

**EXAMEN DE ARQUITECTURA DE COMPUTADORES. La Rábida, 7 de Septiembre de 2017**

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

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

TIEMPO REALIZACIÓN EXAMEN: 45 minutos

Espacio para respuestas

|           |  |          |
|-----------|--|----------|
| <b>1</b>  | <b>Atendiendo a las características de los sistemas micro, mini, gran y súper computadores, las máquinas desarrolladas por el fabricante CRAY que soportan cargas de trabajo del orden de multi-petaflops pueden considerarse:</b>   | <b>D</b> |
|           | A) Micro computadores C) Grandes computadores<br>B) Mini computadores D) Súper computadores  |          |
| <b>2</b>  | <b>Sea una arquitectura segmentada sobre la que aplicamos una modificación para aumentar la velocidad del procesador:</b>  | <b>A</b> |
|           | A) Estamos mejorando el tiempo de ejecución de instrucción C) Empeoramos la aceleración<br>B) Estamos mejorando sólo la productividad de la máquina D) No vamos a tener ninguna mejora   |          |
| <b>3</b>  | <b>Atendiendo a la definición de productividad estudiada para sistemas computadores:</b>   | <b>C</b> |
|           | A) Aumenta cuando aumenta el tiempo de ejecución de instrucción C) Aumenta cuando aumenta el número de tareas realizadas por unidad de tiempo<br>B) Disminuye cuando disminuye el tiempo de ejecución de instrucción D) Ninguna de las afirmaciones anteriores es correcta   |          |
| <b>4</b>  | <b>Sea un sistema computador del que conocemos su frecuencia de operación, <math>f = 3</math> GHz:</b>   | <b>B</b> |
|           | A) Podemos conocer su productividad C) Podemos conocer su CPI<br>B) Podemos conocer su tiempo de ciclo D) El tiempo de ciclo es independiente de la frecuencia   |          |
| <b>5</b>  | <b>Sea un microprocesador con un <math>CPI_{medio} = 3</math>, y frecuencia de operación de 2,5 GHz:</b>   | <b>B</b> |
|           | A) El tiempo de ciclo es de 1,2 ns C) El tiempo de ejecución de instrucción es de 3 ns<br>B) El tiempo de ejecución por instrucción=1,2 ns D) No es posible conocer el tiempo de ciclo   |          |
| <b>6</b>  | <b>Sea la máquina A que dedica 5 seg. en ejecutar un programa, mientras que la máquina B dedica 2,5 seg:</b>   | <b>C</b> |
|           | A) La máquina A es el 100% más rápida que B C) La máquina B es el 100% más rápida que A<br>B) La máquina B es el 50% más rápida que A D) No es posible comparar los dos rendimientos   |          |
| <b>7</b>  | <b>Sea el microprocesador Raspberry Pi 2 con 4 MIPS a 1 GHz:</b>   | <b>B</b> |
|           | A) Dedicar 1 ns a cada instrucción C) Dedicar 2,5 $\mu$ s a cada instrucción<br>B) Dedicar de media 250 ciclos para cada instrucción D) Dedicar 0,25 ciclos a cada instrucción   |          |
| <b>8</b>  | <b>Cuando se habla de algoritmos de reemplazo (Aleatorio, FIFO, LRU, LFU,...), nos referimos a técnicas de:</b>  | <b>D</b> |
|           | A) Búsqueda de los bloques en memoria C) Escritura o actualización de los bloques<br>B) Organización de la memoria caché D) Sustitución de bloques en memoria caché  |          |
| <b>9</b>  | <b>Atendiendo a los tipos de fallos de caché (forzosos, de conflicto, de capacidad):</b>   | <b>D</b> |
|           | A) El fallo forzoso se da siempre cuando la memoria caché está llena C) Después de un fallo de conflicto, siempre viene un fallo de capacidad<br>B) El fallo de capacidad se da cuando hay hueco en la memoria caché D) El fallo de conflicto se puede dar cuando hay hueco en la memoria caché                          |          |
| <b>10</b> | <b>Atendiendo al formato de dirección en un sistema computador con memoria caché:</b>  | <b>D</b> |
|           | A) Es el mismo para la función de correspondencia directa y totalmente asociativa C) No permite conocer cuántas palabras tiene cada bloque<br>B) Es el mismo para todas las funciones de correspondencia D) Ninguna de las afirmaciones anteriores es correcta   |          |
| <b>11</b> | <b>Para conocer si un bloque solicitado se encuentra en memoria caché:</b>   | <b>A</b> |
|           | A) La correspondencia T. Asociativa necesita tantos comparadores como bloques tiene la memoria caché C) La correspondencia T. Asociativa necesita tantos comparadores como bloques tiene la memoria principal<br>B) Siempre se necesitan un comparador por memoria D) Siempre se necesita un comparador por cada palabra |          |
| <b>12</b> | <b>La/s técnica/s estudiada/s para mejorar el rendimiento de un procesador son:</b>  | <b>D</b> |
|           | A) Memoria entrelazada y aumento del ancho de palabra C) Memoria superescalar<br>B) Memoria multinivel D) Acelerar el caso común (ley de Amdahl)   |          |
| <b>13</b> | <b>De los algoritmos de reemplazo, el que penaliza al bloque más recientemente introducido en memoria caché es el:</b>   | <b>B</b> |
|           | A) FIFO C) Aleatorio<br>B) LFU D) LRU  |          |
| <b>14</b> | <b>De las estrategias de escritura para reemplazar un bloque de memoria caché:</b>   | <b>D</b> |
|           | A) La postescritura aumenta los accesos a M. Principal C) La escritura nunca implica acceso a M. Principal<br>B) Todas suponen el mismo número de accesos D) Ninguna de las afirmaciones anteriores es correcta  |          |
| <b>15</b> | <b>Indica el lugar donde pueden estar almacenados los operandos en una máquina de 2 direcciones:</b>   | <b>A</b> |
|           | A) Banco de registros, Memoria y Pila C) Banco de Registros exclusivamente<br>B) Registro Acumulador D) Ninguna de las afirmaciones anteriores es correcta   |          |

