

EXAMEN DE ARQUITECTURA DE COMPUTADORES. La Rábida, 21 de Junio de 2019

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	<p>Sea un sistema computador A que dedica 10 segundos a un proceso, si se dice que es el 20% más rápido que el computador B:</p> <p>A) El computador B dedica 12 segundos al mismo proceso C) El computador A dedica 2 segundos más que el B B) El computador B dedica 20 segundos al mismo proceso D) El computador A dedica 0,2 segundos menos que el B</p>	A
2	<p>Sea una computador sobre el que se aplica una mejora que hace que aumente su velocidad en un factor de x5 durante el 60% del tiempo: [aceleración global = 1/(fracción sin mejora + (fracción mejorada/aceleración mejorada))]</p> <p>A) La aceleración global conseguida es de 1.92 C) La fracción mejorada es del 40% B) La aceleración global conseguida es de 5 D) La fracción mejorada es de 0.3</p>	A
3	<p>En un sistema computador con frecuencia de procesamiento de 1 GHz, dedica 30 ns a ejecutar 5 instrucciones:</p> <p>A) Equivale a 6 CPI C) Equivale a 1 ns por instrucción B) Equivale a 150 CPI D) Equivale a 0.6 ns por instrucción</p>	A
4	<p>Sea un computador personal con procesador i7 a 2 GH, 12 GB de RAM y 512 GB de SSD, atendiendo a la clasificación estudiada, se considera:</p> <p>A) Un microcomputador C) Un súper computador B) Un gran computador D) Ninguna de las afirmaciones anteriores es correcta</p>	A
5	<p>Sea un sistema computador con rendimiento valorado en 2 MIPS, ¿cuánto tiempo dedicará a la ejecución de un programa formado por 100 instrucciones?:</p> <p>A) 200 ms C) 50 ms B) 200 μs D) 50 μs</p>	D
6	<p>En relación a la M. Principal y M. Caché, indique la afirmación que NO es cierta:</p> <p>A) La M. Caché contiene una copia parcial de la MP C) La MP tiene mayor tamaño que la M. Caché B) La transferencia de información se hace a nivel de palabra D) MP y M. Caché comparten el mismo tamaño de bloque</p>	B
7	<p>En relación a las técnicas de mejora de rendimiento de memoria:</p> <p>A) Bancos de memoria se aplica a M. Caché C) Memoria ancha se aplica a disco duro B) Bancos de memoria y memoria ancha se aplican a MP D) Ninguna de las afirmaciones anteriores es correcta</p>	B
8	<p>De las funciones de correspondencia de la M. Caché:</p> <p>A) La asociativa por conjuntos reduce el tamaño de bloque C) La totalmente asociativa genera fallos de conflicto B) La directa no requiere definir algoritmo de reemplazo D) Las tres comparten el mismo formato de dirección</p>	B
9	<p>Para localizar el bloque donde se encuentra la palabra solicitada por la CPU, la función de correspondencia asociativa por conjuntos, necesita:</p> <p>A) Un solo elemento comparador C) Tantos elementos comparadores como conjuntos tiene B) Tantos elementos comparadores como bloques formen el conjunto D) Tantos elementos comparadores como bloques tiene en total</p>	B
10	<p>Sea una memoria caché de dos niveles L1 y L2, si de 1000 referencias a memorias solicitadas por la CPU, el nivel L1 contiene 700 aciertos y L2 250 aciertos:</p> <p>A) La tasa local de fallos de L1 es del 7% C) La tasa global de fallos es del 5% B) La tasa local de fallos de L2 es del 25% D) La tasa global de aciertos es del 950%</p>	C
11	<p>Sea la secuencia de instrucciones LOAD A, SUB B, STORE C, con A, B, C posiciones de memoria:</p> <p>A) Implementa la operación aritmética C = A-B para un computador de 0 dirección C) Implementa la operación aritmética C = A-B para un computador de 1 dirección B) No implementa ninguna operación aritmética D) Implementa la operación aritmética C = A-B para un computador de 3 direcciones</p>	C
12	<p>Sea la arquitectura Von Neumann estudiada, para realizar una operación de acceso a memoria:</p> <p>A) Se necesita definir el valor del registro de Datos (RM) siempre C) Se necesita definir el valor del registro de Direcciones siempre B) Se necesita definir el valor del registro PC D) Se necesita definir el valor del registro SP</p>	C
13	<p>Sea un computador con unidad de control microprogramada con secuenciamiento implícito, repertorio de 512 instrucciones y Memoria de Control de 2 Kpos:</p> <p>A) La etapa traductora ROM es de 9 pos x 11 bits C) La etapa traductora ROM es de 512 pos x 11 bits B) No necesita etapa traductora D) Ninguna de las afirmaciones anteriores es correcta</p>	C
14	<p>Sea un computador con unidad de control microprogramada con secuenciamiento implícito, y Memoria de Control de 2 Kpos:</p> <p>A) El repertorio está formado por 2 K instrucciones C) El firmware lo conforman 2¹¹ microinstrucciones B) El firmware lo conforman 11 microinstrucciones D) Faltan datos para conocer el nº de microinstrucciones</p>	C

