



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 Sesiada (para el dato -1, siendo el sesgo de $S = 2^{n-1}$).

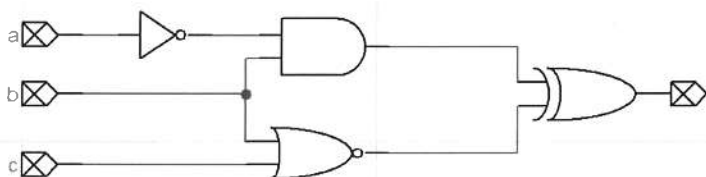
	Representación interna	Valor decimal que representa
(Signo Magnitud)		- 9
(Complemento 1)		- 1
(Complemento 2)		+9
(Sesiada)		- 1

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.

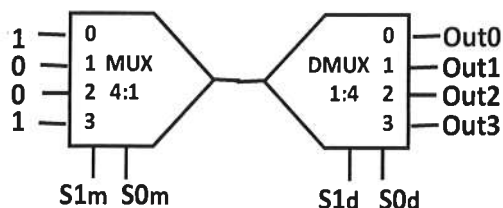


Figura 2a

Señales de control del MUX		Señales de control del DMUX		Salidas del DMUX			
$S1m$	$S0m$	$S1d$	$S0d$	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				

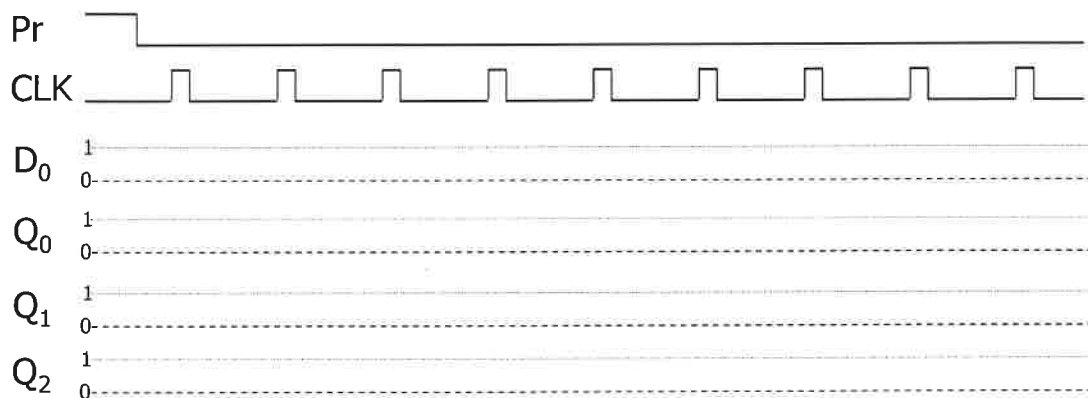
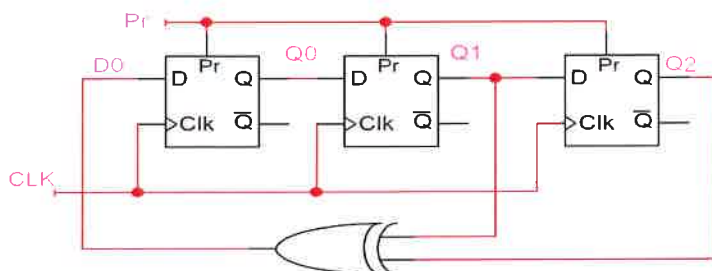
Figura2b

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.