|    |  |   |   |
|----|--|---|---|
| 16 | Sea la ejecución de la instrucción ADD .4, .7 en un sistema con arquitectura Von Neumann, si implica 4 ciclos de reloj y la activación de 17 señales de control:   |   | C |
|    | A) La instrucción la componen 4 microprogramas<br>B) La instrucción la componen (17 + 4) microinstrucciones  |   |   |
| 17 | Sea una unidad de control microprogramada con secuenciamiento explícito y 8 bits en el campo código de operación y memoria de control de 500 posiciones. Indica cuál de las siguientes afirmaciones es cierta: |   | D |
|    | A) La etapa traductora será de 256 posiciones x 500<br>B) La etapa traductora será de 256 posiciones x 9   |   |   |
| 18 | De las siguientes instrucciones, indica las que son de salto condicional:  |   | B |
|    | A) RETI y CMP .4, #0<br>B) BZ A0005h   |   |   |
| 19 | Sea la arquitectura DLX estudiada que no realiza operaciones multiciclo:   |   | A |
|    | A) Los riesgos por dependencia de datos serán de tipo RAW<br>B) Sólo presenta riesgos estructurales  |   |   |
| 20 | Si durante el arranque de un sistema computador, el fabricante ha programado el registro SR para que inicialmente contenga el valor F0000h:  |   | D |
|    | A) El resto de registros deben contener ese mismo valor<br>B) El programa cargador (boot-strap) estará almacenado a partir de esa dirección  |   |   |
| 21 | Atendiendo al esquema de la arquitectura DLX estudiada en el Tema 3 y 4, corresponde a:  |   | B |
|    | A) Un computador de 4 direcciones<br>B) Un computador de 3 direcciones   |   |   |
| 22 | Sea un sistema del que se dice que la pila crece hacia direcciones crecientes:   |   | C |
|    | A) La ejecución de RET implica incrementar SR<br>B) La ejecución de RET implica incrementar SP   |   |   |
| 23 | Sea un sistema con unidad de control microcableada, una instrucción de bifurcación condicional:  |   | B |
|    | A) Tiene almacenado un único microprograma<br>B) Estos sistemas no almacenan microprogramas  |   |   |
| 24 | Sea un sistema con arquitectura Von Neumann, ancho de palabra de memoria de 16 bits y formato de instrucción de 32 bits:   |   | D |
|    | A) El incremento de RI se hará en 2 unidades<br>B) El incremento de PC y SP se hará en 2 unidades  |   |   |
| 25 | Sea la arquitectura DLX, al ejecutar la operación LH R2, 10(R1) (Carga en parte alta del registro):  |   | C |
|    | A Se modificará todo el registro R2<br>B) Se modificará todo el registro R1  |   |   |
| 26 | La arquitectura supersegmentada:   |   | D |
|    | A) Obliga a realizar a la vez las mismas fases de una misma instrucción<br>B) Permite realizar a la vez las mismas fases de distintas instrucciones  |   |   |
| 27 | Sea el extracto del bucle:<br>LOOP:<br>ADDF F0, F2, F4<br>SF 0(R1), F0<br>SUB R1, R1, #8<br>BNEZ R1, LOOP  | que pasa a ser de la forma:<br>LOOP:<br>ADDF F0, F2, F4<br>SF 0(R1), F0<br>SUB R1, R1, #8<br>ADDF F0, F2, F4<br>SF 0(R1), F0<br>SUB R1, R1, #8<br>..... | A |
|    | estaríamos aplicando:<br>A) Desenrollado del bucle<br>B) Planificación de traza  |   |   |
| 28 | El objetivo de la arquitectura VLIW es:  |   | B |
|    | A) Tener un registro de instrucción de mayor tamaño<br>B) Reduce el número de detenciones  |   |   |
| 29 | En la arquitectura DLX con la estructura revisada para las instrucciones de salto, la ejecución de la instrucción BEQZ R1, A0000h implica:   |   | D |
|    | A) Un riesgo de control<br>B) Un riesgo por dependencia de datos   |   |   |
| 30 | En un sistema computador, las características básicas que informan de su rendimiento son:  |   | D |
|    | A) Capacidad y velocidad de memoria<br>B) Frecuencia, CPI, recuento de instrucciones y capacidad de memoria  |   |   |

