

Apellidos :

Nombre :

Grupo :

D.N.I. :

EJERCICIOS: (7,5 puntos)

1. (1 pto.) Suponga que un computador trabaja con datos enteros y con longitud de palabra $n = 8$ bits. Dados los datos de la columna de la derecha de la tabla siguiente, indique su representación interna en la forma Signo Magnitud (para el dato -9), Complemento a 1 (para el dato -1), Complemento a 2 (para el dato +9) y Representación Sesgada (para el dato -1, siendo el sesgo de $S = 2^{n-1}$).

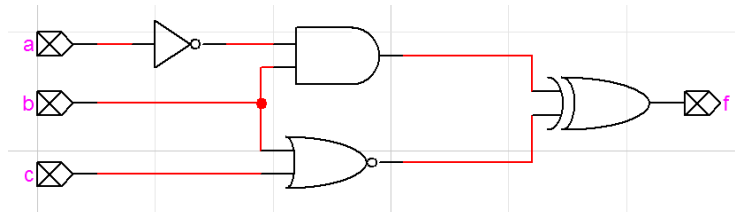
	Representación interna	Valor decimal que representa
(Signo Magnitud)		- 9
(Complemento 1)		- 1
(Complemento 2)		+9
(Sesgada)		- 1

2. (1,5 pto.) En el computador Sencillo (CS1) los datos son de 8 bits, las direcciones son de 6 bits. Tiene un Repertorio de 4 Instrucciones (STOP, ADD, SUB, STA). La fase de captación siempre consume 2 ciclos de reloj. La fase de ejecución consume STOP (1 ciclo), ADD y SUB (3 ciclos) y STA (2 ciclos). Si la frecuencia de reloj del CS1 es de 50 MHz y se ejecuta un programa benchmark que contiene 24 instrucciones ADD, 26 instrucciones SUB, 10 instrucciones STA y una instrucción STOP. Indicar:

- Número de ciclos de reloj totales consumidos por el programa.
- Tiempo de ejecución del programa de benchmark.
- Prestaciones del CS1 en MIPS (Millions Instructions Per Second).
- Tamaño de memoria máximo direccionable en Bytes.

3. (1 pto.) Analice el circuito de la figura 1 y obtenga razonadamente la tabla de verdad de la función de conmutación $Z(X,Y)$ resultante. ¿Existe un circuito equivalente del tipo AND/OR?

Figura 1



4. (0,5 pto.) Para el circuito de la Figura 2a, (donde la salida de datos de un multiplexor 4:1 se ha conectado con la entrada de datos de un demultiplexor 1:4), complete las filas de la tabla de la Figura 2b, escribiendo los valores "0" o "1" que se obtendrían en las salidas del demultiplexor, considerando las señales de control $S1m$ $S0m$ (del MUX) y $S1d$ $S0d$ (del DMUX) indicadas en dicha tabla y las entradas aplicadas en el MUX (Figura 2a). Se muestra, como ejemplo, la respuesta a las dos primeras filas.