Sea un sistema con arquitectura von Neumann, indica las instrucciones que implican acceso de lectura en pila:				
15	A) CALL y BZ B) Sólo la instrucción RETI		C) ADD .2, .4 con trap por desbordamiento D) RET y RETI	D
Comparando la arquitectura Von Neumann con la arquitectura DLX:				
16	A) Ambas son arquitecturas segmentadas B) La arquitectura Von Neumann aumenta el paralelismo a nivel de instrucción		C) Ambas son arquitecturas serie D) La arquitectura DLX aumenta el paralelismo a nivel de instrucción	D
Sea la instrucción LF F0, 10(R1) ejecutada en una arquitectura DLX:				
17	A) La fase de ejecución requiere de una unidad funcional de tipo flotante B) No puede ejecutarse porque F0 = 0		C) No puede ejecutarse porque R1 debe ser de tipo float D) La fase de ejecución pasa por la unidad funcional de enteros	D
Sea la arquitectura DLX estudiada con unidades funcionales de tipo float multiciclo:				
18	A) Sólo presentará riesgos de tipo RAW B) Puede presentar riesgos de tipo RAW, WAW y WAR		C) Puede presentar riesgos de tipo WAW y WAR D) Puede presentar riesgos de tipo RAW y WAR	D
Sea la secuencia de instrucciones LF F0, 10(R1) ADDF F2, F0, F2:				
19	A) Se beneficia si existe adelantamiento ALU-MEM B) Se beneficia si existe adelantamiento ALU-ALU		C) Se beneficia si existe adelantamiento MEM-MEM D) Se beneficia si existe adelantamiento MEM-ALU	D
20	Sea el conjunto de instrucciones:	MULTF F2, F0, F1 DIVF F3, F0, F1	ejecutadas sobre una arquitectura DLX con una unidad funcional de Multiplicación/División flotante de 5 ciclos no segmentada:	A
	A) Las fases de ejecución durarán 10 ciclos en total B) Las fases de ejecución durarán 6 ciclos en total		C) Presenta un riesgo de control D) Ninguna de las afirmaciones anteriores es correcta	
Para implementar la operación vectorial F0*V1, siendo F0 un registro tipo float y V1 un vector de 64 elementos:				
21	A) No se puede multiplicar un registro float por un registro vectorial B) Se debe emplear la instrucción MULTV y repetirla en un bucle un total de 64 veces en la arquitectura DLXV		C) Se emplearía la instrucción vectorial MULTV en la arquitectura DLX escalar D) Se emplearía la instrucción vectorial MULTV en la arquitectura DLXV	D
Sea X un vector de 255 elementos sobre el que se realiza una operación vectorial en una arquitectura DLXV con longitud máxima de vector MVL = 64:				
22	A) El vector se divide en 3 secciones de 64 elementos y una última de 63 B) El vector se opera en su totalidad sin seccionamiento		C) El vector se divide en 4 secciones de 64 elementos D) El vector se divide en 2 secciones de 128 elementos	A
Sea un bucle con sentencias S1: B(i) = A(i)*K y S2: C(i) = B(i-1) + A(i), con i = 1...64:				
23	A) La sentencia S1 no es vectorizable B) Las dos sentencias son vectorizables		C) Ninguna sentencia es vectorizable D) La sentencia S2 no es vectorizable	D
Sea la arquitectura DLXV con 16 bancos de memoria, si la CPU solicita la dirección 128 y siguientes:				
24	A) Ésta se ubicará en el Banco 0 B) Ésta se ubicará en el Banco 16		C) Ésta se ubicará en el Banco 1 D) Ésta se ubicará en el Banco 128	A
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:				
25	A) 6 ciclos B) 128 x 6 ciclos		C) 3x 128 x 6 ciclos D) 134 ciclos	D
En base a la clasificación de Flynn sobre los sistemas computadores, en una arquitectura de tipo MISD:				
26	A) Existen tantos flujos de instrucciones como procesadores B) Existe una única unidad de procesamiento		C) El flujo de instrucciones es compartido D) Ninguna de las afirmaciones anteriores es correcta	A
Atendiendo a la clasificación de los multiprocesadores estudiada, en el multiprocesador UMA:				
27	A) Cada procesador tiene su propia memoria local B) Todos los procesadores tienen el mismo tiempo de acceso de memoria		C) El tiempo de acceso depende de la posición del procesador D) Todos los procesadores comparten la misma memoria caché	B
Sea el polinomio R(x) = ax+b, si se desea calcular empleando una arquitectura súper-escalar tipo array sistólico:				
28	A) Es suficiente un 1 ciclo y 2 elementos de proceso B) Se necesitan mínimo 3 ciclos y 3 elementos de proceso		C) Es posible con 2 ciclos y 1 elemento de proceso D) Se necesitan mínimo 2 ciclos y 3 elementos de proceso	C
Considerando una arquitectura súper-escalar multihebra con bloqueo:				
29	A) Conmuta a otra hebra después de cada ciclo B) Cada cauce trabaja sobre una única hebra		C) Conmuta a otra hebra cuando finaliza con la anterior D) Conmuta a otra hebra después de cada detención	D
En el diseño de un arquitectura de computador, indique la afirmación que NO es cierta:				
30	A) La arquitectura Von Neumann presenta más ventajas que el resto de arquitecturas estudiadas B) La frecuencia de procesamiento condiciona el tiempo de ejecución		C) El repertorio de instrucciones condiciona el registro RI D) En la arquitectura DLX, las etapas para la ejecución de una instrucción pueden necesitar más de un ciclo	A

