

INTRODUCCIÓN A LOS COMPUTADORES
(6/2/2007; 2ª PARTE: ejercicios; 7 puntos)

CUESTIONARIO I

1) Se tiene el patrón de bits:

H' 8001 0000

Indicar la información representada, suponiendo que los valores almacenados representan:

a) Caracteres codificados en EBCDIC

Consultando la tabla de caracteres EBCDIC se tiene

80	01	00	00
Φ	SOH	NUL	NUL

RESULTADO: Φ SOH NUL NUL

b) Un número entero de 32 bits en signo y magnitud.

8001 0000 → 1000 0000 0000 0001 0000 0000 0000 0000

Al estar representado el número, N , en signo y magnitud, el primer bit es el de signo, como es 1, es negativo. El resto de bits es el valor absoluto del número; con lo que:

$$N = -1\,0000\,0000\,0000\,0000_2 = -2^{16} = -65.536$$

RESULTADO: - 65.536

c) Un número real IEEE754 precisión sencilla.

8001 0000 →	1	000 0000 0	000 0001 0000 0000 0000
	s	e	m

Como el exponente sesgado es todo ceros, según la norma IEEE754, el número es renormalizado, con lo que el exponente almacenado es -126, y no hay un 1 implícito en la posición cero; es decir:

$$N = -2^{-126} \cdot 0,0000\,001_2 = -(2^{-126} \cdot 2^{-7}) = -2^{-133} = -9,18355 \cdot 10^{-41}$$

RESULTADO: - 9,18355·10⁻⁴¹

d) Dos instrucciones máquina de CODE-2.

Consultando la tabla de instrucciones de CODE-2 se tiene que

8001; corresponde a la operación lógica NAND:

NAND r0,r0,r1

0000; corresponde a la instrucción de carga de un registro, concretamente:

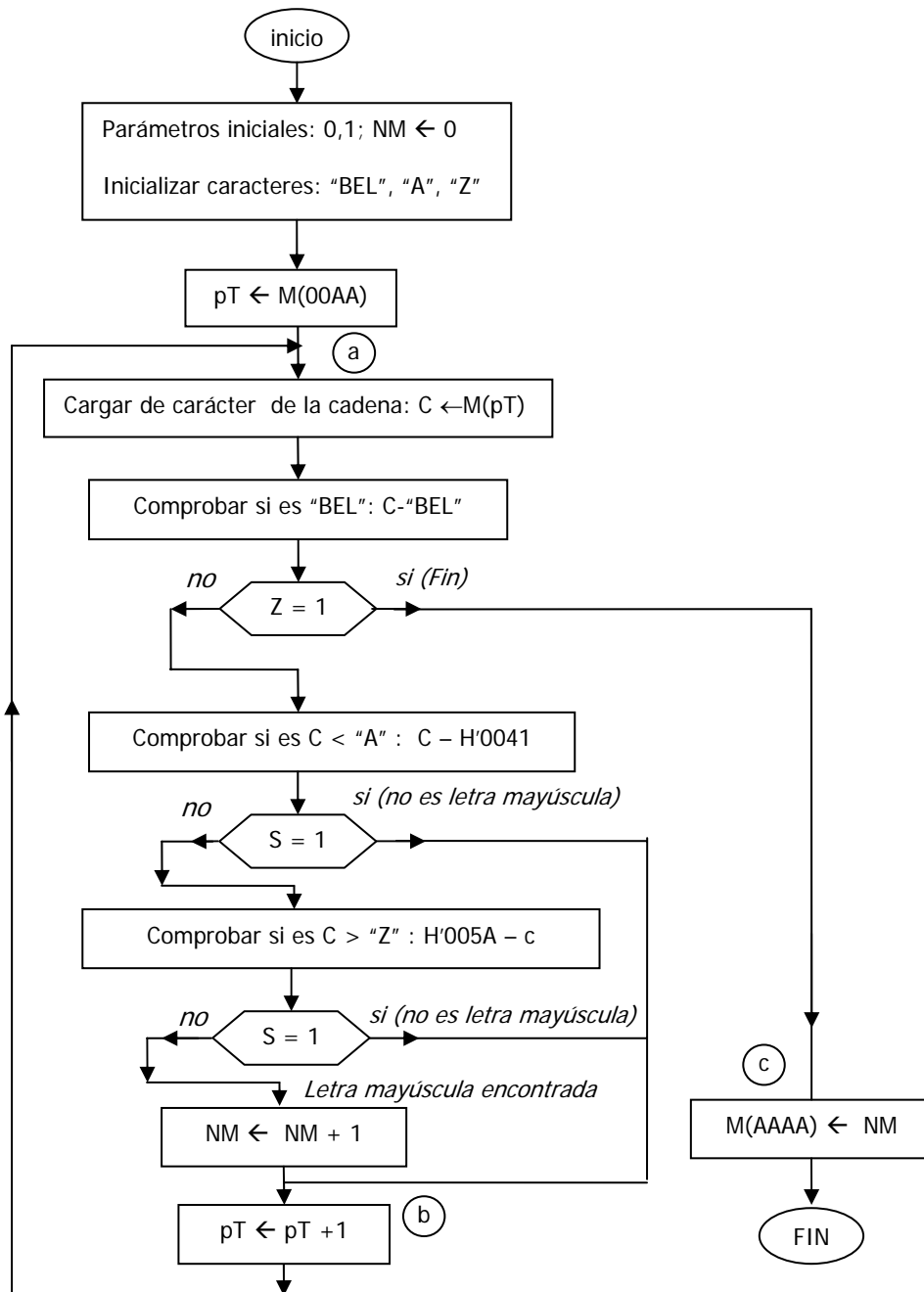
LD r0, [00]