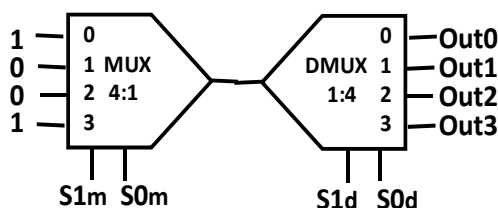


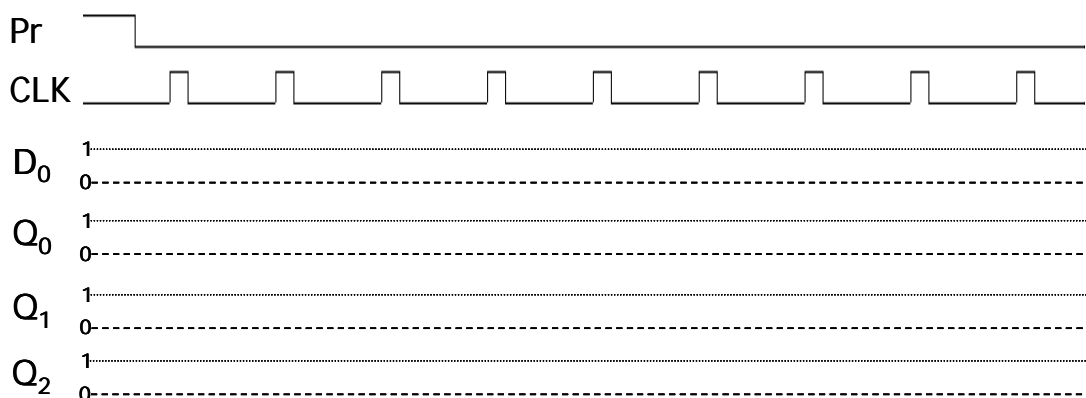
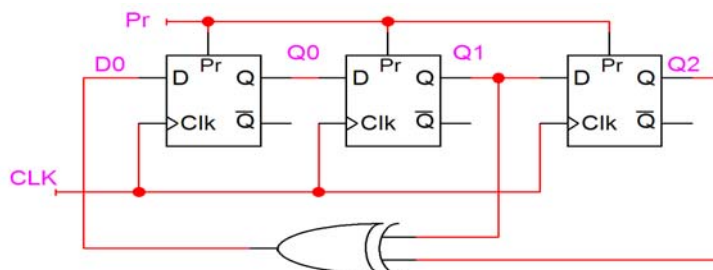
Figura 2a

Señales de control del MUX $S1m$ $S0m$	Señales de control del DMUX $S1d$ $S0d$	Salidas del DMUX			
		Out0	Out1	Out2	Out3
0 0	0 0	1	0	0	0
0 1	0 0	0	0	0	0
1 1	1 0				
0 0	1 1				
1 0	0 1				
1 1	1 1				

Figura2b

5. (1 pto.) Dadas las siguientes funciones de conmutación, en las que "x₀" representa la variable menos significativa:
- $f_0(x_2, x_1, x_0) = \sum m(0, 2, 7)$
 - $f_1(x_2, x_1, x_0) = \sum m(1, 2, 6)$
 - $f_2(x_2, x_1, x_0) = \sum m(3, 4, 5, 6)$
- Implemente dichas funciones mediante una ROM de tamaño adecuado. ¿Cuál es el tamaño de dicha memoria ROM? Dibuje explícitamente la estructura interna de la ROM con las conexiones adecuadas entre el plano AND y el plano OR.

6. (1 pto.) Complete el siguiente diagrama de tiempos para el circuito de la figura. Al principio la señal de Preset (Pr) está activada tal y como se indica en el cronograma, por lo que todos los biestables comienzan con un valor Q=1.



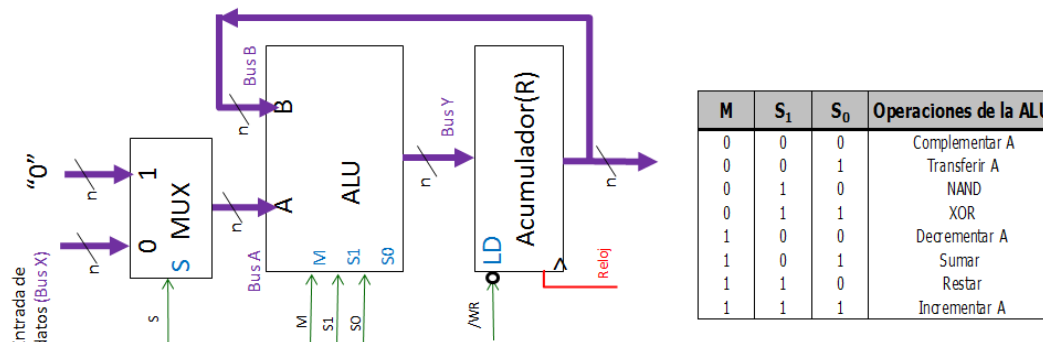
7. (1 pto.) Diseñe un generador de secuencia síncrono, con una señal de control M, tal que si M=0 la secuencia sea ascendente (1,4,4,7...), y si M=1 la secuencia sea descendente (7,4,4,1...). Utilice el tipo de biestable que prefiera.

Si M=0 se genera la Secuencia: (1, 4, 4, 7, 1, 4, 4, 7,)

Si M=1 se genera la Secuencia: (7, 4, 4, 1, 7, 4, 4, 1,)

8. (0,5 pto) Para la unidad de procesamiento de la figura.

- Rellenar la tabla adjunta con los valores de las señales de control. La primera fila viene rellena como ejemplo.
- Indicar la secuencia de **DOS** microoperaciones RT (entre las que se indican en la tabla) para almacenar en R un valor final igual a '15', teniendo siempre X='8'.



