

ESTRUCTURA Y TECNOLOGIA DE COMPUTADORES II

TEMA 6: *Diseño del procesador*

**SOLUCION
PROBLEMAS PROPUESTOS EN EXAMEN**

**Curso 2007-2008
Jose Manuel Díaz Martínez
Tutor de la asignatura ETC-II**

CONTENIDO

SOLUCION PROBLEMA 6.1 3

SOLUCION PROBLEMA 6.2 4

SOLUCION PROBLEMA 6.3 6

SOLUCION PROBLEMA 6.4 6

SOLUCION PROBLEMA 6.5 8

SOLUCION PROBLEMA 6.6 9

SOLUCION PROBLEMA 6.7 9

SOLUCION PROBLEMA 6.8 10

SOLUCION PROBLEMA 6.9 11

SOLUCION PROBLEMA 6.10 11

SOLUCION PROBLEMA 6.11 13

SOLUCION PROBLEMA 6.12 14

SOLUCION PROBLEMA 6.1

Antes de comenzar a contestar a los diferentes apartados conviene darse cuenta de que la UAL es un circuito combinacional que no tiene ninguna capacidad de memoria, de manera que cuando las señales de control activan una función específica de la UAL, su salida pasa a ser el resultado de la operación correspondiente con sus entradas. Por este motivo, la salida de la UAL no puede conectarse directamente al bus, ya que esta salida se realimentaría a la entrada. En su lugar, el registro R2 permite el almacenamiento temporal de la salida.

Se utiliza en este problema la siguiente notación:

AC acumulador

RD registro de dirección de memoria que especifica la próxima dirección de memoria donde se va a leer o a escribir.

RM registro de datos de memoria contiene el dato a escribir en la memoria o recibe el dato leído en la memoria.

RI registro de instrucción, en el se almacena la instrucción que se va ejecutar .

RI(datos) campo de datos de la instrucción almacenada en el RI.

RI(dirección) campo de dirección de la instrucción almacenada en el RI.

(XX) Contenido del registro XX.

a) Al tratarse de un operando con direccionamiento inmediato significa que en uno de los campos de la instrucción se encuentra almacenado uno de los operandos. Las microoperaciones a realizar son:

$$R1 \leftarrow (RI(datos))$$

$$R2 \leftarrow (R1) + (AC)$$

$$AC \leftarrow (R2)$$

Es decir primero se carga en R1 el contenido del campo de datos de la instrucción almacenada en RI. A continuación se procede a sumar el contenido de R1 con el contenido del AC. Finalmente se carga en AC el contenido de R2.

b) Al tratarse de un operando con direccionamiento directo, significa que en el campo de dirección de la instrucción almacenada en RI se encuentra la dirección de memoria donde hay que ir a buscar el operando. Las microoperaciones a realizar son:

$$RD \leftarrow (RI(dirección))$$

$$RM \leftarrow \text{memoria}$$

$$R1 \leftarrow (RM)$$

$$R2 \leftarrow (R1) + (AC)$$

$$AC \leftarrow (R2)$$

Es decir primero se carga en RD (registro de direcciones de memoria) el contenido del campo de dirección de la instrucción almacenada en RI. A continuación se descarga en RM el contenido de la posición de memoria especificada en RD. Posteriormente se almacena en R1 el contenido de RM. A continuación se procede a sumar el contenido de R1 con el contenido del AC. Finalmente se carga en AC el contenido de R2.

c) Al tratarse de un operando con direccionamiento indirecto, significa que en el campo de dirección de la instrucción almacenada en RI se encuentra la dirección de memoria donde hay que ir a buscar la dirección de memoria donde se encuentra el operando. Las microoperaciones a realizar son:

$$RD \leftarrow (RI(\text{dirección}))$$

$$RM \leftarrow \text{memoria}$$

$$RD \leftarrow (RM)$$

$$RM \leftarrow \text{memoria}$$

$$R1 \leftarrow (RM)$$

$$R2 \leftarrow (R1) + (AC)$$

$$AC \leftarrow (R2)$$

Es decir primero se carga en RD (registro de direcciones de memoria) el contenido del campo de dirección de la instrucción almacenada en RI. A continuación se descarga en RM el contenido de la posición de memoria especificada en RD, que contiene la dirección de memoria donde hay que buscar el operando. A continuación se carga en RD la dirección de memoria almacenada en RD. El paso siguiente es descargar el operando de la memoria a RM. Posteriormente se almacena en R1 el contenido de RM. A continuación se procede a sumar el contenido de R1 con el contenido del AC. Finalmente se carga en AC el contenido de R2.

