ORGA. DEL COMPUTADOR

(si ven algo mal, corrijan!)

Fecha Final 11/12/23

1. Decir qué número es en base decimal un número que está empaquetado de 3 bytes.

xe ej 2076_10

1) paso de decimal a hexa
2) una vez que tengo el hexa identifico si es + o -
3) el nro a la izquierda del signo es el que va

viceversa -> 417F -> la F quiere decir que es positivo
paso a decimal +417_10

->otra versión puede ser

empaquetar 243_8

paso a decimal

$2.8^2 + 4.8^1 + 3.8^0 =$
128 + 32 + 3 = 163_10

otra forma de hacerlo:
a binario
010 100 011
a decimal
$0.2^8 + 1.2^7 + 0.2^6 + 1.2^5 + 0.2^4 + 0.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 128 + 22 + 2 + 1 =$
163_10

o hacer zoneado:

2. Hacer un código ARM que multiplique cada número del arreglo x 3 y sume todos.

```
.equ EXIT, 0x11    @comando final
.data                @variables
     nums: .word 2,1,4,3,5
     cant: .word 5
     res: .word 0
     tres: .word 3

.text                @indica código
```

```
.global _start

_start:
      ldr r0,=nums
      ldr r1,=cant
      ldr r3,=tres
      mov r4,#0
      ldr r5,[r3]

loop:
      cmp r1,#0
      beq fin
      ldr r6,[r0]
      add r0,#4
      mul r6,r5
      add r4,r6
      sub r1,#1
      b loop
fin:
      ldr r0,=res
      str r4,[r0]
      swi EXIT
.end
```

```
@ Hacer un código ARM que multiplique cada número del arreglo x 3 y sume todos.
.data
    cadena:.word 1, 2, 3, 4, 5
    largo: .word 5
    output: .word 0

.text

.global _start
_start:
    ldr r1,=cadena
    ldr r2,=largo
    mov r3,#0           @contador
    mov r4,#0           @total
    mov r5,#3           @multiplicador

loop:
    ldr r6,[r1]         @ cargo en r6 el valor actual de la cadena
    mul r7,r6,#3        @ lo multiplico por 3 y lo guardo en r7
    add r4,r4,r7        @ lo sumo a r4 (el total)

    add r1,r1,#4        @ paso el puntero al siguiente valor
    add r3,r3,#1        @ subo el contador

    ldr r8,[r2]         @ cargo el valor del largo de la cadena en r8
    cmp r8,r3           @ comparo el largo con el contador
    bne loop            @ salto si no son iguales

end:
    ldr r9,=output      @ cargo la dirección del output en r9
    str r4,[r9]         @ cargo el total en el output

    .end
```

3. Hacer código Debug- DOSBOX que imprima ExAmEn

```
100 MOV AX, B800 ; Mueve el valor hexadecimal B800 a AX, que es la
dirección del segmento de memoria de video (VRAM) en modo texto en
color
103 MOV DS, AX  ; Mueve el valor de AX al registro de segmento de
datos (DS), estableciendo así DS a B800 para que apunte a la VRAM

105 MOV BX, 0000 ; Inicializa el registro BX (contador) en 0



114 INT 20        ; Llama a la interrupción 20h (INT 20) para
terminar el programa
```

```
; Inicio del programa
100 MOV AX, B800 ; Mueve el valor hexadecimal B800 a AX, que es la dirección del segmento de memoria de video (VRAM) en modo texto en color
103 MOV DS, AX   ; Mueve el valor de AX al registro de segmento de datos (DS), estableciendo así DS a B800 para que apunte a la VRAM

105 MOV BX, 0000 ; Inicializa el registro BX (contador) en 0

; Imprimir 'E' en la posición actual del cursor
108 MOV [BX], 'E' ; Mueve el valor ASCII de 'E' al byte en la posición actual de la VRAM
10B INC BX           ; Incrementa BX para apuntar a la siguiente posición en la VRAM

; Imprimir 'x' en la posición actual del cursor
10E MOV [BX], 'x' ; Mueve el valor ASCII de 'x' al byte en la posición actual de la VRAM
10F INC BX           ; Incrementa BX para apuntar a la siguiente posición en la VRAM

; Imprimir 'A' en la posición actual del cursor
102 MOV [BX], 'A' ; Mueve el valor ASCII de 'A' al byte en la posición actual de la VRAM
11 INC BX            ; Incrementa BX para apuntar a la siguiente posición en la VRAM

; Imprimir 'm' en la posición actual del cursor
11E MOV [BX], 'm' ; Mueve el valor ASCII de 'm' al byte en la posición actual de la VRAM
11F INC BX           ; Incrementa BX para apuntar a la siguiente posición en la VRAM

; Imprimir 'E' en la posición actual del cursor
110 MOV [BX], 'E' ; Mueve el valor ASCII de 'E' al byte en la posición actual de la VRAM
111 INC BX           ; Incrementa BX para apuntar a la siguiente posición en la VRAM

; Imprimir 'n' en la posición actual del cursor
112 MOV [BX], 'n' ; Mueve el valor ASCII de 'n' al byte en la posición actual de la VRAM

114 INT 20           ; Llama a la interrupción 20h (INT 20) para terminar el programa
```

