

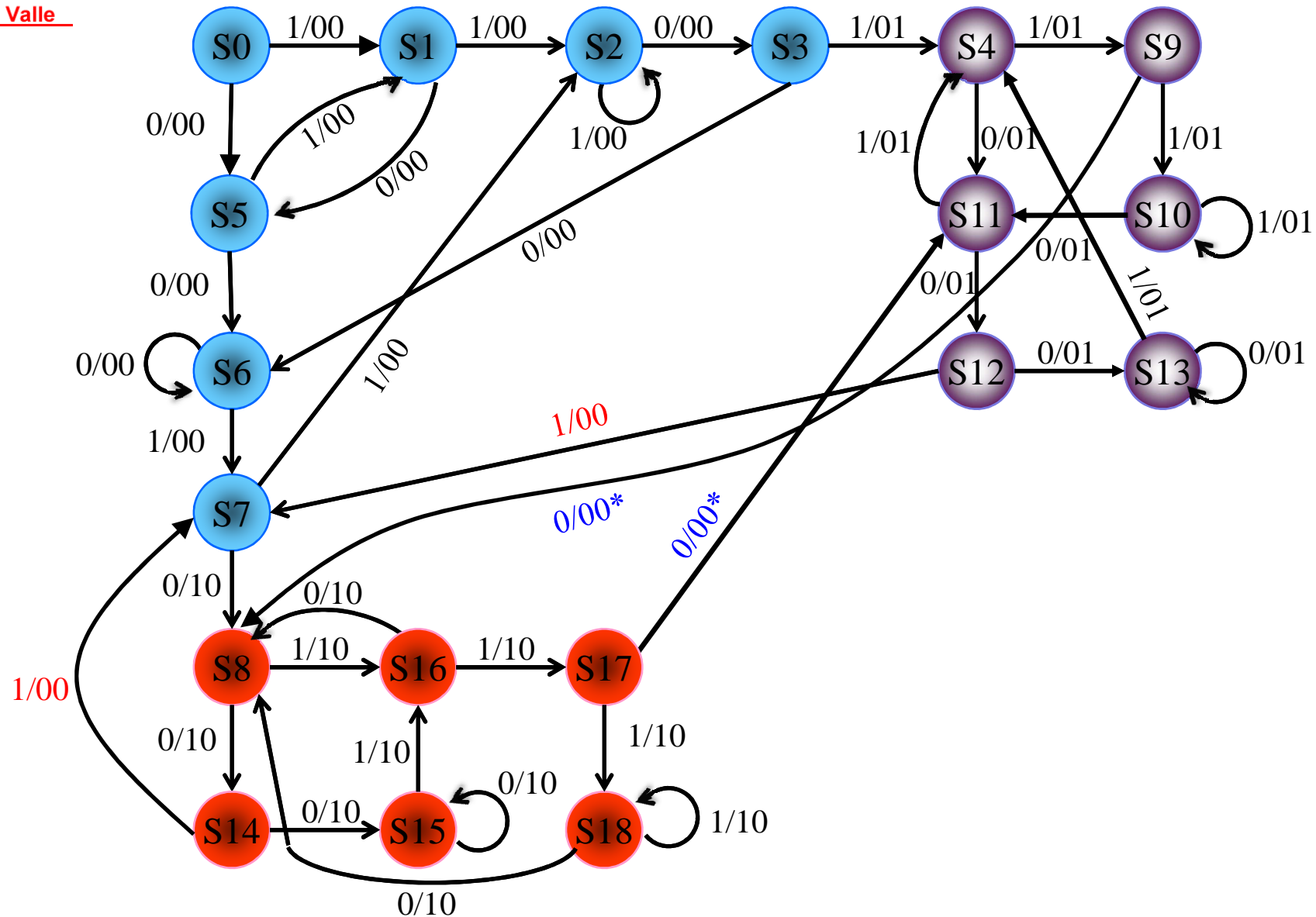
*ANDRES FERNANDO
RESTREPO ALVAREZ*

COD: 0511013

Diseño FSM para controlar motor DC

- Diseñar una FSM síncrona para controlar un motor DC, existe superposición.
- El motor arranca y gira en sentido normal cuando detecta la secuencia 1101.
- El motor arranca y gira en sentido contrario cuando detecta la secuencia 0010
- El motor se detiene, si el circuito detecta la secuencia 1001 y permanece detenido hasta detectar de nuevo la secuencia de arranque.
- Sin embargo, después de arrancar el motor, cada vez que el circuito detecta la secuencia 0110 el motor debe girar en sentido contrario, pero primero debe parar antes de cambiar de giro.
- Usar FSM tipo Mealy.