SOLUCION PROBLEMA 6.2

Se utiliza en este problema la siguiente notación:

AC acumulador

RD registro de dirección de memoria que especifica la próxima dirección de memoria donde se va a leer o a escribir.

RM registro de datos de memoria contiene el dato a escribir en la memoria o recibe el dato leído en la memoria.

RI registro de instrucción, en el se almacena la instrucción que se va ejecutar .

RI(datos) campo de datos de la instrucción almacenada en el RI.

RI(dirección) campo de dirección de la instrucción almacenada en el RI.

(XX) Contenido del registro XX.

a) Cargar acumulador con el contenido de una dirección de memoria. En la siguiente tabla se indican las microoperaciones necesarias, así como las señales de control que hay que activar para la realización de las mismas.

Microoperaciones	Señales de Control
$RD \leftarrow (RI(\text{dirección}))$	c_{14}
$RM \leftarrow \text{memoria}$	c_0
$AC \leftarrow (RM)$	c_9

b) Almacenar el contenido del acumulador en una dirección de memoria. En la siguiente tabla se indican las microoperaciones necesarias, así como las señales de control que hay que activar para la realización de las mismas.

Microoperaciones	Señales de Control
$RD \leftarrow (RI(\text{dirección}))$	c_{14}
$RM \leftarrow (AC)$	c_{10}
$\text{memoria} \leftarrow (RM)$	c_1

c) Sumar al acumulador un operando con direccionamiento inmediato. En la siguiente tabla se indican las microoperaciones necesarias, así como las señales de control que hay que activar para la realización de las mismas.

Microoperaciones	Señales de Control
$RM \leftarrow (RI(\text{dato}))$	c_{15}
$\text{Entrada } 1 \leftarrow (RM)$	c_8
$AC \leftarrow (AC) + \text{Entrada } 1$	c_{11}

d) Sumar al acumulador un operando con direccionamiento directo. En la siguiente tabla se indican las microoperaciones necesarias, así como las señales de control que hay que activar para la realización de las mismas.

Microoperaciones	Señales de Control
$RD \leftarrow (RI(\text{dirección}))$	c_{14}
$RM \leftarrow \text{memoria}$	c_0
$\text{Entrada } 1 \leftarrow (RM)$	c_8
$AC \leftarrow (AC) + \text{Entrada } 1$	c_{11}

e) Sumar al acumulador un operando con direccionamiento indirecto. En la siguiente tabla se indican las microoperaciones necesarias, así como las señales de control que hay que activar para la realización de las mismas.

Microoperaciones	Señales de Control
$RD \leftarrow (RI(\text{dirección}))$	c_{14}
$RM \leftarrow \text{memoria}$	c_0
$RD \leftarrow (RM)$	c_2
$RM \leftarrow \text{memoria}$	c_0
$\text{Entrada } 1 \leftarrow (RM)$	c_8
$AC \leftarrow (AC) + \text{Entrada } 1$	c_{11}

f) AND con el acumulador y un operando en modo directo. En la siguiente tabla se indican las microoperaciones necesarias, así como las señales de control que hay que activar para la realización de las mismas.

Microoperaciones	Señales de Control
$RD \leftarrow (RI(\text{dirección}))$	c_{14}
$RM \leftarrow \text{memoria}$	c_0
$\text{Entrada } 1 \leftarrow (RM)$	c_8
$AC \leftarrow (AC) \text{ AND } \text{Entrada } 1$	c_{12}

g) Bifurcar, en modo de direccionamiento directo. En la siguiente tabla se indican las microoperaciones necesarias, así como las señales de control que hay que activar para la realización de las mismas.