4. Código máquina elemental que busques un dato en una dirección y si es positivo le pongas -1 en esa dirección y sino que le hagas el complemento a la base y lo guardes en el mismo lugar.
~suponiendo que el dato está en la dirección 22~

```
00 1122 //cargo en R1 lo que se encuentre en 22
02 2280 //cargo lo en r2 = 1000 0000
04 8312 // And de R1 y R2 y lo ubico en R3
06 2000 // R0= 00
08 B210 //salto si R2 == R0 a la instrucción 10
0A 24FF // cargo en R4= 1111 1111
0C 3422 // cargo R4 en [22]
0E B01A // salto si R0 = R0
10 24FF //cargo en R4 = 1111 1111
12 9524 //Xor de R2 y R4 y lo guardo en R5
14 2601 // guardo en R6 = 0000 0001
16 5756 //sumo en R7 lo de R5 y R6
18 3722 //cargo lo de R7 a [22]
1A C000 //fin de programa
```

5. Si un disco tiene 7 caras, 800 cilindros, 7 sectores, 512 bytes, ¿cuánto puede almacenar?
7 caras =>
800 cilindros
7 sectores=>
512 bytes => 2^9

Capacidad del disco = caras x cilindros x sectores x bytes
C = 7x800x7x512 = 20070400 bytes
si se quiere pasar a mb podemos hacer
20070400 bytes / 1024^2 = 20 mb +- aproximadamente.

6. hacer un mapa de memoria
[literal el mismo del segundo parcial]

a.
Cantidad de líneas de direccionamiento de bus de direcciones
1M = 1K . 1K = 2^10 . 2^10 = 2^20 (desde A0 hasta A19)

IC1 256Kbytes = 2^8 . 2^10 bytes = 2^18 bytes (desde A0 hasta A17)
IC2 128kbytes = 2^7 . 2^10 bytes = 2^17 bytes (desde A0 hasta A16)
IC3 128kbytes = 2^7 . 2^10 bytes = 2^17 bytes (desde A0 hasta A16)
IC4 512Kbytes = 2^9 . 2^10 bytes = 2^10 bytes (desde A0 hasta A18)

b.

A19	A18	A17	A16	A0	
0	0	0	0	0	IC1
0	0	1	1	1	
0	1	0	0	0	IC2
0	1	0	1	1	
0	1	1	0	0	IC3
0	1	1	1	1	
1	0	0	0	0	IC4
1	1	1	1	1	

2da fecha

En una ALU de 8 bits realzar las siguientes operaciones considerando los operandos en complemento a 2. Obtener el resultado y justificar si se excede o no el rango de representación:

a) **55h - BBh** b) **F6h - 5Fh**

Detectar si el contenido de la dirección EE tiene el bit 2 en uno y el bit 3 en cero. En caso afirmativo dejar en EE el bit 0 en uno y el bi7 en cero, en caso contrario dejar en EE el bit 1 en uno y el bit 6 en cero sin modificar los bits restantes. El punto de carga es la dirección 20.

