

1. Considérese la estructura de la Computadora Mejorada a la que se le han aplicado los siguientes cambios:

- La memoria tiene un ancho de palabra de 8 bits en lugar de 12.
- El registro GPR es de 8 bits.
- El registro OPR es de 8 bits.
- El sumador y el registro acumulador son de 8 bits.
- Las instrucciones tienen un código de operación de 8 bits, por lo que siempre ocupan al menos una posición de memoria. Pueden ocupar dos si tienen operando.

Teniendo en cuenta estos cambios, obtener

- (a) Ciclo de búsqueda.
- (b) Ciclo de ejecución de la instrucción ADD n, teniendo en cuenta que es una instrucción que ocupa dos posiciones de memoria, siendo la primera el código de operación y la segunda un valor inmediato que hay que sumar al acumulador.

(3 puntos)

2. Tomando como punto de partida la estructura de la Computadora Mejorada y completándola con las microoperaciones que sean necesarias, implementar mediante control cableado el Algoritmo de Booth. Incluir también el ciclo de búsqueda. Obtener la función de control para las microoperaciones  $AC \leftarrow AC + BR$ ,  $AC \leftarrow AC + \overline{BR} + 1$ ,  $ashr(AC \& QR)$ .

(4 puntos)

3. En un sistema de memoria virtual paginado el espacio de direcciones es de 2 MB, mientras que el espacio de memoria es de 64 KB. El tamaño de página es de 1KB. Determinar:

- (a) Longitud en bits de una dirección virtual y de una dirección física.
- (b) Número de páginas y bloques del sistema de memoria.
- (c) Dimensiones de las tablas de páginas, según se implementen mediante una memoria de acceso aleatorio o a través de una memoria asociativa.
- (d) Un programa tiene una extensión de 12 páginas consecutivas en memoria virtual. Su primera página comienza en la dirección virtual 002C00 (en hexadecimal). El contenido de la tabla de páginas del programa es el siguiente:

Página	Bloque
0	30
1	24
3	2

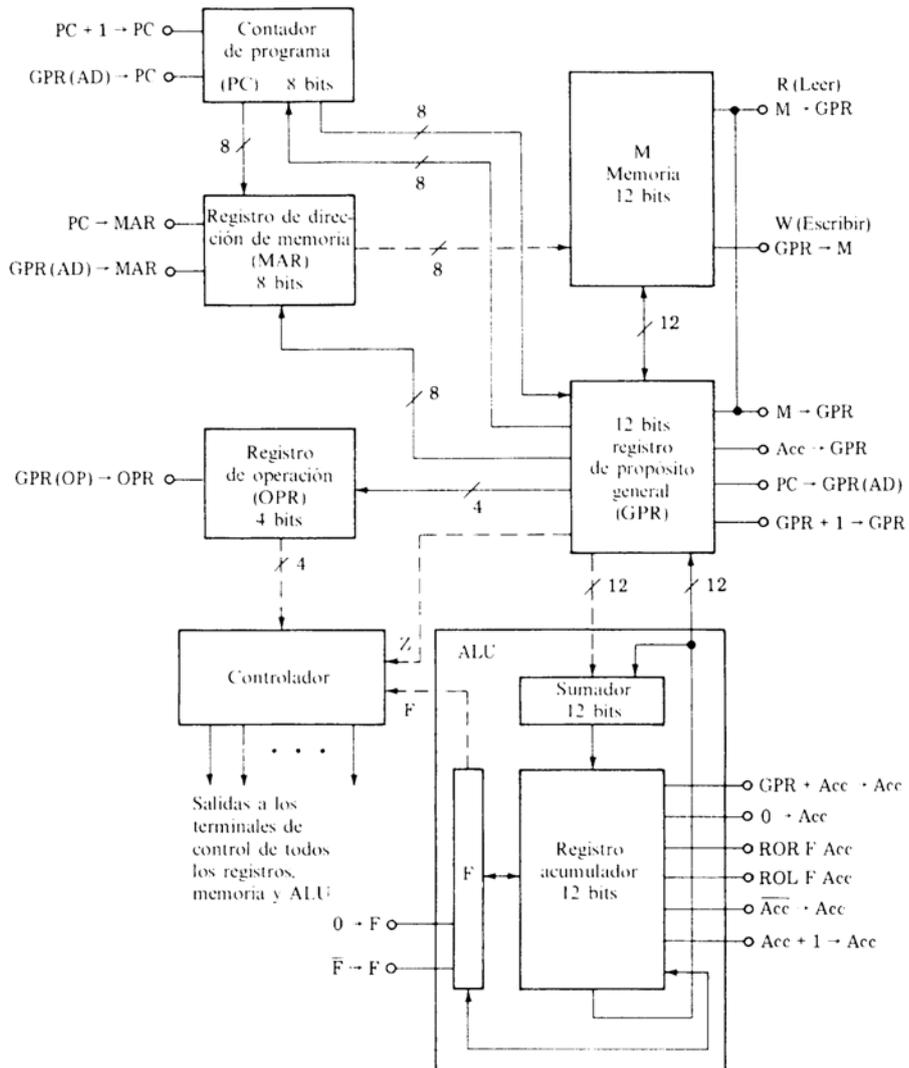
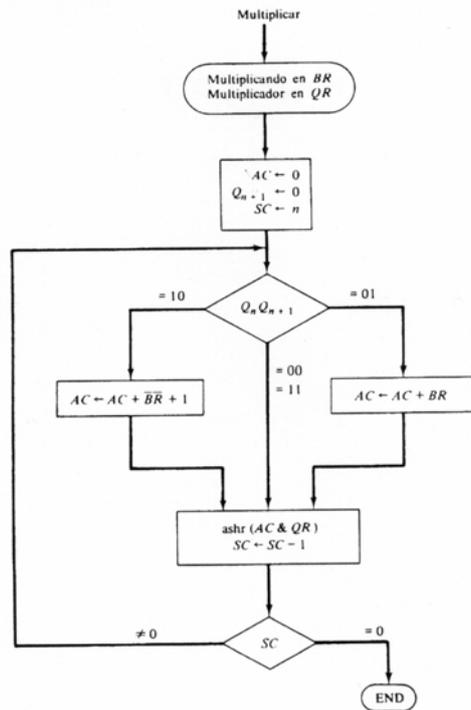
Página	Bloque
4	15
6	56
9	39