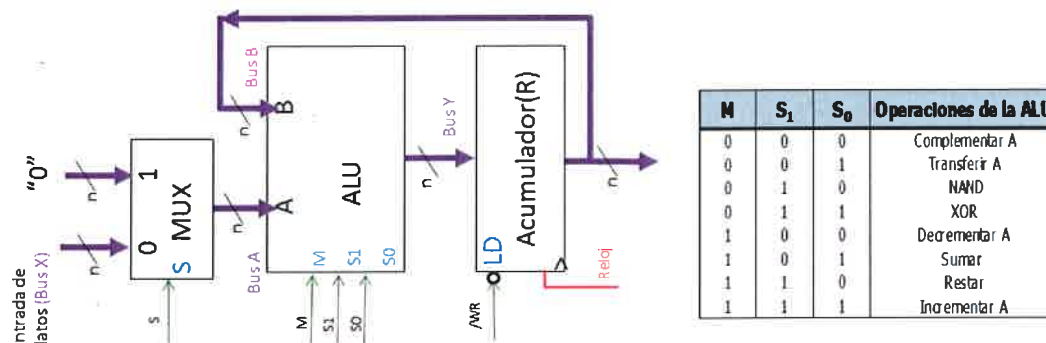
7. (1 pto.) Diseñe un generador de secuencia síncrono, con una señal de control M, tal que si M=0 la secuencia sea ascendente (1,4,4,7...), y si M=1 la secuencia sea descendente (7,4,4,1...). Utilice el tipo de biestable que prefiera.

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'.



Operaciones RT	PALABRA DE CONTROL (Señales de control)					(En hexadecimal)
	Selección de entrada		Controles de la ALU		Escritura en acumulador R	
	S	M	S1	S0	/WR	
R ← -X + R	0	1	0	1	0	0A
R ← -X + 1						
R ← -X						
R ← -X XOR R						
R ← -X - 1						



ugr

Universidad
de Granada

DEPARTAMENTO DE ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES

TECNOLOGÍA Y ORGANIZACIÓN
DE
COMPUTADORES
1º Grado en Ingeniería Informática.

GRANADA, 1 de Julio de 2013
EXAMEN DE PRÁCTICAS.

Apellidos :

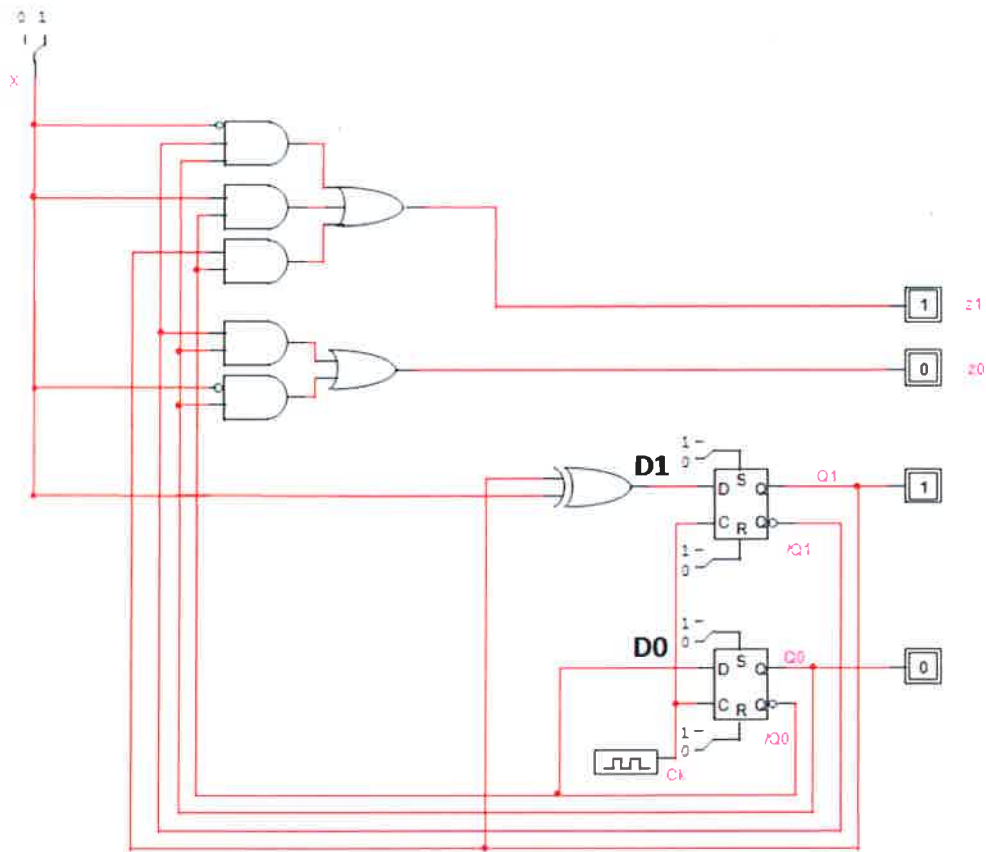
Nombre :