Dos cadenas de caracteres, A y B, de longitudes iguales e igual al dato almacenado en la dirección 01FFh comienzan en las direcciones 0200h la cadena A y 0300h la cadena B.
Armar una cadena de caracteres C a partir de la dirección 0400h intercalando cada elemento de la cadena A con cada elemento de la cadena B uno a uno. El programa comienza en la dirección 0100h

Escribir un código ARM, para el procesador de un surtidor de nafta, que muestre el importe a pagar. El mapeo de memoria del surtidor es el siguiente:
Dirección: FF200020h. Contenido: Litros de nafta vendidos.
Dirección: FF200030h. Contenido: Precio del litro de nafta.
Dirección: FF200040h. Contenido: Importe a pagar que muestra un display.

Diseñar la memoria principal de una computadora que tiene 256 Kbytes de memoria ROM y 512 Kbytes de memoria RAM. Para la memoria ROM se dispone de dos circuitos integrados de 64 Kbytes, tres de 512 Kbytes y uno de 128 Kbytes. Para la memoria RAM se tienen dos circuitos integrados de 256 K palabras de 4 bits, dos de 128 Kbytes y uno de 64 Kbytes. Se pide:

a. Indicar cuántos y cuáles circuitos integrados se van a usar.
b. Armar un mapa de memoria en el que se vea la primera dirección y la última para cada circuito integrado.
c. Decodificar los selectores de integrado (Chip Select).
d. Dibujar el circuito.

¿A qué velocidad se transfieren los datos de un disco con 26 superficies de grabación, 16.000 cilindros, 80 sectores por pista, 1024 Bytes por sector y 5400 rpm?

Sistemas numéricos

Máquina elemental

Assembly

ARM

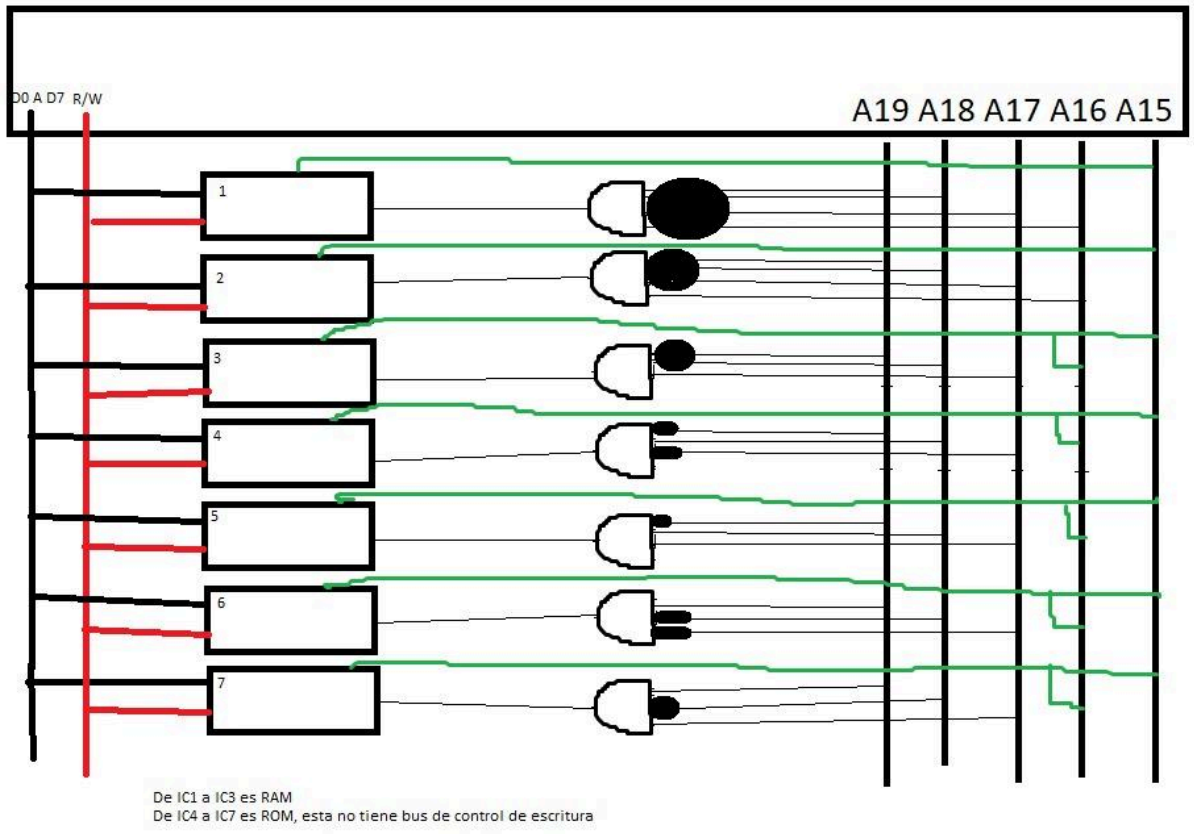
Escribir un código ARM, para el procesador de un surtidor de nafta, que muestre el importe a pagar. El mapeo de memoria del surtidor es el siguiente:
Dirección: FF200050h. Contenido: Litros de nafta vendidos.
Dirección: FF200040h. Contenido: Precio del litro de nafta.
Dirección: FF200000h. Contenido: Importe a pagar que muestra un display.

```
.equ EXIT, 0x11
.equ DIR_LITROS,  0xFF200050
.equ DIR_PRECIO,  0xFF200040
.equ DIR_DISPLAY, 0xFF200000

.global _start
_start:
        ldr r0, =DIR_LITROS ;; r0=  0xFF200050
        ldr r1,[r0] ;;; en r1 esta el contenido de  0xFF200050
        ldr r0,  =DIR_PRECIO ;; r0=  0xFF200040
        ldr r2, [r0] en r2 lo que hay aca dentro 0xFF200040
        ldr r0, =DIR_DISPLAY ;;; r0 = 0xFF200000
        mul r3,r1,r2 ;;; multiplico litro x precio y guardo en r3
        str r3,[r0]

        swi EXIT
.end
```