Microoperaciones	Señales de Control
$RD \leftarrow (RI(\text{dirección}))$	c_{14}
$RM \leftarrow \text{memoria}$	c_0
$CP \leftarrow (RM)$	c_4

h) Bifurcar si acumulador=0, en modo de direccionamiento directo. En la siguiente tabla se indican las microoperaciones necesarias, así como las señales de control que hay que activar para la realización de las mismas.

Microoperaciones	Señales de Control
$RD \leftarrow (RI(\text{dirección}))$	c_{14}
$RM \leftarrow \text{memoria}$	c_0
Si $s_0=1$ $CP \leftarrow (RM)$	c_4

SOLUCION PROBLEMA 6.3

De acuerdo con lo explicado en el libro de Teoría

Afirmación I : Es **Verdadera**

Afirmación II : Es **Falsa** ya que cada fase del ciclo de instrucción se puede descomponer en una secuencia de operaciones elementales llamadas microoperaciones.

SOLUCION PROBLEMA 6.4

- Procesador con cero instrucciones.
- Secuencia de instrucciones para calcular la expresión $X=Y^2 \cdot (X+Z)$

El funcionamiento de un procesador con cero instrucciones se explica en la sección 6.1.4 del libro de teoría. Un problema relacionado con éste es el 6.4 del libro de problemas.

Para resolver este ejercicio se debe proceder a analizar cada una de las posibles respuestas. Se representa en una tabla las instrucciones y el contenido que tendría la pila tras ejecutar cada instrucción. Según se sitúa un dato más a la izquierda en la pila, significa que lleva más tiempo almacenado, es decir, los datos que entran en la pila lo hacen por su derecha. En consecuencia el elemento almacenado más a la derecha en la pila constituye la cima de la pila

La expresión que se pide calcular también se puede expresar equivalentemente de la siguientes formas :

$$X=(X+Z) \cdot Y \cdot Y = Y \cdot Y \cdot (X+Z)$$

Opción A) :

Instrucciones	Contenido de la pila		
Push [X] ;	Y		
Push [Z] ;	Y	Z	
Add ;	Y+Z		
Push [Y] ;	Y+Z	Y	
Push [Y] ;	Y+Z	Y	Y
Mult ;	Y+Z	$Y \times Y$	
Mult ;	$(Y+Z) \times Y \times Y$		
Pop [X] ;			

Luego esta secuencia de instrucciones calcula la expresión pedida.

Opción B) :

Instrucciones	Contenido de la pila			
Push[Y];	Y			
Push[Y];	Y	Y		
Push[X];	Y	Y	X	
Push[Z];	Y	Y	X	Z
Add;	Y	Y	X+Z	
Mult;	Y	$Y \times (X+Z)$		
Mult;	$Y \times Y \times (X+Z)$			
Pop[X];				

Luego esta secuencia de instrucciones calcula la expresión pedida.

Opción C) :

Instrucciones	Contenido de la pila		
Push[Y];	Y		
Push[Y];	Y	Y	
Mult;	$Y \times Y$		
Push[X];	$Y \times Y$	X	
Push[Z];	$Y \times Y$	X	Z
Add;	$Y \times Y$	$X+Z$	
Mult;	$(Y \times Y) \times (X+Z)$		
Pop[X];			

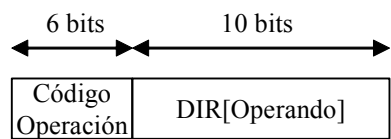
Luego esta secuencia de instrucciones calcula la expresión pedida.

Luego la respuesta correcta es la **D**.

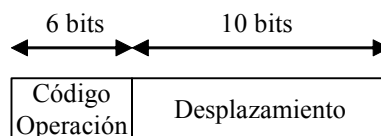
SOLUCION PROBLEMA 6.5

- Procesador de una dirección.
- 6 bits para el código de operación.
- 10 bits para la dirección de un operando.
- Instrucción de bifurcación, con direccionamiento relativo al contador del programa, almacenada en la dirección $DIR_{10}=530$ origina un salto a la posición $DIR_{10}=620$.
- El código de operación de una cierta instrucción es 110011
- Codificación en binario de la instrucción?