Grupo :

D.N.I. :

PRACTICAS: (1 punto)

1. (0,50 pto.) Análisis de un sistema secuencial. ¿Es tipo Mealy o tipo Moore?



Rellena la siguiente Tabla de Excitación del Sistema:

X Q ₁ Q ₀	D ₁	D ₀
0 0 0		
0 0 1		
0 1 0		
0 1 1		
1 0 0		
1 0 1		
1 1 0		
1 1 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

1º)

	Representación interna	Valor decimal que representa
(Signo Magnitud)	1000 1001	- 9
(Complemento 1)	1111 1110	- 1
(Complemento 2)	0000 1001	+9
(Sesgada)	0111 1111	- 1

$$-9)_{SH} = 1\ 000\ 1001$$

$$-1)_{CD} \Rightarrow 1)_{10} = 0000\ 0001 \Rightarrow C_1(1) = 1111\ 1110$$

$$+9)_{C2} \Rightarrow 9)_{10} = 0000\ 1001$$

$$S = 128 = 2^7 \Rightarrow -1 + 128 = 127)_{10} = 0111\ 1111$$

2º)

a) NCiclos:

$$ADD \rightarrow 2 + 3 = 5 \text{ ciclos}$$

$$SUB \rightarrow 2 + 3 = 5 \text{ ciclos}$$

$$STOP \rightarrow 2 + 1 = 3 \text{ ciclos}$$

$$STA \rightarrow 2 + 2 = 4 \text{ ciclos}$$

$$NCiclos = \underbrace{(24 \times 5)}_{ADD} + \underbrace{(26 \times 5)}_{SUB} + \underbrace{(10 \times 4)}_{STA} + \underbrace{(1 \times 3)}_{STOP} =$$

$$= 120 + 130 + 40 + 3 = 293 \text{ ciclos}$$

$$b) \text{ Si } f = 50 \text{ MHz} = 50 \times 10^6 \text{ Hz} \Rightarrow T = 20 \text{ ns}$$

$$t = 293 \text{ ciclos} \times 20 \text{ ns} = 5,860 \mu\text{s}$$

$$c) \text{ Una regla de tres: Hace } (24 + 26 + 10 + 1) \text{ Instr} = 61$$

$$\text{Si en } 5,860 \times 10^{-6} \text{ s} \rightarrow 61 \text{ Instr}$$

$$1 \text{ s} \rightarrow x \text{ Instr}$$

$$x = 10,41 \times 10^6 \text{ INSTR/s} = 10,41 \text{ MIPS}$$

$$d) 2^6 \text{ DIRECCIONES} = 64 \text{ direcciones de 8 bits} = 64 \text{ Bytes}$$

7^a) a) $Z_{H\bar{A}T} = 7_{10} = 111_2 \Rightarrow$ Requiere 3 salidas
 $Z_2 Z_1 Z_0$

b) $N = 4$ estados $\Rightarrow P = 2$ bits tablas (Q_1, Q_0)