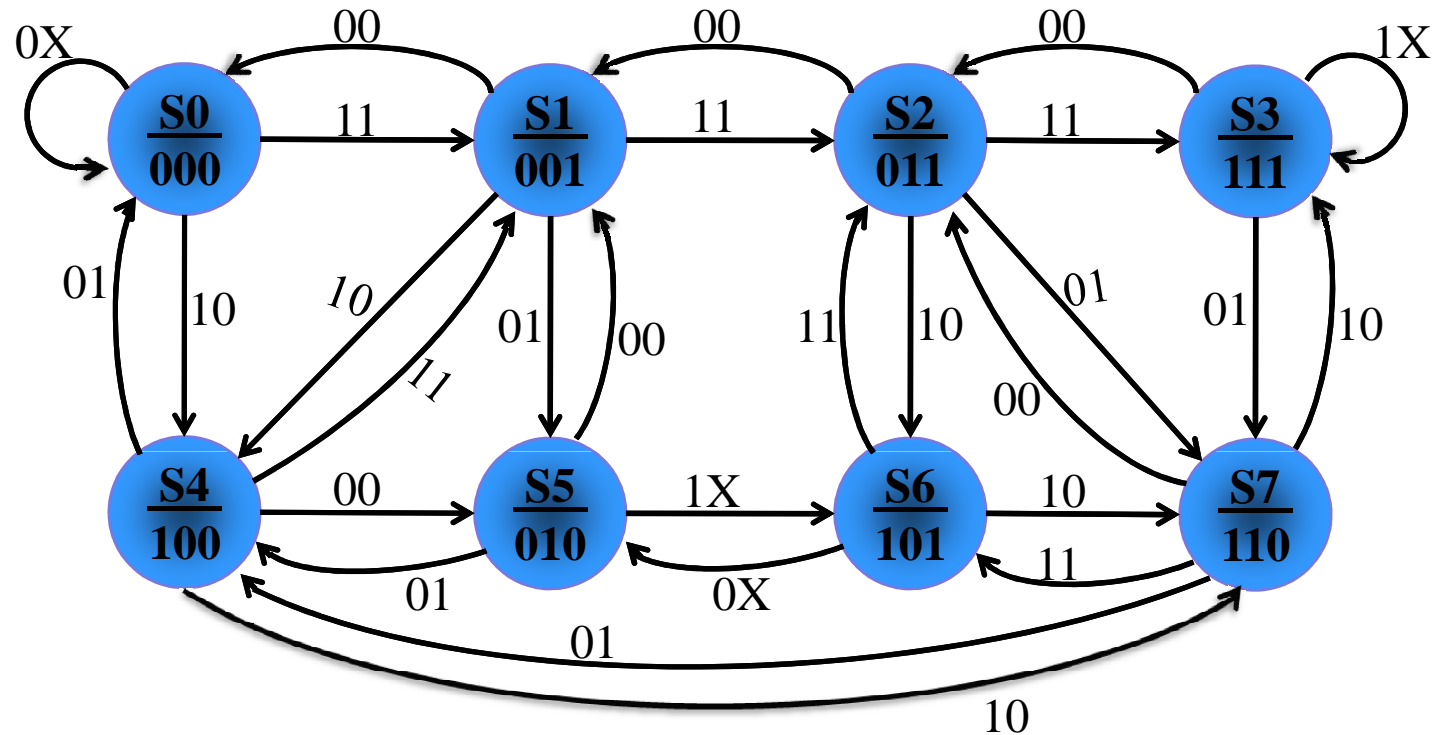
Diagrama de estados



Diseño FSM Registro de desplazamiento de 3 bits

- Diseñar una FSM que permita implementar un registro de desplazamiento de 3 bits, el registro debe realizar el desplazamiento a la izquierda y a la derecha.
- La FSM debe ser tipo Moore
- Usar flip-flop D y codificación directa.

Diagrama de estados



Entradas

- 00 Desplaza hacia la derecha con 0
- 01 Desplaza hacia la izquierda con 0
- 10 Desplaza hacia la derecha con 1
- 11 Desplaza hacia la izquierda con 1

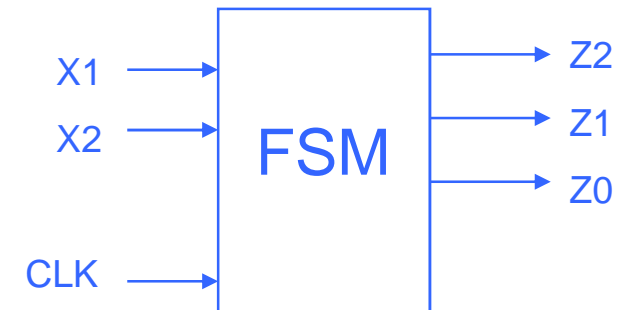
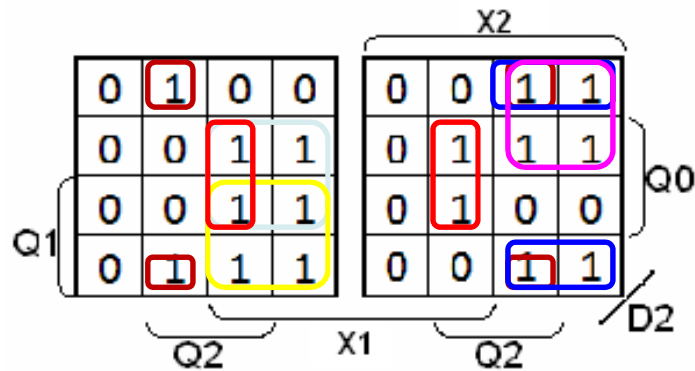