De acuerdo con el enunciado se tiene el siguiente formato de instrucción:



Una instrucción de bifurcación con direccionamiento relativo al contador de programa CP, se puede expresar de la siguiente forma:



Donde la dirección del operando se obtiene a partir de:

$$DIR_{final} = [CP] + Desplazamiento$$

Sustituyendo los datos del enunciado se tiene:

$$620 = 530 + Desplazamiento$$

Luego:

$$Desplazamiento = 90$$

Si se pasa a binario el Desplazamiento se tiene:

$$Desplazamiento_{10} = 90 = 64 + 16 + 8 + 2 = 2^6 + 2^4 + 2^3 + 2$$

$$Desplazamiento_2 = 1011010$$

En el enunciado nos dan el código de operación de la instrucción 110011 y acabamos de calcular el desplazamiento, que expresado sobre un tamaño de 10 bits toma la forma: 0001011010. Luego la instrucción pedida es:

110011 0001011010

SOLUCION PROBLEMA 6.6

- Procesador con cero instrucciones.
- Push M[C]; Push M[D]; Add; Push M[C]; Push M[D]; Add; Mult; Pop M[A]

El funcionamiento de un procesador con cero instrucciones se explica en la sección 6.1.4 del libro de teoría. Un problema relacionado con éste es el 6.4 del libro de problemas.

Se representa en una tabla las instrucciones y el contenido que tendría la pila tras ejecutar cada instrucción. Según se sitúa un dato más a la izquierda en la pila, significa que lleva más tiempo almacenado, es decir, los datos que entran en la pila lo hacen por su derecha. En consecuencia el elemento almacenado más a la derecha en la pila constituye la cima de la pila

Instrucciones	Contenido de la pila		
Push M[C];	C		
Push M[D];	C	D	
Add;	C+D		
Push M[C];	C+D	C	
Push M[D];	C+D	C	D
Add;	C+D	C+D	
Mult;	$(C+D) \cdot (C+D)$		
Pop M[A];			

Luego esta secuencia de instrucciones calcula la expresión $A=(C+D)^2$.

SOLUCION PROBLEMA 6.7

- Procesador de una dirección (procesador con acumulador) con un registro R1
- Operaciones a realizar: Load A; Add B; Add C; Mult D; Add A; Store R1; Mult R1; Div A; Store C

En la siguiente tabla se muestran las instrucciones a ejecutar, su significado, el contenido del acumulador y el contenido del registro. Se va a utilizar la siguiente notación: Ac representa al acumulador, M[X] representa a la posición de memoria X y R1 representa al registro.

	Instrucción	Significado	Acumulador (Ac)	Registro (R1)
1	Load A	$Ac \leftarrow M[A]$	A	--
2	Add B	$Ac \leftarrow Ac + M[B]$	A+B	--
3	Add C	$Ac \leftarrow Ac + M[C]$	A+B+C	--
4	Mult D	$Ac \leftarrow Ac \times M[D]$	$(A+B+C) \cdot D$	--
5	Add A	$Ac \leftarrow Ac + M[A]$	$((A+B+C) \cdot D) + A$	--
6	Store R1	$R1 \leftarrow Ac$	$((A+B+C) \cdot D) + A$	$[(A+B+C) \cdot D] + A$
7	Mult R1	$Ac \leftarrow Ac \times R1$	$((A+B+C) \cdot D + A)^2$	$[(A+B+C) \cdot D] + A$
8	Div A	$Ac \leftarrow Ac / M[A]$	$((A+B+C) \cdot D + A)^2 / A$	$[(A+B+C) \cdot D] + A$
9	Store C	$M[C] \leftarrow Ac$	$((A+B+C) \cdot D + A)^2 / A$	$[(A+B+C) \cdot D] + A$

Al finalizar la operación 9 la posición de memoria C tendrá almacenado el valor $((A+B+C) \cdot D + A)^2 / A$. Luego la operación que calcula esta secuencia de 9 operaciones es:

$$C = ((A+B+C) \cdot D + A)^2 / A$$

SOLUCION PROBLEMA 6.8

- Procesador con instrucciones de cero direcciones
- Cada instrucción ocupa una palabra de memoria.
- La pila es cableada.
- Operación a realizar $Y=Y \cdot (X+Z)$

De acuerdo con el funcionamiento de este tipo de procesadores una posible secuencia de instrucciones a ejecutar para implementar la operación $Y=Y \cdot (X+Z)$ sería:

- 1) Push M[Z];
- 2) Push M[X];
- 3) Add;
- 4) Push M[Y];
- 5) Mult;
- 6) Pop M[Y]

El número de accesos a memoria N_{AM} para ejecutar estas seis instrucciones será la suma de dos componente:

$$N_{AM}=N_{FB}+N_{FE}$$

donde N_{FE} son los accesos a memoria durante la fase de búsqueda de estas instrucciones y N_{FB} son los accesos a memoria durante la fase de ejecución.