M	S ₁	S ₀	Operaciones de la ALU
0	0	0	Complementar A
0	0	1	Transferir A
0	1	0	NAND
0	1	1	XOR
1	0	0	Decrementar A
1	0	1	Sumar
1	1	0	Restar
1	1	1	Incrementar A

Operaciones RT	PALABRA DE CONTROL (Señales de control)					(En hexadecimal)
	Selección de entrada	Controles de la ALU			Escritura en acumulador R /WR	
		S	M	S1		
R<-- X+R	0	1	0	1	0	0A
R<-- X+1						
R<--X						
R<--X XOR R						
R<--X - 1						

2. (0,50 pto.) En la tabla de la figura siguiente se indica el repertorio de las 4 instrucciones del computador simple CS1, indicando sus nombres en ensamblador, el resultado de su ejecución descrita a nivel de transferencia a registros (RT) y su formato en binario.

Ensamblador (\$DirDato en hexadecimal)	Descripción RT	Formato de la Instrucción en binario	
		CO	Dirección del Dato en binario
STOP	Fin ejecución	00	X X X X X
ADD \$DirDato	$AC \leftarrow AC + M(\$DirDato)$	01	$A_5 A_4 A_3 A_2 A_1 A_0$
SUB \$DirDato	$AC \leftarrow AC - M(\$DirDato)$	10	$A_5 A_4 A_3 A_2 A_1 A_0$
STA \$DirDato	$M(\$DirDato) \leftarrow AC$	11	$A_5 A_4 A_3 A_2 A_1 A_0$

Dada la **Tabla P.2**, donde se especifica un programa en ensamblador, realice lo siguiente:

- a) Completar las filas de la tabla , rellenando los códigos de operación y de dirección del dato (operando) en binario.
- b) Rellenar la última columna escribiendo las instrucciones en hexadecimal.
- c) Tras ejecutar el programa, ¿qué valor se almacena en la memoria en la dirección \$3D , sabiendo que antes de ejecutar el programa, la memoria tenía almacenado el dato 33 (en hexadecimal) en la dirección \$3B y el dato 03 (en hexadecimal) en la dirección \$3C ?

Programa en ensamblador (\$DirDato en hexadecimal)	Descripción RT del programa	Instrucción en binario		Instrucción en hexadecimal
		CO 2 bits	Dirección del dato en binario con 6 bits	
STA \$3A	$M(\$3A) \leftarrow AC$	11	11 1010	FA
SUB \$3A	$AC \leftarrow AC - M(\$3A)$			
ADD \$3B	$AC \leftarrow AC + M(\$3B)$			
ADD \$3C	$AC \leftarrow AC + M(\$3C)$			
STA \$3D	$M(\$3D) \leftarrow AC$			
STOP	Fin ejecución			

Tabla P.2

Apellidos :

Nombre :

Grupo :

D.N.I. :

EJERCICIOS:

1. **(1,00 pto.)** Suponga que un computador trabaja con datos enteros y con longitud de palabra $n = 8$ bits. Dados los datos de la columna de la izquierda en representación interna, indique su valor en decimal en la columna de la derecha. Para representación sesgada (el sesgo es $S = 2^{n-1}$, donde $n=8$ es el número de bits).

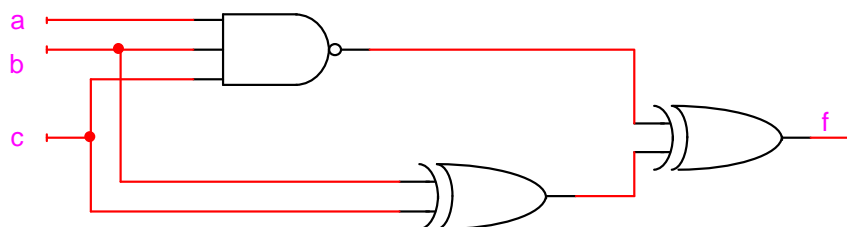
	Representación interna	Valor decimal que representa
(Signo Magnitud)	1000 0101	
(Complemento 1)	0000 0111	
(Complemento 2)	1111 1110	
(Sesgada)	1000 0100	

2. **(1,50 pto.)** Un procesador dispone, entre otros, de los siguientes elementos: Registro de Dirección (AR) de 32 bits, Registro de Datos (DR) de 16 bits y Contador de Programa (PC). El procesador funciona con un reloj de frecuencia 8 MHz y está conectado mediante el bus de datos con la memoria y para cada transferencia de un dato se requieren 3 ciclos de reloj.