RESULTADO: NAND r0,r0,r1 y LD r0, [00]

2) **En la memoria de CODE-2 se encuentra una cadena de caracteres UNICODE, cuya dirección de inicio se encuentra en la posición de memoria H'00AA y que finaliza con el carácter de control de emisión de un pitido "BEL". Hacer un programa que ubique en la posición de memoria H'AAAA el número de letras mayúsculas que contiene dicho texto.**

a) Realizar un organigrama de la subrutina.

Hacemos que un puntero (pT) vaya recorriendo la cadena y llevando cada uno de los caracteres (C) a un registro. Como los códigos de caracteres están ordenados; y concretamente las letras mayúsculas van de la A → H'0041 a la Z → H'005A, si el carácter está comprendido entre estos valores, es una letra mayúscula, y entonces incrementamos en 1 el número (NM) de mayúsculas.



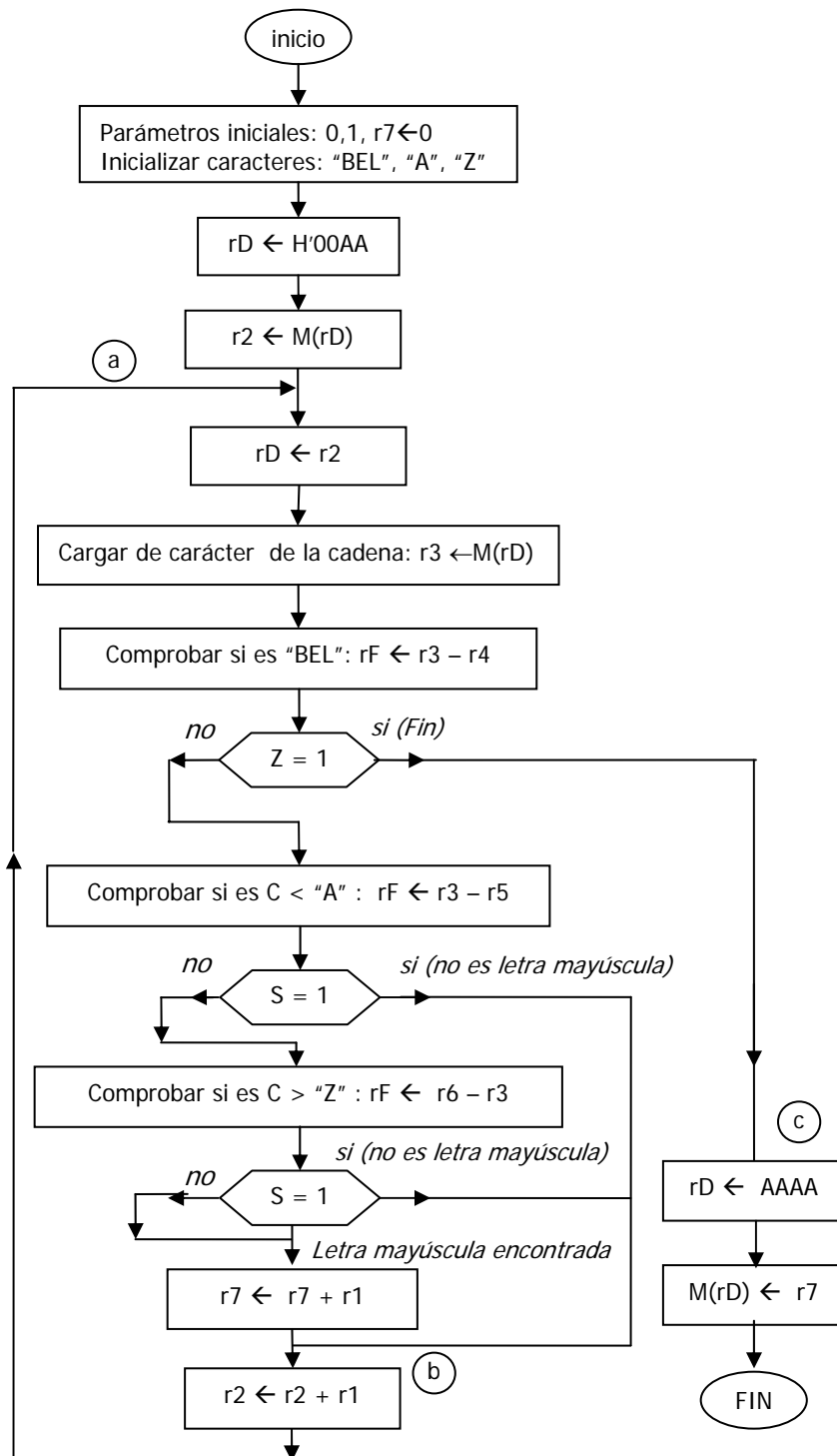
b) Efectuar la asignación de registros y de memoria.