Memoria



(los puntos quiere decir negado)

Discos

26 superficies de grabación
16000 cilindros
80 sectores por pista
1024 bytes por sector
5400 rpm
yo se que :
tiempo de transferencia = b/rN
tiempo de acceso = tseek + t delatencia
tiempo de lectura/escritura = tseek+1/2r+tiempo de transferencia
r es la velocidad de rotación

Velocidad de transferencia = velocidad de rotación x tamaño de sector x Bytes por pista
5400 rpm -> a rev x seg 5400/60 = 90
90x80x1024=7372800 bytes por segundo = 7MB/s aproximadamente

3era fecha 05/02

Sistemas numéricos

Resta en representación de enteros de 8 bits y poner las flags estan en hexadecimal

55
-BB -> paso a binario

Máquina elemental

Restar BB-CC (lo que está en esas direcciones) y poner el resultado en EE si es positivo o el complemento si es negativo
el codigo arranca en 20

xor
1111 1111 = FF
0000 0001 =01

and 1000 0000
+ 0000 0000
- 1000 0000

20 11BB ; guardo en r1 lo que hay en BB
22 12CC ; guardo en r2 lo que hay en CC
24 23FF ; en r3 = **1111 1111**
26 2401; r4 = **0000 0001**
28 9523 ; xor entre r2 y r3
2A 5554; suma de r5 y r4 y lo guardó en r5
2C 5515; suma del complemento de cc + lo que hay en bb —> r5= rta
2E 2080; cargo la máscara 1000 0000 en r0
30 8750; hago and de la respuesta y la máscara para evaluar si es + o -
32 B736 ; compara lo de r7 (si + o -) con r0 ;; si es = seria negativo
34 B03A; salto incondicional al final.
36 9553 ; aca pq es negativo hace el complemento del resultado. hago xor para el complemento -1
38 5554; sumo 1
3A 35EE; guardo en [EE] el resultado pq es positivo
3E C000; fin de programa.

Emu8086

Cadena A en 0200h y B en 0300h. Ambas de largo 10. Guardar resultado en 0400h

Llenar primero todo con una y después todo con otra y a la primera ponerle el destello y a la segunda fondo blanco con letras rojas

Datos: color blanco:1111b color rojo: 0100

```
.org 100h

mov cx,10h
mov si, 0200h  ; inicio cadena A
mov di, 0300h  ;inicio cadena B
mov ax, 0400h ; inicio cadena C

loop:
        mov bx,[si] ; paso lo que hay en [si] al registro bx
        mov [ax], bx ; lo que quedo en bx, lo escribo en [ax]
        add ax,02h
```