Indicar:

- Número de bits del bus de datos (DB).
- Número de bits del bus de direcciones (AB).
- Tamaño en bits del registro Contador de Programa (PC).
- Tamaño máximo posible de la memoria principal (en MB o GB).
- Velocidad de transferencia de datos entre el procesador y la memoria.

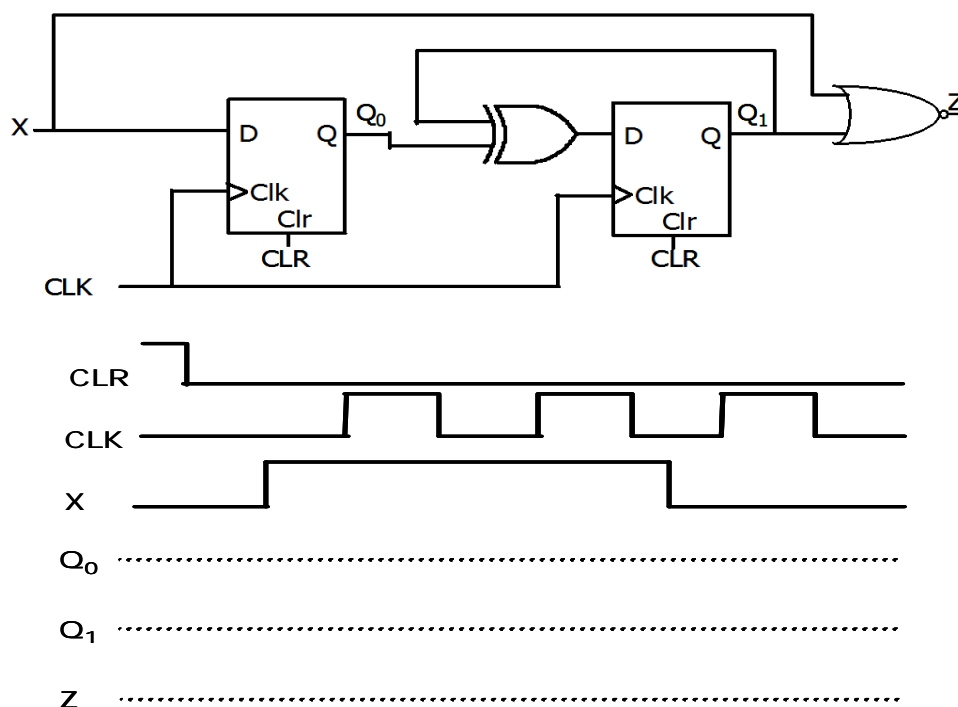
3. **(1,00 pto.)** Analice el circuito de la figura y obtenga razonadamente: a) la tabla de verdad de la función de conmutación $f(a,b,c)$, b) una realización equivalente del circuito con estructura AND/OR, y c) una realización utilizando un único multiplexor de tamaño adecuado.



4. **(1,00 pto.)** Diseñe un circuito combinacional con estructura NAND/NAND que implemente la multiplicación de dos números binarios enteros positivos de 2 bits, $X=(x_1 x_0)$ e $Y=(y_1 y_0)$, tal que genere la salida de 4 bits, $Z=(z_3 z_2 z_1 z_0)$, requeridas para representar el resultado de la multiplicación. Para ello, realice lo siguiente:

- Tabla de verdad y mapas de Karnaugh de las salidas.
- Minimización con mapas de Karnaugh de las funciones de salida.
- Dibujar el circuito resultante con estructura NAND/NAND.

5. (1,00 pto.) Complete el siguiente diagrama de tiempos para el circuito de la figura.



6. (1,00 pto.) Un sistema secuencial síncrono tiene dos entradas (X_1 y X_0), y una salida (Z). Su función es comparar las secuencias que recibe por ambas entradas. Si $X_1 = X_0$ durante **al menos** tres ciclos de reloj consecutivos, el circuito genera $Z=1$ a partir del tercer ciclo (**mientras** $X_1 = X_0$); en cualquier otro caso, produce $Z = 0$, tal como se refleja en el siguiente ejemplo:

$X_1 = 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ \dots$
 $X_0 = 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ \dots$
 $Z = 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ \dots$

Obtenga el *diagrama de estados* y la *tabla de estados* del sistema secuencial síncrono.