Asignación de registros y de memoria

Parámetro o variable	Registro	Posición de memoria	Comentario
0000	r0		Para pasar un contenido de un registro a otro
0001	r1		Para incrementar el puntero de la cadena
pT	r2		Puntero para recorrer la cadena en memoria
C	r3		Carácter en análisis
0007	r4		Código UNICODE de carácter de control "BEL"
0041	r5		Código UNICODE del carácter "A"
005A	r6		Código UNICODE del carácter "Z"
Programa		0000	Dirección de carga del programa

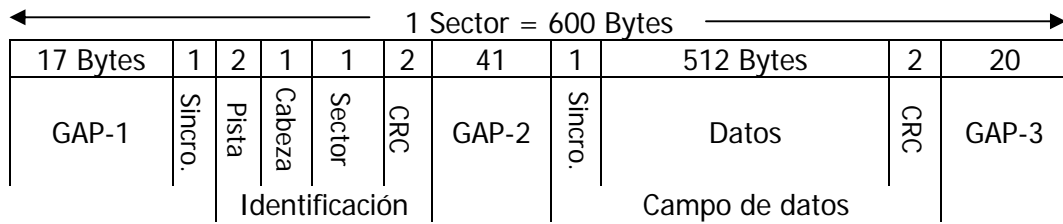
c) Redactar el programa en nemónicos y código máquina

Teniendo en cuenta la asignación de registros realizadas, el organigrama puede ser el siguiente:



De este organigrama es inmediato obtener el programa en nemónicos

- 3) Se dispone de una unidad de disco en el que el formato de grabación de los sectores es el que se indica en la figura.



Suponiendo que la pista 800, se encuentra a una distancia del centro de $1\frac{1}{2}$ " tiene un total de 16 sectores, girando el disco cuando se encuentra en dicha pista la cabeza lectora/grabadora a 7.200 rpm; obtener:

- a) La densidad lineal de grabación en dicha pista.

El número de bits que tiene la pista es:

$$C_{pista} = n_{sectores} \cdot n_{bits/sector} = 16 \cdot (17 + 1 + 2 + 1 + 1 + 2 + 41 + 1 + 512 + 2 + 20) Bytes \cdot 8 \frac{bits}{Byte} = 76.800 bits$$

Por otra parte la longitud de esa pista es:

$$l_{pista} = 2 \cdot \pi \cdot r = 2 \cdot 3,1416 \cdot 1,5 = 9,4248 \text{ pulgadas}$$

Luego la densidad lineal será:

$$\lambda = \frac{C_{pista}}{l_{pista}} = \frac{76.800 bits}{9,4248"} = 8.149 bpi$$

RESULTADO: 8.159 bpi

- b) El tiempo máximo que puede utilizar la controladora del disco para comprobar si el código de error CRC de la identificación es válido para proceder a leer el campo de datos (Pista: este tiempo coincide con el tiempo de recorrido del GAP-2).

