1. Determinar la secuencia de microoperaciones necesarias para implementar las siguientes instrucciones en la Computadora Mejorada:

   (a) JPC m: Salta a la dirección de memoria m − PC. Implementarla sin utilizar la microoperación GPR ← M.

   (b) LDR m: Carga en el acumulador el contenido de la posición de memoria m + Acc, donde Acc es el valor que tiene el registro acumulador antes de ejecutarse la instrucción.

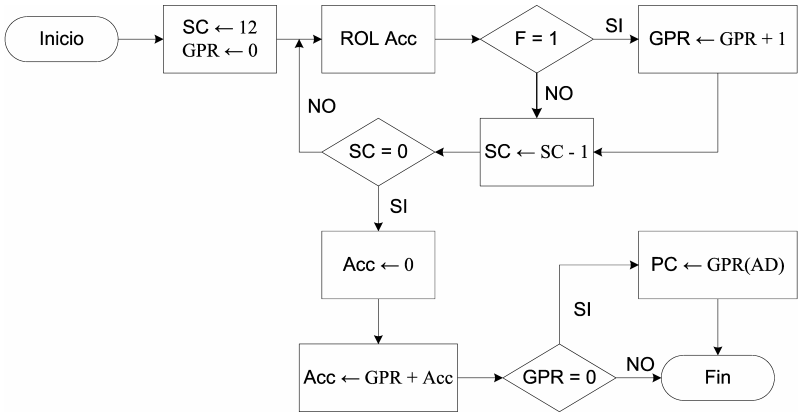
   (1 punto)

2. Tomando como punto de partida la estructura de la Computadora Mejorada, la cual se adjunta al final del examen, se le incorpora la microoperación GPR ← 0. También se la dota de un registro contador de iteraciones (SC), controlado por las microoperaciones SC ← 12 y SC ← SC – 1, que permiten inicializarlo y decrementarlo respectivamente.

   Como bits de status en esta computadora se encuentran los siguientes:

   - F: Valor del registro F, de un bit, situado en la unidad aritmético-lógica.
   - Z_{GPR}: Bit de cero del registro GPR.
   - Z_{SC}: Bit de cero del registro SC.

   Se pide implementar la unidad de control microprogramada (tanto la tabla de la CROM como la de la LCB) de la instrucción CUA. Dicha instrucción reemplazará el contenido del acumulador por un número que indica la cantidad de unos que contenía (el acumulador) antes de ejecutarse la instrucción. En caso de que el acumulador tuviera cero unos (todos sus bits fueran cero), debe realizar además un salto a la posición de memoria 0. El diagrama de flujo de esta instrucción se muestra a continuación:



   (1.5 puntos)

3. Se dispone de un sistema de memoria de 64K × 8. La unidad mínima direccionable es el byte. En memoria se encuentran dos arrays de elementos de 8 bits, almacenados en las direcciones representadas por el esquema siguiente:

Array A. Dirección de comienzo: 18F0												
Contenido	12	4	67	81	231	89	106	198	206	11	39	162

Array B. Dirección de comienzo: B2D0												
Contenido	102	44	54	241	163	25	175	102	39	89	156	250