EXAMEN DE ARQUITECTURA DE COMPUTADORES

La Rábida, 21 de Junio de 2019

PROBLEMA 1. (2,5 ptos.). Un sistema computador (sin memoria virtual) tiene una memoria principal de 256 Kpalabras de 8 bits cada una de ellas, dividida en bloques de 64 palabras/bloque; y una memoria caché de 512 Bytes, dividida en 2 conjuntos.

1. Justificar e indicar, de la **Memoria Principal**:

- a. su capacidad en bits
- b. el número de bloques que tiene;

y, de la **Memoria Caché**:

- c. su capacidad en bits
- d. su número de palabras
- e. su número de bloques
- f. su número de bloques por conjunto
- g. su número de palabras por bloque.

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, después de haber estado la memoria caché “vacía”, a continuación, en el *instante 1* se encuentran en la memoria caché los bloques de memoria principal B6, B8, B15, B17, B0, B1 y B2, leídos en ese orden y todas sus direcciones ordenadamente una vez. Suponer que el algoritmo de reemplazamiento de bloques es el LRU (Least Recently Used).

3. Mostrar el contenido de la memoria caché en el *instante 1*.

4. Identificar cuántos fallos de caché se han producido y de qué tipo hasta ese *instante 1*. Indicar, en binario y en decimal, con la lectura de qué dirección/es se produjeron el/los fallo/s.

A continuación, la CPU lee la secuencia de direcciones de memoria: 385, 520 y 260 (*instante 2*), según el orden marcado en la misma.

5. Mostrar el contenido de la memoria caché en el *instante 2*.

6. Hasta ese *instante 2*, ¿cuántos fallos y de qué tipo se han producido en total?. Indicar, en binario y en decimal, la/s dirección/es con la que se ha/n producido el/los nuevo/s fallo/s.