Tabla de estados

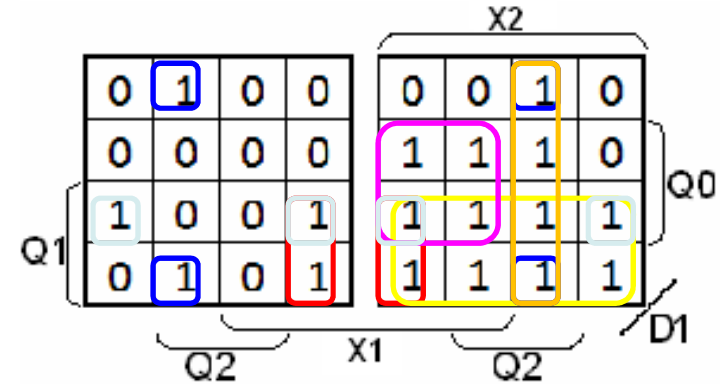
IN		E.P			P.E			FF - D			OUT		
X2	X1	Q2	Q1	Q0	Q2 ⁺	Q1 ⁺	Q0 ⁺	D2	D1	D0	Z2	Z1	Z0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0	0	1	0	1	1
0	0	0	1	1	0	1	0	0	1	0	1	1	1
0	0	1	0	0	1	0	1	1	0	1	1	0	0
0	0	1	0	1	0	0	1	0	0	1	0	1	0
0	0	1	1	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	0	1	0	0	1	0	1	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1	0	1	0	0	1
0	1	0	1	0	1	1	1	1	1	1	0	1	1
0	1	0	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	0	0	0	0	0	0	0	1	0	0
0	1	1	0	1	1	0	0	1	0	0	0	1	0
0	1	1	1	0	1	0	1	1	0	1	1	0	1
0	1	1	1	1	1	0	0	1	0	0	1	1	0