En este caso $N_{FB}=6$ ya que hay que 6 instrucciones en este programa. Se supone que cada instrucción ocupa una palabra (como así lo indica el enunciado) y que en un acceso a memoria se lee o se escribe una palabra.

Por otra parte, el enunciado indica que la pila es cableada lo que significa que no se implementa en la memoria. En consecuencia los únicos tipos de instrucciones que requieren acceder a memoria durante su ejecución son: Push y Pop. Las instrucciones tipo Push requieren de un acceso a memoria para leer el operando de memoria que se carga en la pila. Lo mismo sucede con las instrucciones de tipo Pop pero en este caso para escribir en memoria el valor almacenado en la cumbre de la pila. En conclusión puesto que el programa consta de 3 instrucciones tipo Push y una instrucción tipo Pop, $N_{FE}=4$.

Luego, el número total de accesos a memoria es $N_{AM}=6+4=10$ **accesos**.

Se puede comprobar que se obtendría el mismo resultado si se considerara la secuencia de instrucciones

- 1) Push M[Y];
- 2) Push M[X];
- 3) Push M[Z];
- 4) Add;
- 5) Mult;
- 6) Pop M[Y]

SOLUCION PROBLEMA 6.9

- Procesador de una dirección (procesador con acumulador) con un registro R1
- Operaciones a realizar: Load A; Add B; Mult D, Add A, Store R1, Mult R1, Mult R1; Mult A; Div B; Add A; Store C

En la siguiente tabla se muestran las instrucciones a ejecutar, su significado, el contenido del acumulador y el contenido del registro. Se va a utilizar la siguiente notación: Ac representa al acumulador, M[X] representa a la posición de memoria X y R1 representa al registro.

	Instrucción	Significado	Acumulador (Ac)	Registro (R1)
1	Load A	$Ac \leftarrow M[A]$	A	--
2	Add B	$Ac \leftarrow Ac + M[B]$	A+B	--
3	Mult D	$Ac \leftarrow Ac \times M[D]$	$(A+B) \cdot D$	--
4	Add A	$Ac \leftarrow Ac + M[A]$	$(A+B) \cdot D + A$	--
5	Store R1	$R1 \leftarrow Ac$	$(A+B) \cdot D + A$	$[(A+B) \cdot D] + A$
6	Mult R1	$Ac \leftarrow Ac \times R1$	$((A+B) \cdot D + A)^2$	$[(A+B) \cdot D] + A$
7	Mult R1	$Ac \leftarrow Ac \times R1$	$((A+B) \cdot D + A)^3$	$[(A+B) \cdot D] + A$
8	Mult A	$Ac \leftarrow Ac \times M[A]$	$((A+B) \cdot D + A)^3 \cdot A$	$[(A+B) \cdot D] + A$
9	Div B	$Ac \leftarrow Ac / M[B]$	$((A+B) \cdot D + A)^3 \cdot A / B$	$[(A+B) \cdot D] + A$
10	Add A	$Ac \leftarrow Ac + M[A]$	$((A+B) \cdot D + A)^3 \cdot A / B + A$	$[(A+B) \cdot D] + A$
11	Store C	$M[C] \leftarrow Ac$	$((A+B) \cdot D + A)^3 \cdot A / B + A$	$[(A+B) \cdot D] + A$