Página	Bloque
10	25
11	7

Realizar la traducción de las siguientes direcciones virtuales a direcciones físicas:

- 4.1. 002C35
- 4.2. 0034D1
- 4.3. 012127
- 4.4. 005959

(3 puntos)



**Ejercicio 1.**

- |  |  |
|--|--|
| <p>(a) <b>Ciclo de búsqueda</b></p> <ol style="list-style-type: none"> <li>1. MAR ← PC</li> <li>2. GPR ← M, PC ← PC + 1</li> <li>3. OPR ← GPR</li> </ol> | <p>(b) <b>Ciclo de ejecución de ADD n</b></p> <ol style="list-style-type: none"> <li>1. MAR ← PC</li> <li>2. GPR ← M, PC ← PC + 1</li> <li>3. Acc ← GPR + Acc</li> </ol> |
|--|--|

En el ciclo de ejecución de ADD se debe contemplar que, una vez leído de memoria el operando de la instrucción (n), el valor del registro PC debe apuntar al código de operación de la siguiente instrucción. De esta forma se consigue que la ejecución del ciclo de búsqueda de la siguiente instrucción se realice de forma correcta. Esto se consigue incrementando el valor del PC una vez leído el operando.

**Ejercicio 2.**

	Condición	Microoperación	Siguiente
<b>Ciclo de búsqueda</b>	$t_0$	MAR ← PC	SR ← SR + 1
	$t_1$	GPR ← M, PC ← PC + 1	SR ← SR + 1
	$t_2$	OPR ← GPR(OP)	SR ← SR + 1
<b>Ciclo de ejecución</b>	$i_1 \cdot t_3$	MAR ← GPR(AD)	SR ← SR + 1
	$i_1 \cdot t_4$	GPR ← M	SR ← SR + 1
	$i_1 \cdot t_5$	AC ← 0, $Q_{n+1} \leftarrow 0$ , SC ← n	SR ← SR + 1
	$i_1 \cdot t_6$	SC ← SC - 1	SR ← SR + 1
	$q_1 \cdot t_6 \cdot \overline{Q_n} \cdot Q_{n+1}$	AC ← AC + BR	—
	$q_1 \cdot t_6 \cdot Q_n \cdot \overline{Q_{n+1}}$	AC ← AC + $\overline{BR} + 1$	—
	$i_1 \cdot t_7$	ashr (AC & QR)	—
	$i_1 \cdot t_7 \cdot Z_{SC}$	—	SR ← 0
	$i_1 \cdot t_7 \cdot \overline{Z_{SC}}$	—	SR ← 6