Assembler

Sistema que hace sonar una alarma si el sensor detecta a alguien
En 0xFF200040 hay un sensor (1 detecta, 0 no hay nada)
En 0xFF200000 hay una alarma (1 está sonando, 0 no suena)

```
.equ EXIT,  0x11
.data
        sensor: .word 0xFF200040
        alarma: .word  0xFF200000

.global _start
_start

        ldr r1, =sensor @en r1 esta está a direccion de memoria 0xFF200040
        ldr r2,[r1]  @en r2 esta el valor de lo que esta en esta dirección 0xFF200040  (lo que hay dentro)

        ldr r1,=alarma @en r1 esta esta direccion de memoria 0xFF200000
        str r2,[r1] @STORE lo de r2 a r1.
        swi EXIT
.end
```

Otra forma de hacerlo
.equ SWITCH, 0xFF200040 @ la dirección está en la ventana del switch en el simulador
.equ LED, 0xFF200000 @ la dirección está en la ventana del LED en el simulador
.global _start

_start: ldr r0,=SWITCH ldr r1,[r0] @ lectura del dispositivo de entrada almacenada en r1
ldr r0,=LED str r1,[r0] @ escritura del dispositivo de salida con el dato de r1
.end

Memoria

Armar un banco de memoria de una rom de 1gb y una ram de 2
ROM: 2x256MB y 1x512MB
RAM: 2x512MB y 1x1GB

Banco de memoria 1gb + 2gb = 3gb => 2^2 * 2^30 = 2^32 -> de A0…. A31
4GB

ROM tengo 1gb total.
256MB = 2^8*2^20= 2^28 -> A0….A27 IC1
256MB = 2^8*2^20= 2^28 -> A0….A27 IC2
512MB = 2^9*2^20= 2^29 -> A0….A28 IC3

RAM tengo 2gb total
512MB = 2^9*2^20= 2^29 -> A0….A28 IC4
512MB = 2^9*2^20= 2^29 -> A0….A28 IC5
1GB = 2^0 *2^30 = 2^30 -> A0….A29 IC6

a) son 6 circuitos integrados

A31	A30	A29	A28	A27	A26	…….. A0	
0	0	0	0	0	0	…….. 0	
0	0	0	0	1	1	…….. 1	IC1 +1
0	0	0	1	0	0	…….. 0	
0	0	0	1	1	1	…….. 1	IC2 +1
0	0	1	0	0	0	…….. 0	
0	0	1	1	1	1	…….. 1	IC3 +1
0	1	0	0	0	0	…….. 0	
0	1	0	1	1	1	…….. 1	IC4 +1
0	1	1	0	0	0	…….. 0	
0	1	1	1	1	1	…….. 1	IC5 +1
1	0	0	0	0	0	…….. 0	
1	0	1	1	1	1	…….. 1	IC6

Chip select

	A31	A30	A29	A28	A27
IC1	0	0	0	0	X
IC2	0	0	0	1	x
IC3	0	0	1	x	x
IC4	0	1	0	x	x
IC5	0	1	1	x	x
IC6	1	0	x	x	x

U6

V o F. Los sectores tienen pistas. Falso
V o F. Las pistas tienen sectores. Verdadero
Cuántas pistas hay si se tienen 36.000 cilindros. -> 36000 pistas

práctica inventada:

Memoria

Armar un banco de memoria con ROM (512KBytes) y RAM (1MByte)

Chips posibles de ROM: 2 de 128KBytes y 3 de 256KBytes
Chips posibles de RAM: 1 de 512KBytes , 1 de 256KBytes, 2 de 128KBytes y 4 de 64KBytes

ROM: 512KBytes = 2^9*2^10 = A0….A18
RAM: 1MByte = 2^0*2^20= A….19

total: 2MBytes = 2^1*2^20 = A0…A20
RAM:
1x 512KBytes = 2^9*2^10 = A0…A18 IC1
1x 256KBytes = 2^8*2^10 = A0…A17 IC2
1x 128Kbytes = 2^7*2^10 = A0..A16 IC3
1x 128Kbytes = 2^7*2^10 = A0..A16 IC4