IN		E.P			P.E			FF - D			OUT		
X2	X1	Q2	Q1	Q0	Q2 ⁺	Q1 ⁺	Q0 ⁺	D2	D1	D0	Z2	Z1	Z0
1	0	0	0	0	1	0	0	1	0	0	0	0	0
1	0	0	0	1	1	0	0	1	0	0	0	0	1
1	0	0	1	0	1	1	0	1	1	0	0	1	1
1	0	0	1	1	0	1	1	0	1	1	1	1	1
1	0	1	0	0	1	1	1	1	1	1	1	0	0
1	0	1	0	1	1	1	0	1	1	0	0	1	0
1	0	1	1	0	1	1	1	1	1	1	1	0	1
1	0	1	1	1	0	1	1	0	1	1	1	1	0
1	1	0	0	0	0	0	1	0	0	1	0	0	0
1	1	0	0	1	0	1	0	0	1	0	0	0	1
1	1	0	1	0	0	1	1	0	1	1	0	1	1
1	1	0	1	1	0	1	1	0	1	1	1	1	1
1	1	1	0	0	0	0	1	0	0	1	1	0	0
1	1	1	0	1	1	1	0	1	1	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0	1
1	1	1	1	1	1	1	0	1	1	0	1	1	0

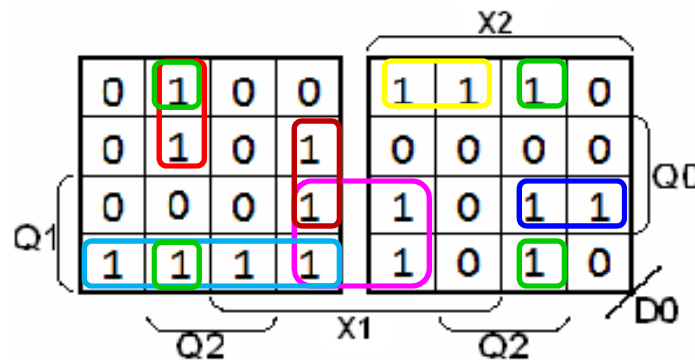
Mapas de Karnaugh



$$D2 = (X1' * Q2 * Q0') + (X2 * X1' * Q0') + (X2 * X1' * Q1') \\ + (X2' * X1 * Q0) + (X2' * X1 * Q1) + (X1 * Q2 * Q0)$$



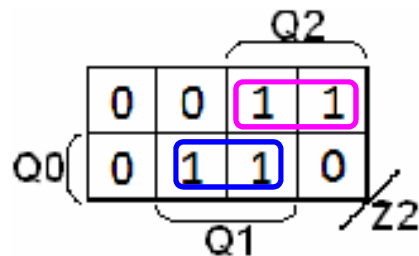
$$D1 = (X1 * Q2' * Q1) + (Q2' * Q1 * Q0) + (X2 * Q1) \\ + (X2 * X1' * Q2) + (X2 * X1 * Q0)$$



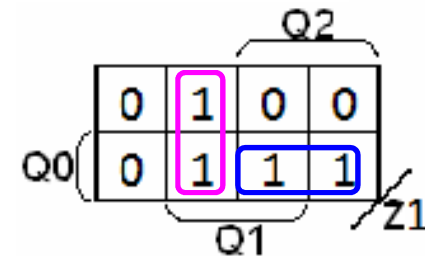
$$D0 = (X2' * Q1 * Q0') + (X2' * X1' * Q2 * Q1') + \\ (X2' * X1 * Q2' * Q0) + (X1 * Q2' * Q1) + (X1' * Q2 * Q0') \\ + (X2 * X1' * Q1 * Q0) + (X2 * X1 * Q1' * Q0')$$