Como, una vez leída la cabecera de un sector, el disco sigue girando; para dar tiempo a hacer la comprobación de los CRC se incluye en el formato el GAP-2, constituido por 41 Bytes. Es decir, el tiempo máximo que debe tardar la controladora de disco en hacer la comprobación ha de coincidir con el tiempo en que se tardan en recorrer esos 41 Bytes. Llamando T_r al periodo de giro del motor (tiempo que se tarda en recorrer una pista completa), el tiempo que se invierte en recorrer un Byte será:

$$t_{Byte} = \frac{T_r}{C_{pista}} = \frac{1/\omega_r}{9.600 Bytes} = \frac{60}{9.600} = 0,87 \frac{\mu s}{Byte}$$

Es decir, el tiempo en recorrer los 41 Bytes de GAP-2 será:

$$t_{GAP} = n_{Bytes} \cdot t_{Byte} = 41 \cdot 0,87 = 35,6 \mu s$$

RESULTADO: 35,6 μs

- 4) Suponga un SO de memoria virtual paginada, que utiliza el método de reemplazo NRU; y que cada 4 accesos el bit de referencia, *R*, se pone a cero (es decir, después del 4°, 8°, 12° Acceso). La memoria principal, se estructura en 3 marcos de página, es de tan solo 192 Kpalabras e inicialmente está completamente vacía. Si la memoria se va ocupando de las posiciones de direcciones más bajas a las más altas, y el programa hace referencia sucesiva a las siguientes direcciones:

5A0735	A708AB	7C07BF	CB0DB5	5A6548	CB743E	9EDE30
escritura	lectura	escritura	escritura	escritura	escritura	lectura

Obtener la dirección virtual a la que corresponde la dirección física 2ABC7, después del último acceso.

Solución.

Como hay 3 marcos de página, el tamaño de cada página será:

$$C_{pag} = \frac{C_{memoria}}{N_{mp}} = \frac{192 Kp}{3} = 64 Kp$$

Es decir, como $64 Kp = 2^{16} p$; el desplazamiento de cada página o cada marco de página se dará con 16 bits (4 cifras HEX). En otras palabras, las dos primeras cifras hexadecimales especifican la página, y las últimas cuatro el desplazamiento dentro de ella (o dentro del marco de página, en su caso).

Recuérdese que el método de reemplazo NRU se puede gestionar con dos bits; uno de ellos, *R*, se pone a uno cuando se referencia el marco de página, y el otro, *M*, se pone a uno cuando se modifica la página de ese marco (se escribe en ella). Cuando hay que desalojar una página, se hace de acuerdo con la siguiente prioridad:

R	M
0	0
0	1
1	0
1	1

A continuación se indica como se van rellenando los distintos marcos de página, según van apareciendo los accesos.

	5A0735	R	M	A708AB	R	M	7C07BF	R	M	CB0DB5	R	M	5A6548	R	M	CB743E	R	M	9EDE30
0	5A	1	1	5A	1	1	5A	1	1	5A	0	1	5A	1	1	5A	1	1	5A
1				A7	1	0	A7	1	0	CB	0	1	CB	0	1	CB	1	1	CB
2							7C	1	1	7C	0	1	7C	0	1	7C	0	1	9E
Nota 1										Nota 2					Nota 3				

Nota 1: Se sustituye la página del marco con $R=1, M=0$, ya que las otras tienen menor prioridad de sustitución

Nota 2: Como es el 4° acceso, después de él se ponen a cero todos los bits *R*

Nota 3: Se sustituye la página del marco con $R=0, M=1$, ya que las otras tienen menor prioridad de sustitución

La dirección física (**2ABC7**) tiene cinco cifras hexadecimales; según se justifico antes, las cuatro últimas corresponden al desplazamiento, luego la primera (2) corresponde al marco de página, siendo el desplazamiento ABC7. Como en el marco de página 2 se encuentra la página 9E, la dirección virtual a la que pertenece es: **9E ABC7**.

Dirección virtual: 9E ABC7