## EXAMEN DE ARQUITECTURA DE COMPUTADORES

La Rábida, 7 de Septiembre de 2017

**PROBLEMA 1.** (2,5 ptos.). Un sistema computador (sin memoria virtual) tiene una memoria principal de 1024 Kpalabras de 8 bits cada una de ellas y una memoria caché de 16 Kpalabras, con función de correspondencia Asociativa por Conjuntos, bloques con un tamaño de 512 palabras y conjuntos de 8 bloques.

- a) Justificar e indicar para la **Memoria Principal** y para la **Memoria Caché**:
  - 1) La capacidad en bytes.
  - 2) El número de bloques.
- b) 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, después de haber estado la memoria caché “vacía” (*instante 0*), a continuación en el *instante 1* se encuentran en la memoria caché los bloques de memoria principal B0, B1, B2, B3, B4, B5, B6, B8, B9 y B12, leídos en ese orden y todas sus direcciones ordenadamente una vez. Suponer que el algoritmo de reemplazamiento de bloques es el FIFO (First In, First Out).

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

A continuación, la CPU lee la secuencia de direcciones de memoria: 511, 1024, 1536, y 5120 (*instante 2*), según el orden marcado en la misma.

- e) Mostrar el contenido de la memoria caché en el *instante 2*.
- f) Entre el *instante 1* y el *instante 2*, ¿cuántos fallos y de qué tipo se han producido?.
- g) Desde el *instante 0* hasta el *instante 2*, ¿cuántos aciertos se han producido?.
- h) Indicar la frecuencia de uso de cada uno de los bloques contenidos en la memoria caché en el *instante 2*.

A continuación, la CPU lee las direcciones 6145, 8192 y 8193 (*instante 3*), en ese orden.

- i) Mostrar el contenido de la Memoria Caché en el *instante 3*.
- j) Entre el *instante 2* y el *instante 3*, ¿cuántos aciertos y fallos se han producido y en qué direcciones?
- k) Hasta ese *instante 3*, ¿cuál es la frecuencia de uso del bloque 12?

**PROBLEMA 2.** (2,25 ptos.). Para el sistema computador representado en la figura, y para la secuencia de instrucciones siguiente:

| Dirección de Memoria<br>(en hexadecimal) | Instrucción<br>(en ensamblador) |
|--|---------------------------------|
| B0000h                                   | DIV .1, .2                      |
| B0001h                                   | NOP                             |
| C0000h                                   | MOVE [00120h], .5               |
| C0001h                                   | RETI                            |

1. Mostrar los formatos de las instrucciones.
2. Definir la secuencia de operaciones elementales y el solapamiento posible de las mismas. Especificar la secuencia según el flujo marcado por el programa propuesto, y considerando una sola vez la ejecución de cada instrucción. La última instrucción ejecutada es la de NOP.
3. Definir el valor de las señales de control correspondiente a la secuencia de operaciones elementales:  $SP, D \leftarrow SP+1$  y  $RM \leftarrow SR$ , 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, DIV, con direccionamiento directo absoluto a registro. Al comienzo de la ejecución, el contenido del registro R2 es cero, provocando un cepo o trap en la ejecución de la instrucción. La dirección donde se encuentra el comienzo de la Rutina de Servicio a la Interrupción es la C0000h
- b) SEGUNDA INSTRUCCIÓN: La instrucción de No Operación, NOP.
- c) TERCERA INSTRUCCIÓN: La instrucción de transferencia MOVE, con modo de direccionamiento directo absoluto a registro para el primer operando e indirecto absoluto a memoria para el segundo operando.
- d) CUARTA INSTRUCCIÓN: La instrucción de retorno de interrupción, RETI.
- 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 después del 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