Mapas de Karnaugh

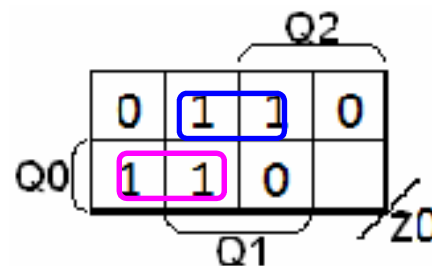
lógica de salida



$$Z2 = (Q1 * Q0) + (Q2 * Q0')$$

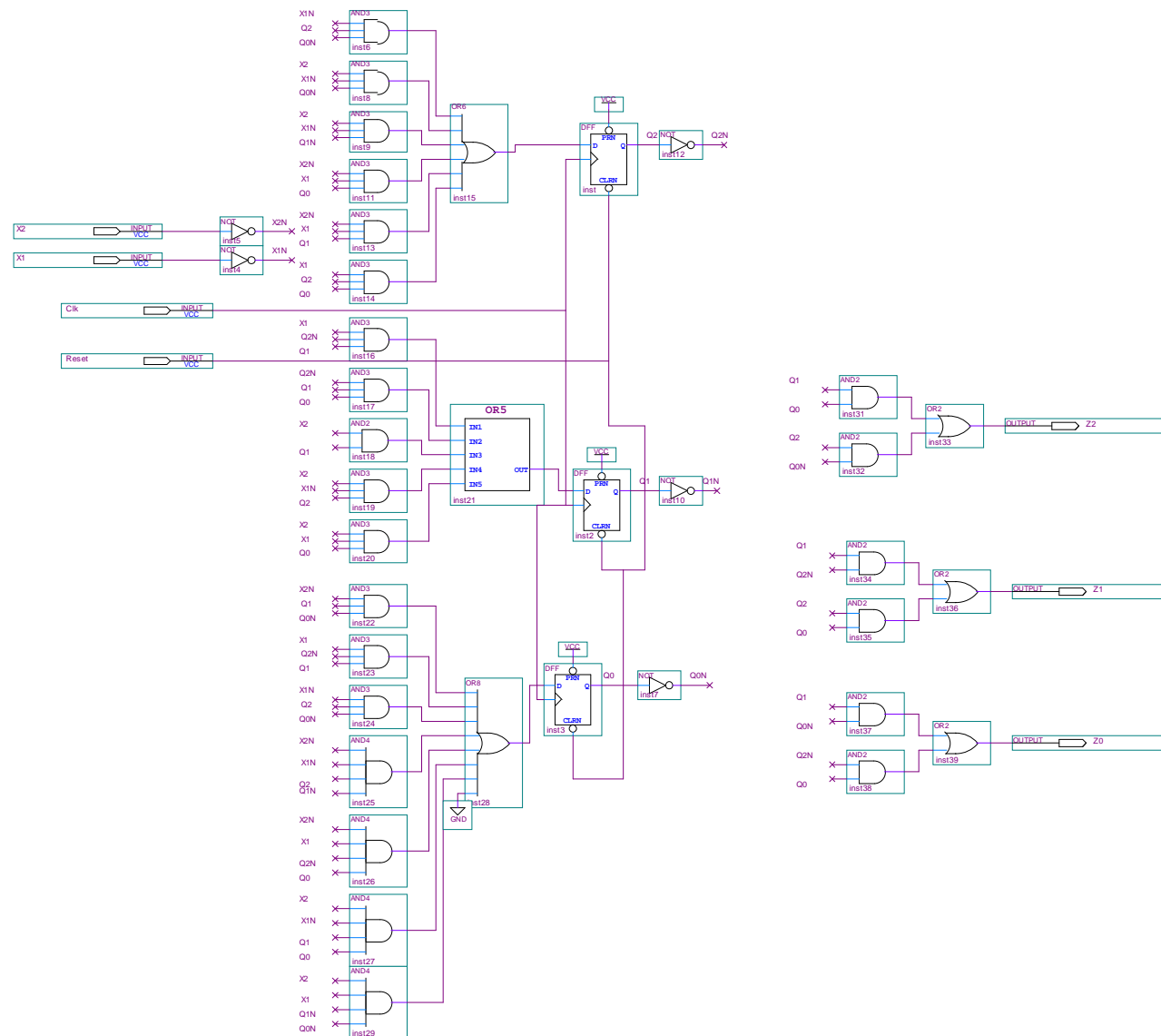