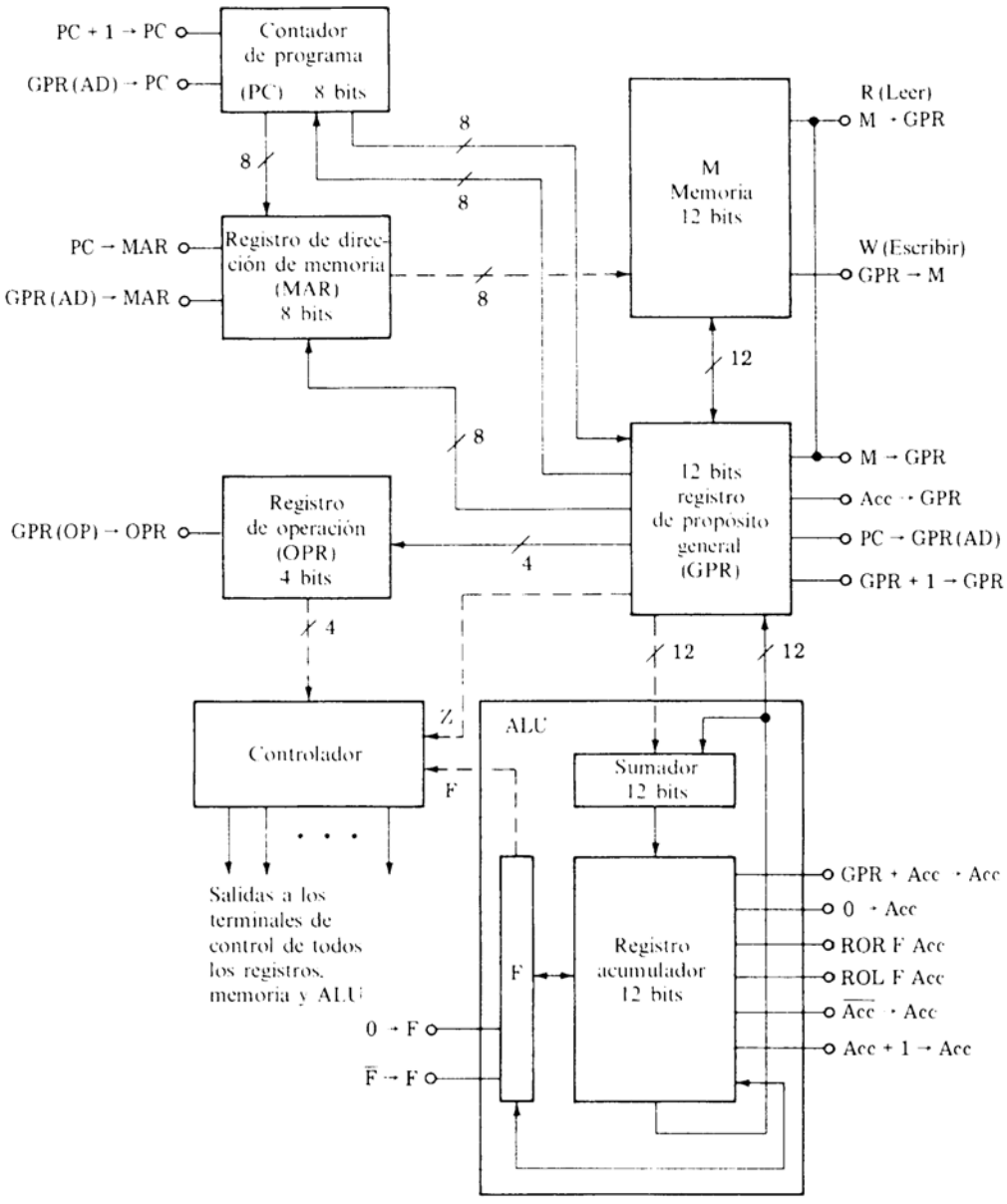
El sistema de memoria incorpora una memoria caché con mapeo directo que permite almacenar 1K palabras de memoria principal. El tamaño de bloque es de 4 palabras.

Se pide representar gráficamente el contenido de la memoria caché después de 3 iteraciones del bucle de código que se presenta a continuación:

```
1: register int sp;
2:
3: sp = 0;
4: for (int i=0; i<n; i+=2)
5:     sp += A[i]*B[i];
```

Nota: Representar únicamente las líneas de caché que han sido modificadas por el programa en ejecución.

(1.5 puntos)



Ejercicio 1.

(a) JPC m

1. Acc ← 0
2. Acc ← GPR + Acc
3. Acc ← $\overline{Acc}$
4. Acc ← Acc + 1, GPR(AD) ← PC
5. Acc ← GPR + Acc
6. Acc ← $\overline{Acc}$
7. Acc ← Acc + 1
8. GPR ← Acc
9. PC ← GPR(AD)

(b) LDR m

1. Acc ← GPR + Acc
2. GPR ← Acc
3. MAR ← GPR(AD)
4. GPR ← M, Acc ← 0
5. Acc ← GPR + Acc

En la instrucción JPC, para conseguir tener en el acumulador la expresión m – PC lo que se hace es introducir en primer lugar m (ciclos 1 y 2). Seguidamente se le cambia de signo, obteniendo así –m (ciclos 3 y 4). En el ciclo 5 sumamos PC al acumulador, por lo que ya tendremos –m+PC. Si cambiamos de signo todo esto (ciclos 6 y 7) tendremos –(–m+PC) = m – PC, que es lo que necesitábamos.

Ejercicio 2.