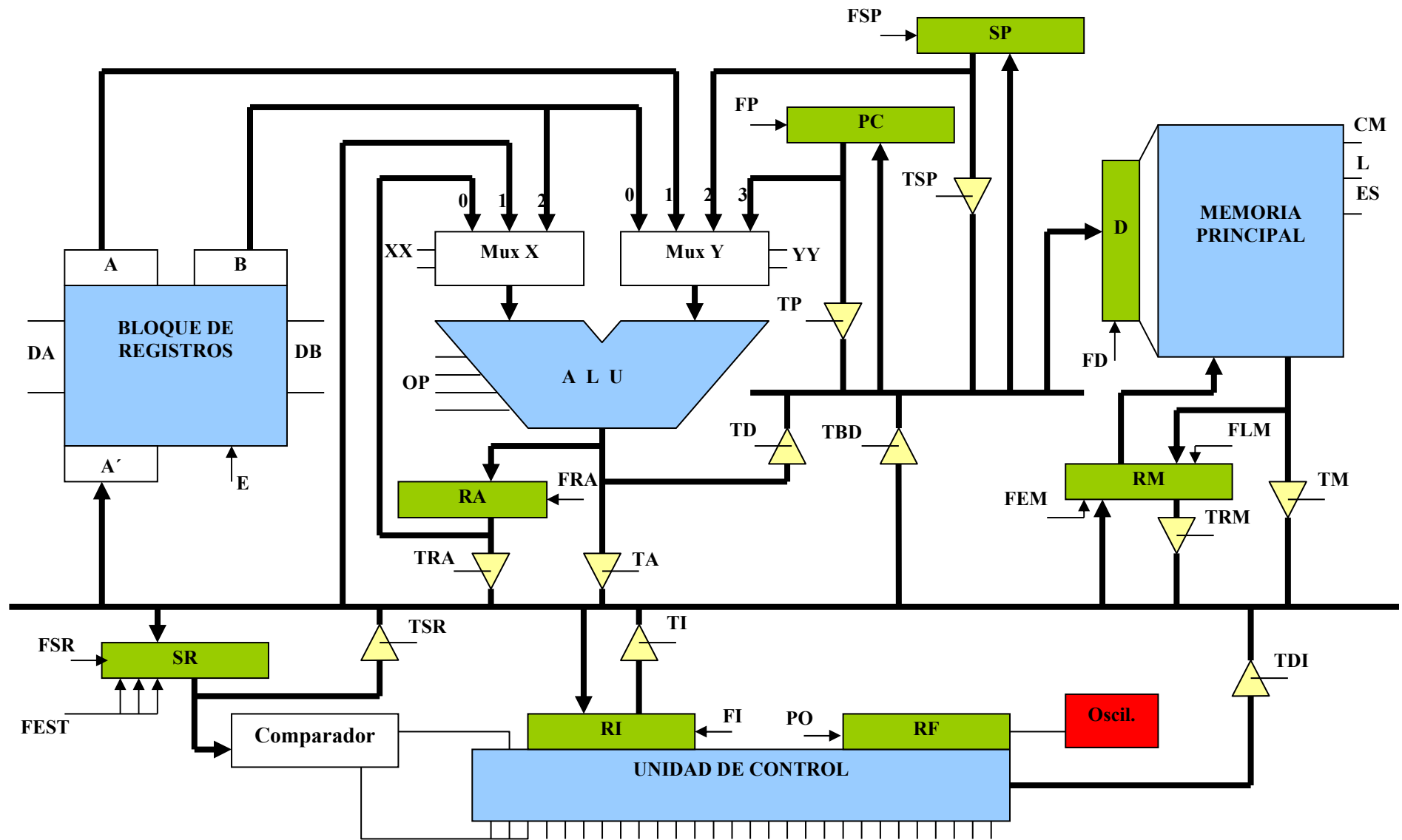
de pila indica la dirección de memoria principal en la que se va a introducir el próximo dato en la pila.

- j) Para cuando hay que salvar el contenido de SR y el de PC, siempre se almacenará primero el de SR.