Tabla transición: Tipo Moore

H	Q_1^t, Q_0^t	Q_1^{t+1}, Q_0^{t+1}	Z_2, Z_1, Z_0	D_1, D_0	T_1, T_0
0	00	01	001	01	01
0	01	10	100	10	11
0	10	11	100	11	01
0	11	00	111	00	11
1	00	01	001	11	11
1	01	00	100	00	01
1	10	01	100	01	11
1	11	10	111	10	01

$$Z_2 = Q_1 + Q_0$$

$$Z_1 = Q_1 \cdot Q_0$$

$$Z_0 = \overline{Q_1 \oplus Q_0}$$

$$D_1 = H \oplus Q_1 \oplus Q_0$$

$$D_0 = \overline{Q_0}$$

$$T_1 = (\bar{H} \cdot Q_0) + (H \cdot \bar{Q}_0)$$

$$T_0 = 1$$

$H \backslash Q_1, Q_0$	00	01	11	10
0		1		1
1	1		1	

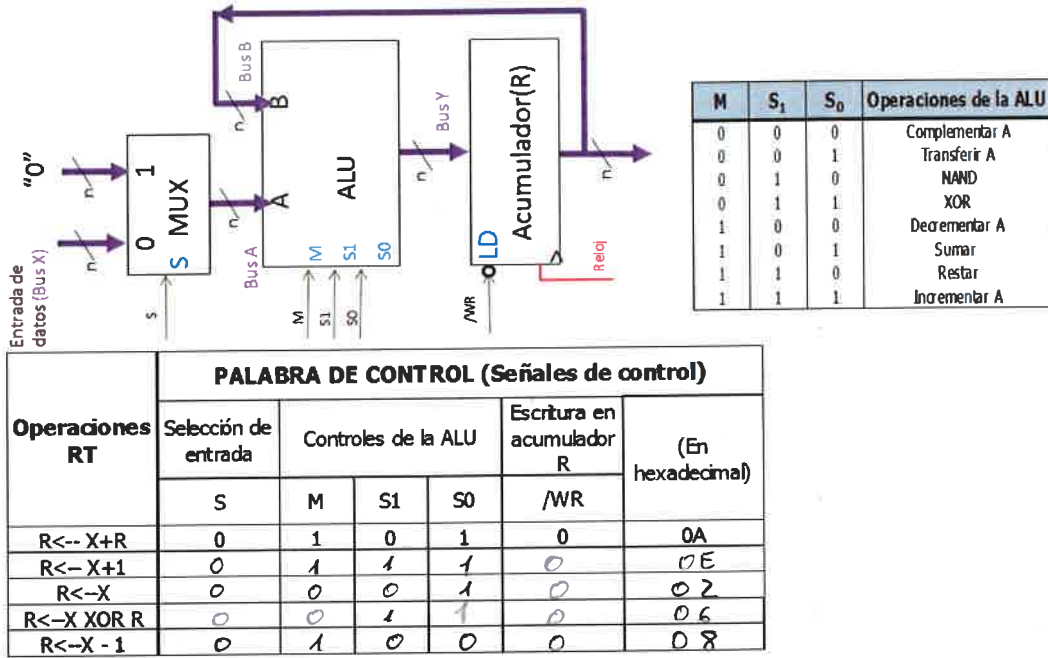
D_1

$H \backslash Q_1, Q_0$	00	01	11	10
0		1	1	
1	1			1

T_1

8. (0,5 pts) 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 $X=8$.



b) Si $X = 8$ y $15 = 8 + 7$

b.1) $R \leftarrow X - 1 = 7$;

S	M	S ₁	S ₀	\overline{WR}
0	1	0	0	0

b.2) $R \leftarrow X + 7 = 8 + 7 = 15$;

S	M	S ₁	S ₀	\overline{WR}
0	1	0	1	0



ugr

Universidad
de Granada

DEPARTAMENTO DE ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES

TECNOLOGÍA Y ORGANIZACIÓN DE
COMPUTADORES
1º Grado en Ingeniería Informática.

GRANADA, 1 de Julio de 2013
EXAMEN DE PRÁCTICAS.