7. Hasta ese *instante 2*, ¿cuántos aciertos se han producido?.

Por último, la CPU lee la dirección 3 (*instante 3*).

8. Mostrar el contenido de la memoria caché en el *instante 3*.

9. Con esas última lectura, ¿se ha producido un fallo o un acierto?. Si ha sido un fallo, ¿de qué tipo?.

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

Dirección de Memoria (en hexadecimal)	Instrucción (en ensamblador)
A0000h	BNZ A0002h
A0001h	MOVE .2, 120(.3)
A0002h	CALL A0003h
A0003h	RETI

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, y **considerando una sola vez la ejecución de cada instrucción.**
3. Definir el valor de las señales de control correspondiente a la secuencia ordenada de operaciones elementales $D \leftarrow PC$ y $PC \leftarrow \text{Desplazamiento (en RI)} + R3$, 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 bifurcación condicional BNZ, con direccionamiento directo absoluto, para el caso en que el bit Z esté a "0".
- b) SEGUNDA INSTRUCCIÓN: La instrucción de transferencia MOVE, con modo de direccionamiento directo absoluto a registro para el primer operando y directo relativo a registro base para el segundo operando.
- c) TERCERA INSTRUCCIÓN: La instrucción de llamada a subrutina CALL, con modo de direccionamiento directo absoluto a memoria.
- d) CUARTA INSTRUCCIÓN: La instrucción RETI de retorno de interrupción.
- 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 indica la dirección de memoria principal en la que se introdujo el último dato en la

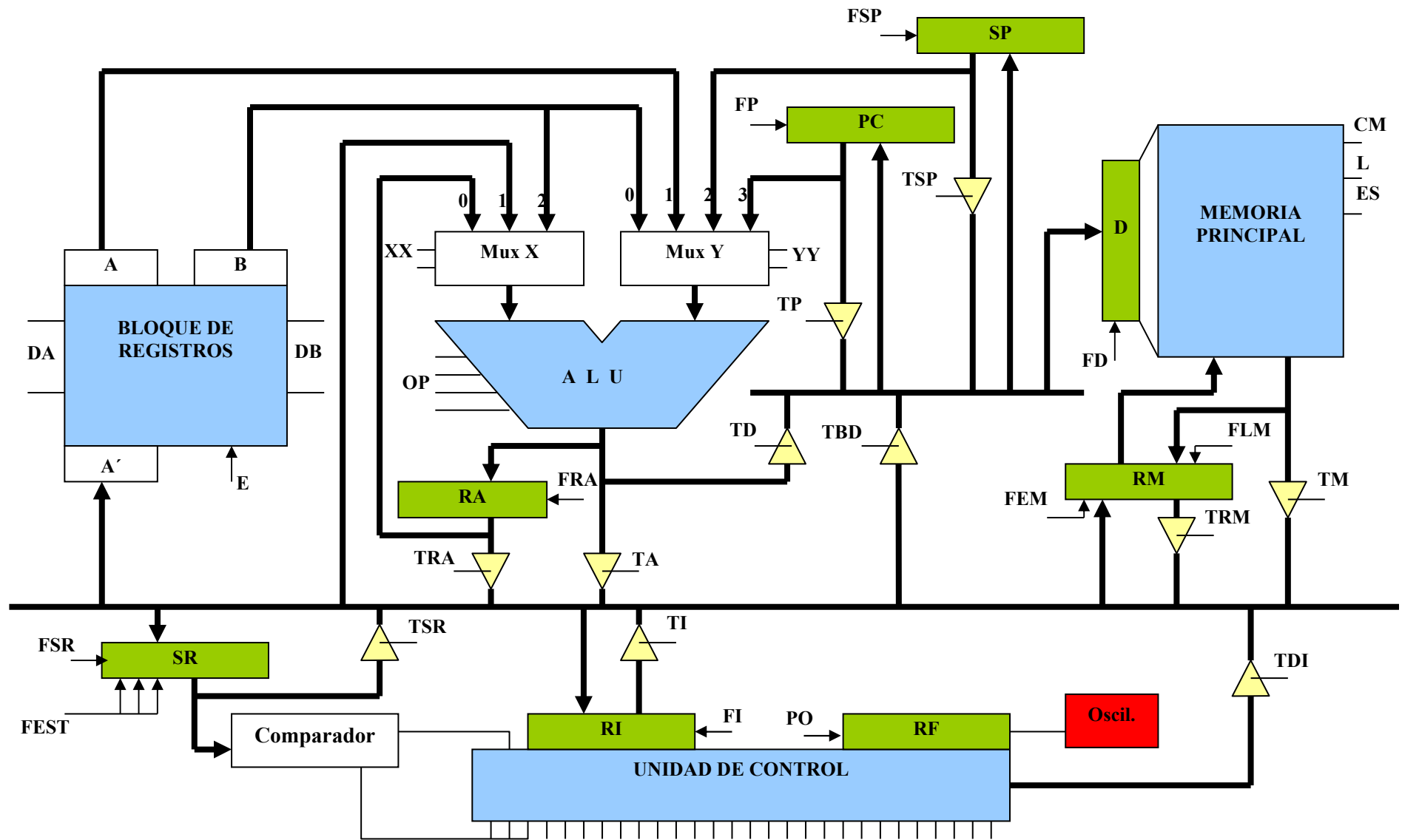
pila. **Antes de empezar a ejecutar la secuencia de instrucciones dadas, la pila tiene en su cabecera el valor A0001h.**