ROM:
1x 256KBytes = 2^8*2^10 = A0…A17 IC5
1x 256KBytes = 2^8*2^10 = A0…A17 IC6

A20	A19	A18	A17	A16	A15	……..	A0	
0	0	0	0	0	0	……..	0	
0	0	1	1	1	1	……..	1	IC1 +1
0	1	0	0	0	0	……..	0	
0	1	0	1	1	1	……..	1	IC2 +1
0	1	1	0	0	0	……..	0	
0	1	1	0	1	1	……..	1	IC3 +1
0	1	1	1	0	0	……..	0	
0	1	1	1	1	1	……..	1	IC4 +1
1	0	0	0	0	0	……..	0	
1	0	0	1	1	1	……..	1	IC5 +1
1	0	1	0	0	0	……..	0	
1	0	1	1	1	1	……..	1	IC6

	A20	A19	A18	A17	A16
IC1	0	0	x	x	x
IC2	0	1	0	x	x
IC3	0	1	1	0	x
IC4	0	1	1	1	x
IC5	1	0	0	x	x
IC6	1	0	1	x	x

total: 2MBytes = 2^1*2^20 = A0…A20

ROM:
1x 256KBytes = 2^8*2^10 = A0…A17 IC1
1x 256KBytes = 2^8*2^10 = A0…A17 IC2

RAM:
1x 128Kbytes = 2^7*2^10 = A0..A16 IC3
1x 128Kbytes = 2^7*2^10 = A0..A16 IC4
1x 256KBytes = 2^8*2^10 = A0…A17 IC5
1x 512KBytes = 2^9*2^10 = A0…A18 IC6

A20	A19	A18	A17	A16	A15	……..	A0	
0	0	0	0	0	0	……..	0	
0	0	0	1	1	1	……..	1	IC1 +1

0	0	1	0	0	0	…….. 0	
0	0	1	1	1	1	…….. 1	IC2 +1
0	1	0	0	0	0	…….. 0	
0	1	0	0	1	1	…….. 1	IC3 +1
0	1	0	1	0	0	…….. 0	
0	1	0	1	1	1	…….. 1	IC4 +1
0	1	1	0	0	0	…….. 0	
0	1	1	1	1	1	…….. 1	IC5 +1
1	0	0	0	0	0	…….. 0	
1	0	1	1	1	1	…….. 1	IC6

chip select

	A20	A19	A18	A17
IC1	0	0	0	x
IC2	0	0	1	x
IC3	0	1	0	0
IC4	0	1	0	1
IC5	0	1	1	x
IC6	1	0	x	x

Hacer un banco de memoria de 256MB con 2 chips de 32MB, uno de 64 MB y otro de 128MB en ese orden

total 256MB = 2^8*2^20 = A0…A27

32MB = 2^5*2^20 = A0…A24 IC1
32MB = 2^5*2^20 = A0…A24 IC2
64MB = 2^6*2^20 = A0…A25 IC3
128MB = 2^7*2^20 = A0…A26 IC4

A27	A26	A25	A24	A23	…….. A0	
0	0	0	0	0	…….. 0	
0	0	0	1	1	…….. 1	IC1 +1
0	0	1	0	0	…….. 0	

0	0	1	1	1	.....	1 IC2
						+1
0	1	0	0	0	.....	0
0	1	1	1	1	.....	1 IC3
						+1
1	0	0	0	0	.....	0
1	1	1	1	1	.....	1 IC4

chip select

	A27	A26	A25
IC1	0	0	0
IC2	0	0	1
IC3	0	1	x
IC4	1	x	x

RESUELTO por el profe.

U1 Resolver las siguientes operaciones e indicar las flags correspondientes.

a) A2-73

```
 1010 0010    A
-0111 0011  - B

___________
 1000 1101   Cb(B)
+1010 0010  + A

___________
10010 1111
```

C = 1 es el 1 que me llevé y que quedó más a la izquierda que el bit más significativo.
V = 1 es que el resultado no entra en 8 bits. Se necesitan 9. Me doy cuenta porque los bits de signo no son coherentes ya que se suman dos negativos y el rsultado da positivo.
N = 0 el resultado no es negativo.
Z = 0 el resultado no dió cero
P = 1 porque los 5 unos del resultado más éste bit de paridad dan una cantidad par de unos (6).