Al finalizar la operación 11 la posición de memoria C tendrá almacenado el valor $(((A+B) \cdot D + A)^3 \cdot A) / B + A$. Luego la operación que calcula esta secuencia de 11 operaciones es:

$$C = (((A+B) \cdot D + A)^3 \cdot A) / B + A$$

SOLUCION PROBLEMA 6.10

- Procesador con 32 registros.
- Operando inmediato de 16 bits.
- 142 instrucciones en su repertorio.
- En un determinado programa:
 - el 20% de las instrucciones tienen un registro de entrada y un registro de salida.
 - el 30% tienen dos registros de entrada y uno de salida.
 - el 25% tiene un registro de entrada, un registro de salida y un operando inmediato
 - el 25% restante tiene un operando inmediato y un registro de salida

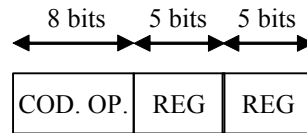
a) Puesto que se tiene un repertorio de 142 instrucciones, como $142 \leq 2^n \Rightarrow n=8$, se necesitan 8 bits para codificarlas dentro del código de operación de la instrucción [COD. OP]

Por otra parte el procesador dispone de 32 registros, como $32 \leq 2^n \Rightarrow n=5$, se necesitan 5 bits para codificar el identificador del registro [REG].

Asimismo, del enunciado se sabe que el campo operando inmediato [OP] tiene un tamaño de 16 bits

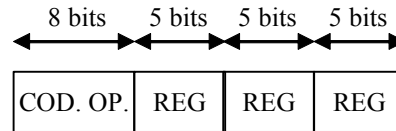
Con estos datos solamente se necesita sumar el tamaño de cada campo que compone un determinado tipo de instrucción y redondearlo a un múltiplo de 8:

- *Tipo 1*: Instrucciones con un registro de entrada y un registro de salida:



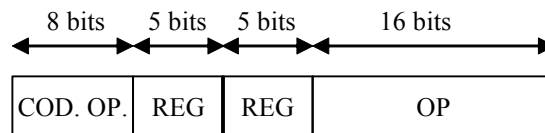
Se necesitan $L_1 = 8+5+5=18 \leq 24$ bits

- *Tipo 2*: Instrucciones con dos registros de entrada y uno de salida:



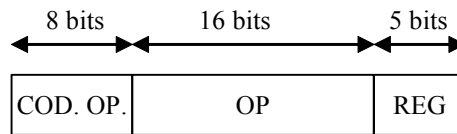
Se necesitan $L_2 = 8+5+5+5=23 \leq 24$ bits

- *Tipo 3*: Instrucciones con un registro de entrada, un registro de salida y un operando inmediato:



Se necesitan $L_3 = 8+5+5+16=34 \leq 40$ bits

- *Tipo 4*: Instrucciones con un operando inmediato y un registro de salida:



Se necesitan $L_4 = 8+16+5=29 \leq 32$ bits

b) En primer lugar se va a calcular la memoria que ocupa el programa si se utilizase un repertorio de instrucciones de longitud fija (LF). La longitud que se debe considerar para dicho repertorio se corresponde con la mayor de las longitudes de los diferentes tipos de instrucciones. En este caso las instrucciones de mayor longitud son las del tipo 3, con 40 bits. Luego la memoria que ocupa el programa considerando que todas sus instrucciones ocupan 40 bits es:

$$C_P^{LF} = 40 \cdot N_T \text{ bits}$$

donde N_T es el número de instrucciones de que consta el programa. Es obvio que en promedio el número de bits que ocupa una instrucción de un repertorio de LF de 40 bits es de 40 bits/instrucción. Formalmente se calcularía de la siguiente forma:

$$\frac{C_P^{LF}}{N_T} = 40 \text{ (bits/instrucción)}$$

Se va a considerar ahora que el repertorio de instrucciones es de longitud variable (LV), en dicho caso la memoria que ocupa el programa es:

$$C_P^{LV} = 24 \cdot N_1 + 24 \cdot N_2 + 40 \cdot N_3 + 32 \cdot N_4 \text{ bits}$$

donde N_i $i=1,2,3,4$ es el número de instrucciones del tipo i que ocupa el programa. En promedio el número de bits que ocupa una instrucción de un repertorio de LV es:

$$\frac{C_P^{LV}}{N_T} = 24 \cdot \frac{N_1}{N_T} + 24 \cdot \frac{N_2}{N_T} + 40 \cdot \frac{N_3}{N_T} + 32 \cdot \frac{N_4}{N_T} = 24 \cdot 0.2 + 24 \cdot 0.3 + 40 \cdot 0.25 + 32 \cdot 0.25 = 30 \text{ (bits/instrucción)}$$