Funciones de control:

- $AC \leftarrow AC + BR$ :  $q_1 \cdot t_6 \cdot \overline{Q_n} \cdot Q_{n+1}$
- $AC \leftarrow AC + \overline{BR} + 1$ :  $q_1 \cdot t_6 \cdot Q_n \cdot \overline{Q_{n+1}}$
- $ashr(AC \ \& \ QR)$ :  $i_1 \cdot t_7$

**Ejercicio 3.**

(a) El sistema permite direccional 2M palabras =  $2 \cdot 2^{20} = 2^{21}$  palabras en memoria virtual. Serán necesarias 21 líneas de dirección, por lo que la longitud de una dirección virtual es de 21 bits.

En memoria física, se puede direccionar 64 K palabras =  $2^6 \cdot 2^{10} = 2^{16}$  palabras. Serán necesarias 16 líneas de dirección, por lo que la longitud de una dirección física es de 16 bits.

(b) Ya que hay  $2^{21}$  palabras en memoria principal, y teniendo en cuenta que el tamaño de página es de 1K ( $2^{10}$ ) palabras, el número de páginas en memoria virtual será de  $2^{21}/2^{10} = 2^{11}$  páginas.

Como en memoria física hay  $2^{16}$  palabras, y el tamaño de bloque es igual al de página ( $2^{10}$ ) palabras, el número de bloques en memoria física será de  $2^{16}/2^{10} = 2^6$  bloques.

- (c) La implementación de la tabla de páginas utilizando memoria de acceso aleatorio requerirá  $2^{11}$  entradas (una para cada página del sistema). Cada una de ellas debe tener 1 bit de validez más 6 bits para indicar el bloque correspondiente. Por lo tanto, el tamaño total de la tabla de páginas será de  $2^{11} \cdot (1+6) = 14336$  bits.

Para implementar la tabla de páginas utilizando memoria asociativa serán necesarias  $2^6$  entradas (una para cada bloque de la memoria principal). Cada una de ellas debe tener 1 bit de validez, 11 bits para indicar la página, y 6 bits más para indicar el bloque. Por lo tanto, el tamaño total de la tabla de páginas será de  $2^6 \cdot (1+11+6) = 1152$  bits.

- (d) Hay que tener en cuenta que el programa comienza en la dirección 002C00, por lo que a todas las direcciones que se facilitan se les debe restar ésta, con objeto a obtener una dirección relativa al comienzo del programa, y no una dirección absoluta en memoria virtual. Esto es necesario ya que la tabla de páginas que se adjunta es propia del programa y no de todo el sistema.

(d.1) 002C35 → Dirección relativa al programa:  $002C35 - 002C00 = 000035$ .

000035 → 0 0000 0000 0000 0011 0101

Página: 0000000000 → 0. Según la tabla de páginas se corresponde con el bloque 30 → 011110. Por tanto, la dirección física es:

0111 1000 0011 0101 → 7835.

(d.2) 0034D1 → Dirección relativa al programa:  $0034D1 - 002C00 = 0008D1$ .

0008D1 → 0 0000 0000 1000 1101 0001

Página: 0000000010 → 2. Según la tabla de páginas, esta página no se encuentra en memoria principal, por lo que no se puede traducir la dirección. Se producirá un fallo de página y sólo cuando la unidad de memoria haya colocado la página en memoria principal y haya actualizado la tabla de páginas se podrá realizar la traducción de la dirección.

(d.3) 012127 → Dirección relativa al programa:  $012127 - 002C00 = 00F527$ .

00F527 → 0 0000 1111 0101 0010 0111

Página: 00000111101 → 61. Teniendo en cuenta que el programa sólo tiene 12 páginas, esta dirección no pertenece al programa y por lo tanto no se puede traducir.

(d.4) 005959 → Dirección relativa al programa:  $005959 - 002C00 = 002D59$ .

002D59 → 0 0000 0010 1101 0101 1001

Página: 00000001011 → 11. Según la tabla de páginas se corresponde con el bloque 7 → 000111. Por tanto, la dirección física es:

0001 1101 0101 1001 → 1D59.