**PROBLEMA 3.** (2,25 ptos.). El siguiente fragmento de código se ejecuta en un procesador DLX. En dicho procesador la etapa MEM requiere 1 ciclo de reloj. Las latencias (en ciclos de reloj) de las unidades funcionales son las siguientes: Sumador/restador entero: 1; Sumador/restador flotante: 2; y Multiplicador/Divisor: 3 (las dos últimas unidades funcionales para operaciones están **segmentadas**).

```
LW R2, 20(R1)
SW 10(R2), R5
LW R3, 10(R5)
LF F0, 30(R5)
ADD R4, R3, R5
SUB R5, R3, R4
FMUL F6, F4, F3
FADD F4, F0, F6
```

- a) Suponiendo que la **memoria** tiene un **único puerto de acceso** y que **no hay adelantamiento**, indicar el estado de cada instrucción durante los ciclos de ejecución del código; calcular el número de ciclos necesarios para ejecutar este código; e indicar si existen bloqueos en la cadena y a qué se deben.
- b) Repetir el apartado a) para una versión del DLX que incluya **adelantamiento** (bypass) **generalizado** y suponiendo que la **memoria** tiene **dos puertos de acceso** (uno para instrucciones y otro para datos). Indicar el estado de cada instrucción durante los ciclos de ejecución del código; calcular el número de ciclos necesarios para ejecutar este código; e indicar si existen bloqueos en la cadena y a qué se deben. Indicar claramente los datos que se adelantan y entre qué etapas actúa el bypass.



| <b>Etap</b> | <b>Instrucción ALU</b>   | <b>Carga/Almacenamiento</b>  | <b>Salto/Bifurcación</b>  |
|-------------|--|--|---|
| <b>IF</b>   | $IR \leftarrow M[PC]$<br>$PC \leftarrow PC+4$  | $IR \leftarrow M[PC]$<br>$PC \leftarrow PC+4$                                      | $IR \leftarrow M[PC]$<br>$PC \leftarrow PC+4$   |
| <b>ID</b>   | $A \leftarrow Rs1; B \leftarrow Rs2$<br>$PC1 \leftarrow PC$<br>$IR1 \leftarrow IR$     | $A \leftarrow Rs1; B \leftarrow Rs2$<br>$PC1 \leftarrow PC$<br>$IR1 \leftarrow IR$ | $A \leftarrow Rs1; B \leftarrow Rs2$<br>$PC1 \leftarrow PC$<br>$IR1 \leftarrow IR$  |
| <b>EX</b>   | $ALU_{output} \leftarrow A \text{ op } (B \text{ or } (IR1_{16})^{16}###IR1_{16..31})$ | $MAR \leftarrow A+(IR1_{16})^{16}###IR1_{16..31}$<br>$SMDR \leftarrow B$           | <div> <b>Salto</b><br/> <math>ALU_{output} \leftarrow PC1+IR1_{16})^{16}###IR1_{16..31})</math><br/> <math>Cond \leftarrow (Rs1 \text{ op } 0);</math> </div> <div> <b>Bifurcación</b><br/> <math>ALU_{output} \leftarrow PC1+ IR1_{26})^6###IR1_{6..31})</math> </div> |
| <b>MEM</b>  | $ALU_{output1} \leftarrow ALU_{output}$  | $LMDR \leftarrow M[MAR]$<br>o<br>$M[MAR] \leftarrow SMDR$                          | If (cond):<br>$PC \leftarrow ALU_{output}$  |
| <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 |
|-----------------|---|---|---|---|---|---|-------------|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| LW R2, 20(R1)   |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| SW 10(R2), R5   |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LW R3, 10(R5)   |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LF F0, 30(R5)   |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ADD R4, R3, R5  |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| SUB R5, R3, R4  |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| FMUL F6, F4, F3 |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| FADD F4, F0, F6 |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Detenciones     |   |   |   |   |   |   | Detenciones |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

| 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 |
|-----------------|---|---|---|---|---|---|-------------|---|---|----|----|----|----|----|----|----|-----------------|----|----|----|----|----|-----------------|----|----|----|----|----|----|----|
| LW R2, 20(R1)   |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |    |    |
| SW 10(R2), R5   |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |    |    |
| LW R3, 10(R5)   |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |    |    |
| LF F0, 30(R5)   |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |    |    |
| ADD R4, R3, R5  |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |    |    |
| SUB R5, R3, R4  |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |    |    |
| FMUL F6, F4, F3 |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |    |    |
| FADD F4, F0, F6 |   |   |   |   |   |   |             |   |   |    |    |    |    |    |    |    |                 |    |    |    |    |    |                 |    |    |    |    |    |    |    |
| Detenciones     |   |   |   |   |   |   | Detenciones |   |   |    |    |    |    |    |    |    | Adelantamientos |    |    |    |    |    | Adelantamientos |    |    |    |    |    |    |    |