7. (1,00 pto.) Para la unidad de procesamiento de la figura:

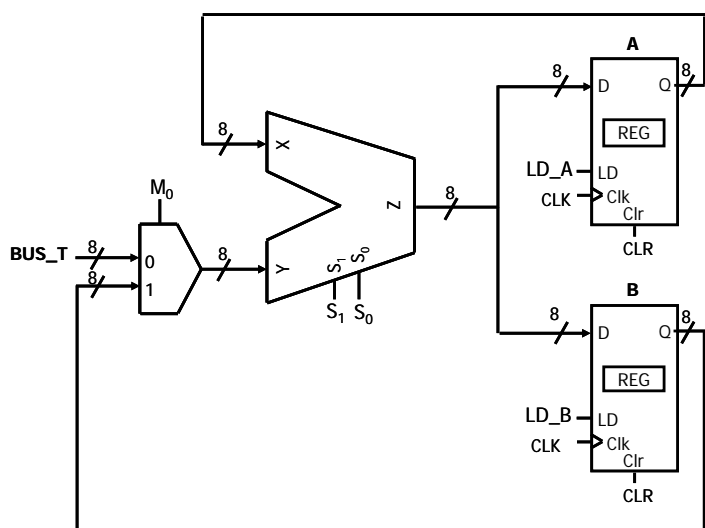


Tabla de Operaciones de la ALU

S1	S0	Z
0	0	$X \text{ MAS } Y$
0	1	Y
1	0	$\overline{X \cdot Y}$
1	1	$Y \text{ MAS } 1$

Complete la siguiente tabla indicando la operación RT que se realiza tras el flanco de subida de la señal de reloj. En la primera fila se ha proporcionado un ejemplo.

LD_A	LD_B	M0	S1	S0	Operación RT
1	0	1	1	1	$A \leftarrow B \text{ MAS } 1, B \text{ no cambia}$
1	0	1	0	1	
1	1	0	0	0	
0	1	1	1	0	
1	1	0	0	1	