- 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 (las unidades funcionales para operaciones en coma flotante **no están segmentadas**).

```
LF F2, 20(R1)
SF 30(R1), F2
ADDI R1, R1, #4
FMULT F6, F7, F8
FADD F6, F7, F9
FDIV F1, F2, F3
FADD F3, F1, F6
SF 30(R0), F1
```

- a) Suponiendo que la **memoria caché está unificada** y **no existe adelantamiento**, indicar el estado de cada instrucción durante los ciclos de ejecución del código; calcular el número de ciclos necesarios para ejecutar este código; indicar si existen bloqueos en la cadena y a qué se deben.
- b) Repetir del apartado anterior considerando que **adelantamiento generalizado** y las unidades funcionales para operaciones en coma flotante **están segmentadas**.



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

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

A full page of blank graph paper. The grid consists of small squares formed by thin, dotted lines. There are 20 columns and 20 rows of squares, creating a total area of 400 small squares. The margins are uniform on all sides.

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
LF F2, 20(R1)																														
SF 30(R1), F2																														
ADDI R1, R1, #4																														
FMULT F6, F7, F8																														
FADD F6, F7, F9																														
FDIV F1, F2, F3																														
FADD F3, F1, F6																														
SF 30(R0), F1																														
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
LF F2, 20(R1)																														
SF 30(R1), F2																														
ADDI R1, R1, #4																														
FMULT F6, F7, F8																														
FADD F6, F7, F9																														
FDIV F1, F2, F3																														
FADD F3, F1, F6																														
SF 30(R0), F1																														
Detenciones							Detenciones										Adelantamientos							Adelantamientos						

Apellidos:

