																														
SW 40(R6),R2																														
Detenciones							Detenciones										Adelantamientos						Adelantamientos							
Apartado b)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
ADDI R1,R2,#22																														
LW R1,20(R4)																														
LW R2,10(R5)																														
ADD R1,R1,R2																														
LF F2,30(R1)																														
FADD F2,F4,F3																														
SF 30(R5),F2																														
SW 40(R6),R2																														
Detenciones							Detenciones										Adelantamientos						Adelantamientos							