Apellidos :

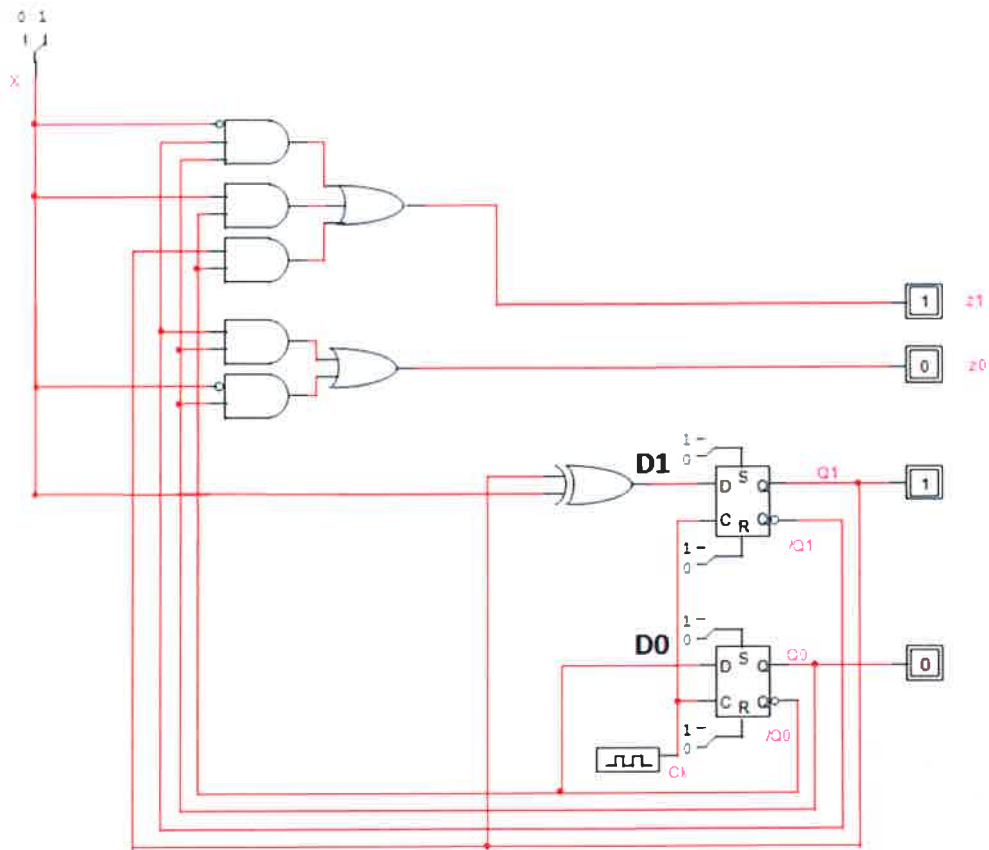
Nombre :

Grupo :

D.N.I. :

PRACTICAS: (1 punto)

1. (0,50 pto.) Análisis de un sistema secuencial. ¿Es tipo Mealy o tipo Moore?



Rellena la siguiente Tabla de Excitación del Sistema:

TIPO MEALY

$$D_1 = Q_1 \oplus X$$

$$D_0 = \bar{Q}_0$$

X Q ₁ Q ₀	D ₁	D ₀
0 0 0	0	1
0 0 1	0	0
0 1 0	1	1
0 1 1	1	0
1 0 0	1	1
1 0 1	1	0
1 1 0	0	1
1 1 1	0	0

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	$AC \leftarrow AC - M(\$3A)$	10	11 1010	BA
ADD \$3B	$AC \leftarrow AC + M(\$3B)$	01	11 1011	7B
ADD \$3C	$AC \leftarrow AC + M(\$3C)$	01	11 1100	7C
STA \$3D	$M(\$3D) \leftarrow AC$	11	11 1101	FD
STOP	Fin ejecución	00	XX XXXX	00

Tabla P.2

$STA \$3A \Rightarrow M(\$3A) \leftarrow AC$
 $SUB \$3A \Rightarrow AC \leftarrow AC - M(\$3A) = AC - AC = 0$
 $ADD \$3B \Rightarrow AC \leftarrow AC + M(\$3B) = 0 + 33 = 33$
 $ADD \$3C \Rightarrow AC \leftarrow AC + M(\$3C) = 33 + 03 = 36$
 $STA \$3D \Rightarrow M(\$3D) \leftarrow AC = 36$
 EN \$3D queda 36