Luego en promedio usando un repertorio de LV se ahorran con respecto al repertorio de LF

$$\frac{C_P^{LF}}{N_T} - \frac{C_P^{LV}}{N_T} = 40 - 30 = 10 \text{ (bits/instrucción)}$$

O en porcentaje, el ahorro de memoria sería del

$$\frac{\frac{C_P^{LF}}{N_T} - \frac{C_P^{LV}}{N_T}}{\frac{C_P^{LF}}{N_T}} = \frac{10}{40} \cdot 100 = 25\%$$

SOLUCION PROBLEMA 6.11

- Procesador con instrucciones de cero direcciones
- Operación a realizar $Y=(X+Y)+Z$

De acuerdo con el funcionamiento de este tipo de procesadores una posible secuencia de instrucciones a ejecutar para implementar la operación $Y=(X+Y)+Z$ sería:

- 1) Push M[X];
- 2) Push M[Y];
- 3) Add;
- 4) Push M[Z];
- 5) Add;
- 6) Pop M[Y];

El número de accesos a memoria N_{AM} para ejecutar estas seis instrucciones será la suma de dos componente:

$$N_{AM} = N_{FB} + N_{FE}$$

donde N_{FE} son los accesos a memoria durante la fase de búsqueda de estas instrucciones y N_{FE} son los accesos a memoria durante la fase de ejecución.

En este caso $N_{FB} = 6$ ya que hay que 6 instrucciones en este programa. Se supone que cada instrucción ocupa una palabra y que en un acceso a memoria se lee o se escribe una palabra.

Por otra parte, los únicos tipos de instrucciones que requieren acceder a memoria durante su ejecución son: Push y Pop. Las instrucciones tipo Push requieren de un acceso a memoria para leer el operando de memoria que se carga en la pila. Lo mismo sucede con las instrucciones de tipo Pop pero en este caso para escribir en memoria el valor almacenado en la cumbre de la pila. En conclusión puesto que el programa consta de 3 instrucciones tipo Push y una instrucción tipo Pop, $N_{FE}=4$.

Luego, el número total de accesos a memoria es $N_{AM}=6+4=10$ **accesos**.

SOLUCION PROBLEMA 6.12

- Procesador de una dirección (procesador con acumulador) con un registro R1
- Operaciones a realizar: Load X; Add Y; Add Z; Mult X, Store R1, Mult R1, Div X, Store X

En la siguiente tabla se muestran las instrucciones a ejecutar, su significado, el contenido del acumulador y el contenido del registro. Se va a utilizar la siguiente notación: Ac representa al acumulador, M[X] representa a la posición de memoria X y R1 representa al registro.

	Instrucción	Significado	Acumulador (Ac)	Registro (R1)
1	Load X	$Ac \leftarrow M[X]$	X	--
2	Add Y	$Ac \leftarrow Ac + M[Y]$	X+Y	--
3	Add Z	$Ac \leftarrow Ac + M[Z]$	X+Y+Z	--
4	Mult X	$Ac \leftarrow Ac \cdot M[X]$	$(X+Y+Z) \cdot X$	--
5	Store R1	$R1 \leftarrow Ac$	$(X+Y+Z) \cdot X$	$(X+Y+Z) \cdot X$
6	Mult R1	$Ac \leftarrow Ac \cdot R1$	$[(X+Y+Z) \cdot X]^2$	$(X+Y+Z) \cdot X$
7	Div X	$Ac \leftarrow Ac / X$	$(X+Y+Z)^2 \cdot X$	$(X+Y+Z) \cdot X$
8	Store X	$M[X] \leftarrow Ac$	$(X+Y+Z)^2 \cdot X$	$(X+Y+Z) \cdot X$

Al finalizar la operación 8 la posición de memoria X tendrá almacenado el valor $(X+Y+Z)^2 \cdot X$. Luego la operación que calcula esta secuencia de 8 operaciones es:

$$X = (X+Y+Z)^2 \cdot X$$