b) B5 - A7

```
 A:   1011 0101
-B:  -1010 0111

________________
Cb(B):0101 1001
+A:  +1011 0101

________________
     10000 1110
```

C = 1
V = 0 porque el resultado de la suma de un número positivo más un número negativo siempre está en el medio de esos dos. No puede ser mayor que el positivo ni menor que el negativo.
N = 0 porque el signo del resultado es positivo
z = 0
P = 1

U2

Se tiene 2 números en representación de enteros en las direcciones de memoria BB y CC. Se pide hacer la resta de BB-CC (del contenido de esas direcciones) y determinar si el resultado dio negativo o no. En caso afirmativo dejar en EE el complemento a la base del resultado, en caso contrario dejar en EE el resultado. El punto de carga es la dirección 20.

```
20 11BB
22 12CC
24 23FF
26 2401
28 9523
2A 5554
2C 5651
2E 2000
30 2780
32 8867
34 B83A si R8 = R0 salta a 3A
Hasta aquí copié tu programa. A partir de aquí tenes error al no hallar el complemento a la base del resultado.
36 9963
38 5694
3A 36EE
```

U3

```
; Dos cadenas de caracteres, A y B, de longitudes iguales a 10 que comienzan en las direcciones
;0200h la cadena A y 0300h la cadena B.
;Armar, una cadena de caracteres C a partir de la dirección 0400h con cada
;elemento de la cadena A y luego con cada elemento de la cadena B (EJ: Cad A xxxxxxxxxx, Cad
;B yyyyyyyyyy, en C quedara xxxxxxxxxxyyyyyyyyyy. Se pide que la primera cadena titile y que la
;segunda tenga fondo blanco y letras en rojo. El programa comienza en la dirección 0100h

org 100h
```

```
mov cx,000Ah
dec cx
mov bx,0400h
mov di, 0200h
mov si, 0300h
primera: mov ax,[di]
mov ah, 8h
mov [bx],ax
inc di
inc bx
inc bx
loop primera
mov cx, 000ah
dec cx
segunda:mov ax,[si]
mov ah, 0F4h
mov [bx], ax
inc si
inc bx
inc bx
loop segunda
int 20h

ret
```

U4

```
@Escribir un código ARM para el procesador que controla un detector de presencia que
@consiste en y un sensor y una alarma.
@El mapeo de memoria es el siguiente:
@La dirección FF200040h contiene un 1 si el sensor detecta presencia o un 0 sin no la detecta.
@La dirección FF200000h hace sonar la alarma si su contenido es un 1 y no suena si es 0.
@ Simulador https://cpulator.01xz.net/?sys=arm-de1soc

.equ SENSOR, 0xFF200040       @ la dirección del sensor de presencia
.equ ALARMA, 0xFF200000       @ la dirección de la alarma

.global _start
_start:
           ldr r0,=SENSOR
           ldr r1,[r0]      @ lectura del dispositivo de entrada
(sensor de presencia) almacenada en r1

           ldr r0,=ALARMA
           str r1,[r0]       @ escritura en el dispositivo de salida
(alarma)

           .end
```

U5
Diseñar la memoria principal de una computadora que tiene 1 GB de memoria ROM y
2 GB de memoria RAM, ambas memorias son de 8 bits de datos. Para la memoria ROM se
dispone de dos circuitos integrados de 256 MB, y uno de 512 MB, todas de 4 bits de palabra.
Para la memoria RAM se tienen dos circuitos integrados de 512 MB y la otra de 1 GB, ambas de
4 bits . Se pide:
a. Indicar cuántos y cuáles circuitos integrados se van a usar.
b. Armar un mapa de memoria en el que se vea la primera dirección y la última para
cada circuito integrado.
c. Decodificar los selectores de integrado (Chip Select).
d. Dibujar el circuito.

ROM 1GB = 10^30 BYTES ---> A0 hasta A29
RAM 2GB = 10^31 BYTES ----> A0 hasta A30
TOTAL 3GB = 10^32 BYTES --> A0 hasta A31

a. RAM 2GB = 1GB (IC1)+ 512MB (IC2)+ 512MB (IC3)
   ROM 1GB = 512MB (IC4)+ 256MB (IC5)+ 256MB (IC6)

b.
1GB ---> A0 .. A29
512MB--> A0..A28
256MB--> A0..A27

La primer dirección de un IC es igual a la última dirección del anterior +1.