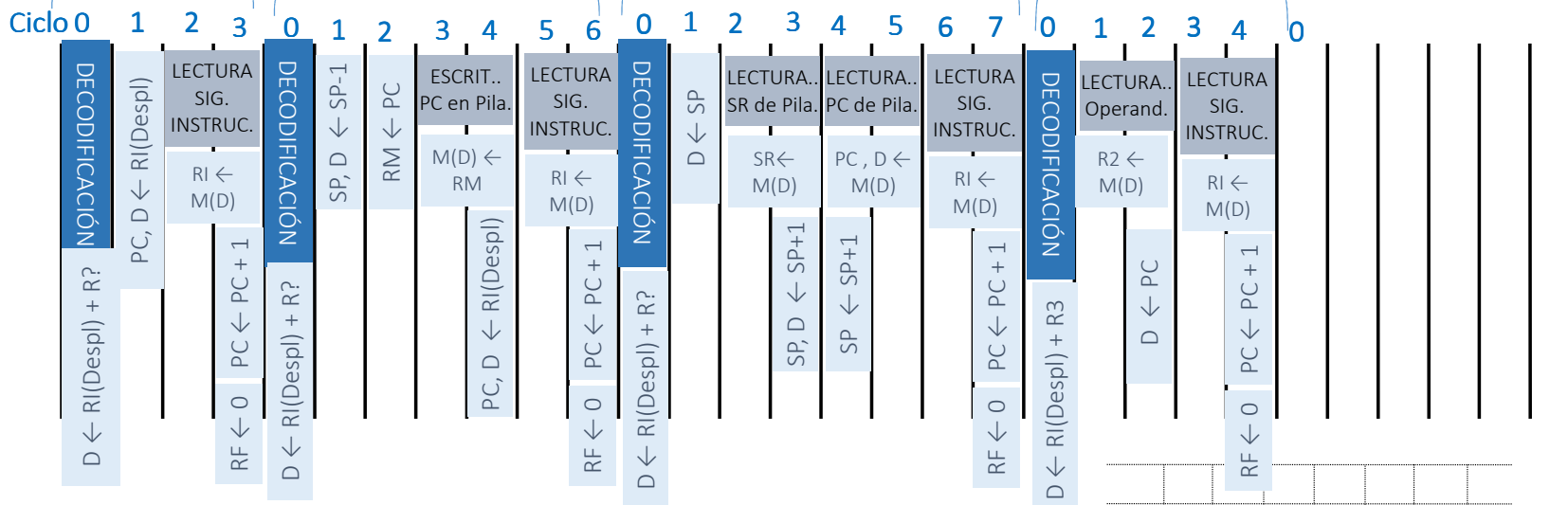
Nombre:

BNZ A0002H, Z = 0

CALL A0003H

RETI

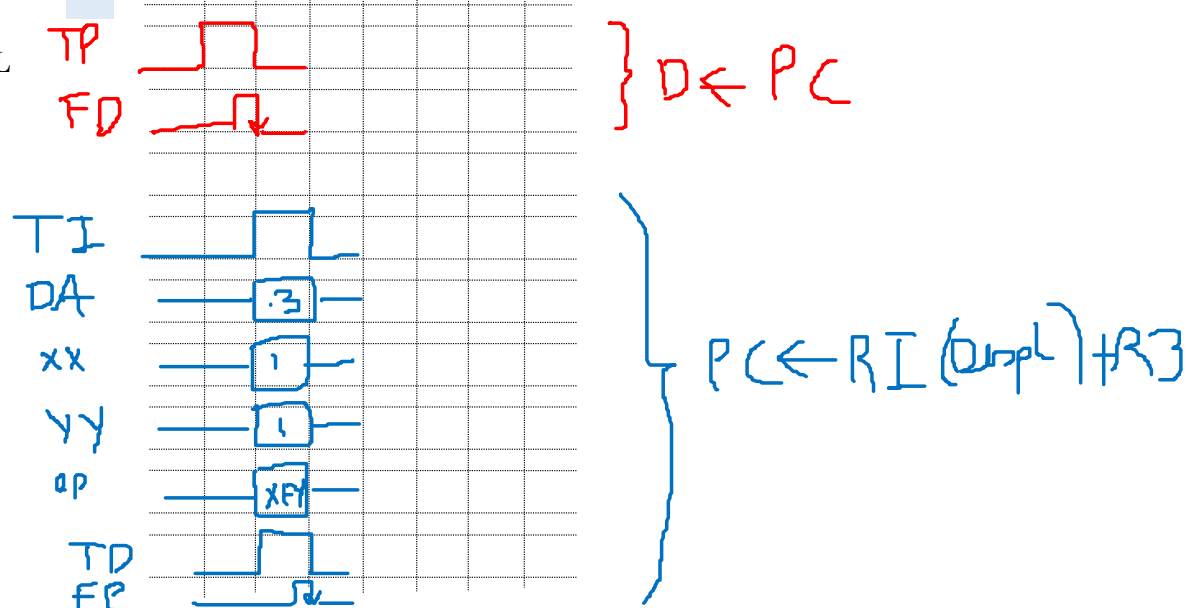
MOVE .2, 120(.3)