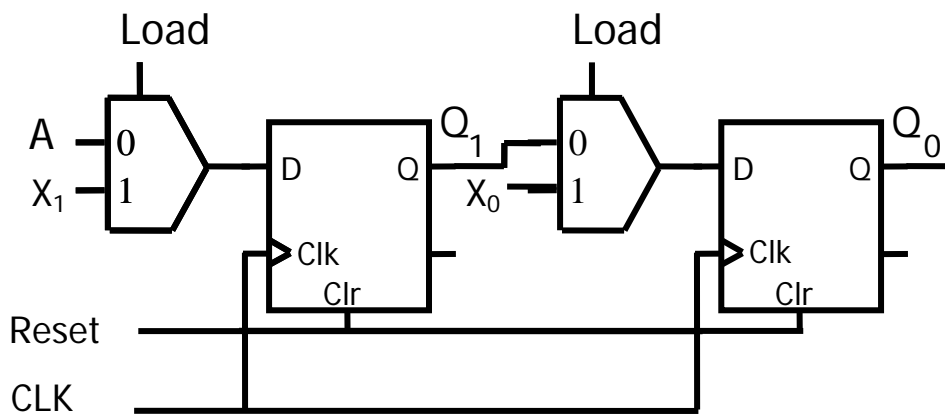


Apellidos : _____

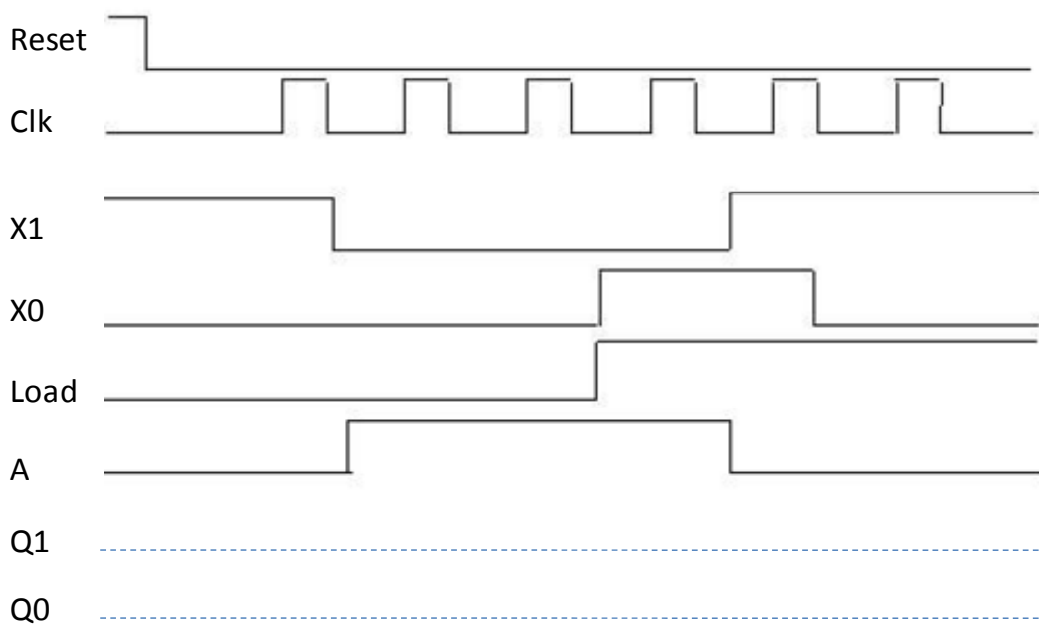
Nombre : _____ Grupo : _____

D.N.I. : _____

1. (0,50 pto.) En la figura se muestra un registro de desplazamiento con posibilidad de carga en paralelo. Para ello, se añaden circuitos que permiten configurar las conexiones de las entradas de los biestables D, bien para conectar en cascada los biestables (operación de desplazamiento) o bien para conectar entradas externas con las entradas D de los biestables (carga paralelo síncrona).



Para dicho circuito, complete el siguiente cronograma.



2. (0,50 pts.) En la tabla de la figura siguiente se indica el repertorio de las 4 instrucciones del computador simple CS1, indicando sus nombres en ensamblador, el resultado de su ejecución descrita a nivel de transferencia a registros (RT) y su formato en binario.

Ensamblador (\$DirDato en hexadecimal)	Descripción RT	Formato de la Instrucción en binario	
		CO	Dirección del Dato en binario
STOP	Fin ejecución	00	X X X X X X
ADD \$DirDato	$AC \leftarrow AC + M(\$DirDato)$	01	$A_5 A_4 A_3 A_2 A_1 A_0$
SUB \$DirDato	$AC \leftarrow AC - M(\$DirDato)$	10	$A_5 A_4 A_3 A_2 A_1 A_0$
STA \$DirDato	$M(\$DirDato) \leftarrow AC$	11	$A_5 A_4 A_3 A_2 A_1 A_0$

Tabla P2a

PROM Synthesizer							
E0	A0	7E	7D	BC	FF	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
0D	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
F8	00	00	00	03	0A	06	00

DIRECCIONES
DE
MEMORIA

→ 00 – 07
→ 08 – 0F
→ 10 – 17
→ 18 – 1F
→ 20 – 27
→ 28 – 2F
→ 30 – 37
→ 38 – 3F

Dada la **Tabla P2b** correspondiente al contenido inicial de la memoria RAM del CS1, donde se almacenan las instrucciones de un programa y datos, ambos en formato hexadecimal, junto con una columna que indica el rango de direcciones de memoria en hexadecimal, correspondiente a cada fila. Realice lo siguiente:

- a) Copiar la notación en hexadecimal de las instrucciones del programa almacenado en memoria (de la dirección 0 a la 6, es decir la primera fila de la PROM, Tabla P2b) en la última columna de la **Tabla P2c**. A partir de esta información completar el resto de la **Tabla P2c**, indicando para cada instrucción: 1) su notación en ensamblador, 2) su descripción RT, 3) su notación en binario.

Tabla P2b

- b) Sabiendo que antes de ejecutar el programa, el contenido de la memoria es el de la **Tabla P2b** y que el acumulador **AC** contiene el valor **FF** en hexadecimal. Indicar los datos en hexadecimal que se verían en la memoria RAM, correspondientes a las direcciones de memoria: \$20, \$38, \$3C, \$3D, \$3E, \$3F, después de ejecutar el programa.

Programa en ensamblador (\$DirDato en hexadecimal)	Descripción RT del programa	Instrucción en binario		Instrucción en hexadecimal
		CO 2 bits	Dirección del dato en binario con 6 bits	
STA \$20	$M(\$20) \leftarrow AC$	11	10 0000	E0

Tabla P2c