A31	A30	A29	A28	A27.....	A1	A0	
0	0	0	0	0	0	0	PRIMERA DE IC1
0	0	1	1	1	1	1	ÚLTIMA DE IC1
0	1	0	0	0	0	0	PRIMERA DE IC2
0	1	0	1	1	1	1	ÚLTIMA DE IC2
0	1	1	0	0	0	0	PRIMERA DE IC3
0	1	1	1	1	1	1	ÚLTIMA DE IC3
1	0	0	0	0	0	0	PRIMERA DE IC4
1	0	0	1	1	1	1	ÚLTIMA DE IC4
1	0	1	0	0	0	0	PRIMERA DE IC5
1	0	1	0	1	1	1	ÚLTIMA DE IC5
1	0	1	1	0	0	0	PRIMERA DE IC6
1	0	1	1	1	1	1	ÚLTIMA DE IC6

La última dirección no es todo unos porque con 32 bits se puede direccionar 4 GBytes y sólo tenemos 3.

c.
A31 A30 A29 A28 IC

```
0   0   0   0     IC1  A31 = 0 Y A30 = 0
0   0   1   1
_____________________
0   1   0   0     IC2  A31 = 0,  A30 = 1 Y A29 = 0
0   1   0   1
_____________________
0   1   1   0     IC3  A31 = 0,  A30 = 1 Y A29 = 1
0   1   1   1
_____________________
1   0   0   0     IC4  A31 = 1,  A30 = 0 Y A29 = 0
1   0   0   1
_____________________
1   0   1   0     IC5  A31 = 1,  A30 = 0, A29 = 1 Y A28 = 0
1   0   1   0
_____________________
1   0   1   1     IC6  A31 = 1,  A30 = 0, A29 = 1 Y A28 = 1
1   0   1   1
```

U6

a. (Verdadero/Falso): una pista se divide en varias unidades llamadas sectores.

b. (Verdadero/Falso): un sector consiste en varias pistas.

c. Un disco de 4 platos tiene 36.000 cilindros. ¿Cuántas pistas tiene?

Googlear disco magnético cilindros sectores pistas cabezas y ver las imagenes

a. verdadero
b. falso
c. cada cilindro es un cilindro imaginario que atraviesa la misma pista de cada plato.
4 platos x 2 caras por plato x 36.000 pistas por cara = 144.000 pistas

FINAL 19/02

unidad 1 - sist numéricos
me dieron dos números en hexa, había que hacer la resta de esos en binario , evaluar si se excede o no y marcar los flags. Eran dos restas.

unidad 2 - máquina elemental
a) armar una tabla de la verdad con 2 entradas y una salida, donde una luz se prende solo si la venta está cerrada y la puerta abierta (algo así) y mostrar el operador lógico a utilizar.
b) codificar un programa con las instrucciones de brookshear que evalúe si un número de 8 bits tiene el bit 3 en 0 y el bit 6 en 1, en caso afirmativo guardar en la dirección [EE] el complemento de ese número y en caso contrario cambiar el bit 6 a 1 y el bit 3 a 0, sin modificar el resto de los digitos.

unidad 3 - assembly
dos cadenas que tienen cierca cantidad de caracteres xej [XXXXX] y [YYY], habia que contar en cada cadena cuantos digitos hay y guardar en las direccione 0400h y 0402h correspondientes a cada una de las cadenas. [en la primera hay 5 caracteres y en la segunda debería haber 3]

unidad 4
la idea era hacer una sección de codigo que se puede reutilizar en otros programas. tenemos 3 variables cantidad, cadena y total. había que sumar(? la verdad no me acuerdo q habia que hacer) cada numero de la cadena y mostrarlo en el total.

unidad 5
memoria
se quiere conectar algo de 64KB a una memoria de 48KB
tenes disponible 2 CI
a)cuantos ci vas a usar, cantidad de lineas q usan y de cuanto son cada uno.
b)dibujar mapa de memoria
c)chip select
d)dibujar circuito completo

unidad 6
Si la tasa de transferencia es de 120 mbps(megabits por segundo), cuanto tardara en transferir 2 gbytes ?