1) FORMATOS

BNZ				A0002H
MOVE		.2		.3 120
CALL				A0003H
RETI				

3) SEÑALES DE CONTROL



MP

BNZ A0002H
MOV .2, 120(.3)
CALL A0003H
RETI



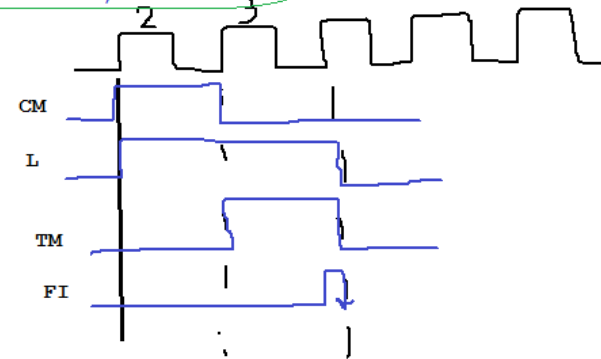
Direcc. A0000H

Direcc. A0001H --> PC

Direcc. A0002H --> PC $PC = D$

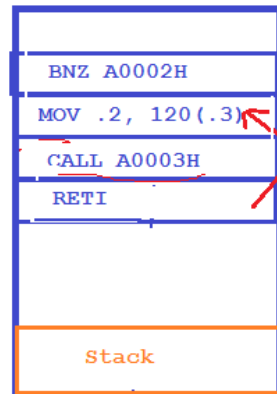
Direc. A0003H $PC = PC + 1$

Instrucción BNZ A0002H
Comprueba $z = 0$?
Si es cierto, $PC = A0002H$



Acceso de Lectura en Memoria

MP



Direcc. A0000H

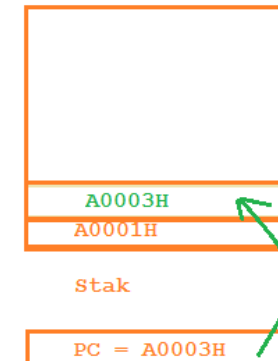
Direcc. A0001H

Direcc. A0002H

Direc. A0003H = PC

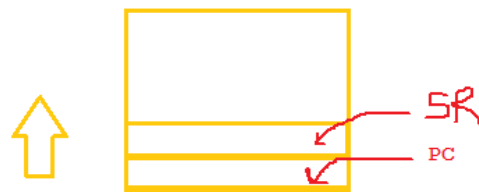
Instrucción CALL A0003H
Salvar PC en Pila
Ejecutamos el Salto:
PC <-- A0003H

PC, D <-- A0003H =
Dirección del salto

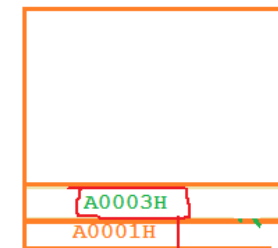


SP= SP -1 -->> SP = 49
-->> SP = 50

Instrucción RETI
Recuperar de la Pila el registro SR y PC

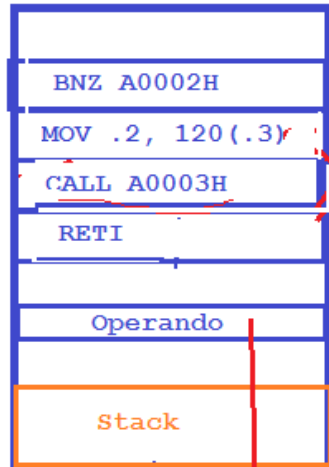


-->> SP = 49
50 -->> SP = SP+1



-->> SP = 49
50

MP



Direcc. A0000H

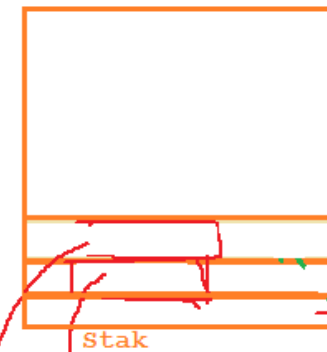
Direcc. A0001H

Direcc. A0002H PC --> D

Direc. A0003H PC = PC +1

Direcc. (120 + R3)

Instrucción MOVE .2, 120(.3)
R2 <<-- M(120 + R3)



49

50 --> SP :

-->> SP = SP+1

Stak

PC

SP