Tabla de la CROM:

Ciclo	Microoperación	Bits LCB ($S_2S_1S_0$)	Microdirección de salto
ADDR(CUA)	SC ← 12, GPR ← 0	0 0 0	
ADDR(CUA)+1	ROL Acc, SC ← SC – 1	0 0 0	
ADDR(CUA)+2		0 1 0	ADDR(CUA)+4
ADDR(CUA)+3	GPR ← GPR + 1	0 0 0	
ADDR(CUA)+4		0 1 1	ADDR(CUA)+1
ADDR(CUA)+5	Acc ← 0	0 0 0	
ADDR(CUA)+6	Acc ← GPR + Acc	1 0 0	ADDR(FETCH)
ADDR(CUA)+7	PC ← GPR(AD)	0 0 1	ADDR(FETCH)

Tabla de la LCB:

S_2	S_1	S_0	F	Z_{GPR}	Z_{SC}	I	B	R
0	0	0	x	x	x	1	0	0
0	0	1	x	x	x	0	1	0
0	1	0	0	x	x	0	1	0
0	1	0	1	x	x	1	0	0
0	1	1	x	x	0	0	1	0
0	1	1	x	x	1	1	0	0
1	0	0	x	0	x	0	1	0
1	0	0	x	1	x	1	0	0

Ejercicio 3.

Al ser una memoria de 64K se necesitarán $\log_2(64 \cdot 1024) = 16$ bits de dirección.

La memoria caché puede almacenar 1K palabras de memoria principal en bloques de 4 palabras. Por tanto, tendrá $1024 \text{ palabras} \cdot \frac{1 \text{ bloque}}{4 \text{ palabras}} = 256 \text{ bloques}$. Para referenciar 256 bloques son necesarios $\log_2(256) = 8$ bits. Para referenciar 4 palabras dentro de un bloque se necesitan $\log_2(4) = 2$ bits.

Así, el formato de una dirección de memoria principal será:

- Palabra dentro del bloque: 2 bits.
- Bloque de caché: 8 bits.
- Etiqueta: 16 – 8 – 2 = 6 bits.

Primera iteración:

Se accede a los elementos A[0] y B[0]. Estos elementos están situados en las direcciones:

A[0] → 18F0 → 0001 1000 1111 0000.
Bloque de caché accedido: 00111100 → 3C
Etiqueta: 000110 → 06

B[0] → B2D0 → 1011 0010 1101 0000.
Bloque de caché accedido: 10110100 → B4
Etiqueta: 101100 → 2C

Se produce un fallo de caché, al estar ésta vacía, por lo que habrá que rellenar el contenido de los bloques indicados, quedando la caché como sigue:

	BV	Etiqueta	Palabras de datos			
3C	1	06	12	4	67	81
B4	1	2C	102	44	54	241

Segunda iteración:

Se accede a los elementos A[2] y B[2]. Estos elementos están situados en las direcciones:

A[2] → 18F2 → 0001 1000 1111 0010.
Bloque de caché accedido: 00111100 → 3C
Etiqueta: 000110 → 06

B[2] → B2D2 → 1011 0010 1101 0010.
Bloque de caché accedido: 10110100 → B4
Etiqueta: 101100 → 2C

Teniendo en cuenta que en los bloques de caché que se van a acceder su bit de validez está a valor 1 y la etiqueta coincide, el dato requerido por la CPU lo suministra directamente la caché, por lo que se trata esta vez de un acierto de caché, en ambos casos. Los datos facilitados por la caché son el A[2] = 67 y el B[2] = 54. La caché no se modifica.

Tercera iteración:

Se accede a los elementos A[4] y B[4]. Estos elementos están situados en las direcciones:

A[4] → 18F4 → 0001 1000 1111 0100.
Bloque de caché accedido: 00111101 → 3D
Etiqueta: 000110 → 06

B[4] → B2D4 → 1011 0010 1101 0100.
Bloque de caché accedido: 10110101 → B5
Etiqueta: 101100 → 2C

Se produce un fallo de caché, ya que los bloques que se están accediendo están vacíos (no contienen datos válidos), por lo que habrá que rellenar el contenido de los bloques indicados, quedando la caché finalmente como sigue:

	BV	Etiqueta	Palabras de datos			
3C	1	06	12	4	67	81
3D	1	06	231	89	106	198
B4	1	2C	102	44	54	241
B5	1	2C	163	25	175	102