$$Z1 = (Q2' * Q1) + (Q2 * Q0)$$

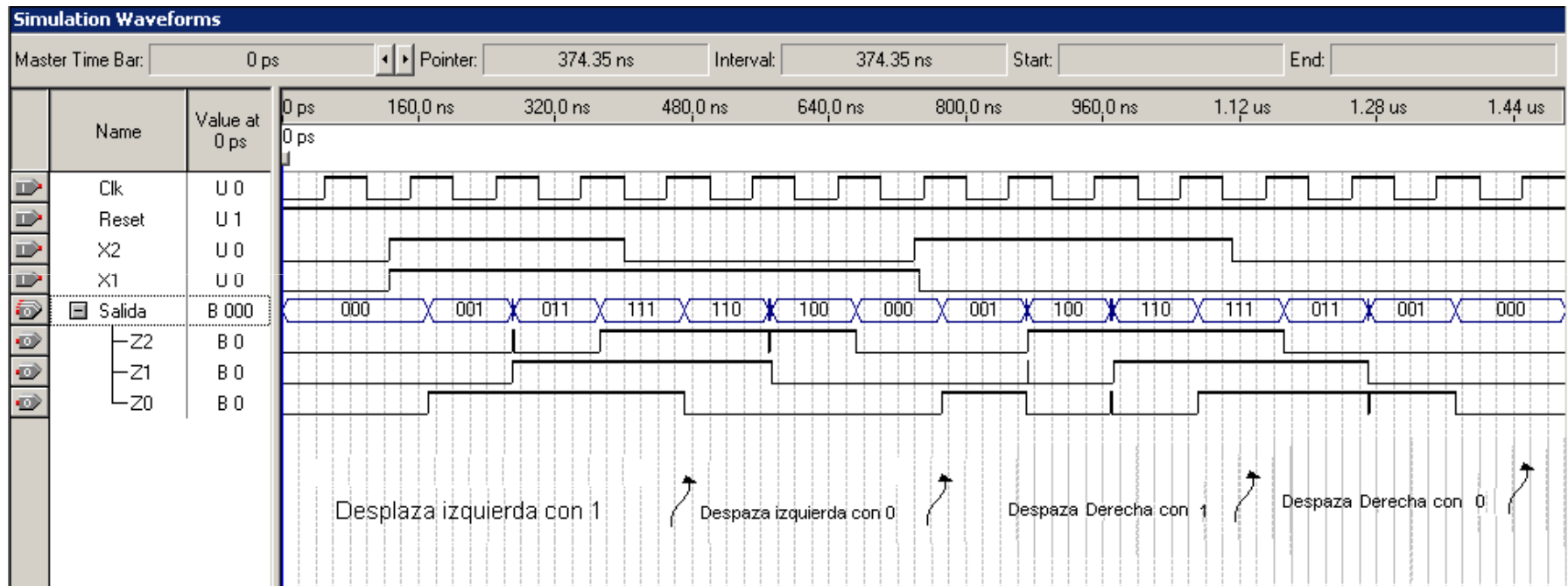


$$Z0 = (Q1 * Q0') + (Q2' * Q0)$$

Implementación en Quartus II



Simulación en Quartus II

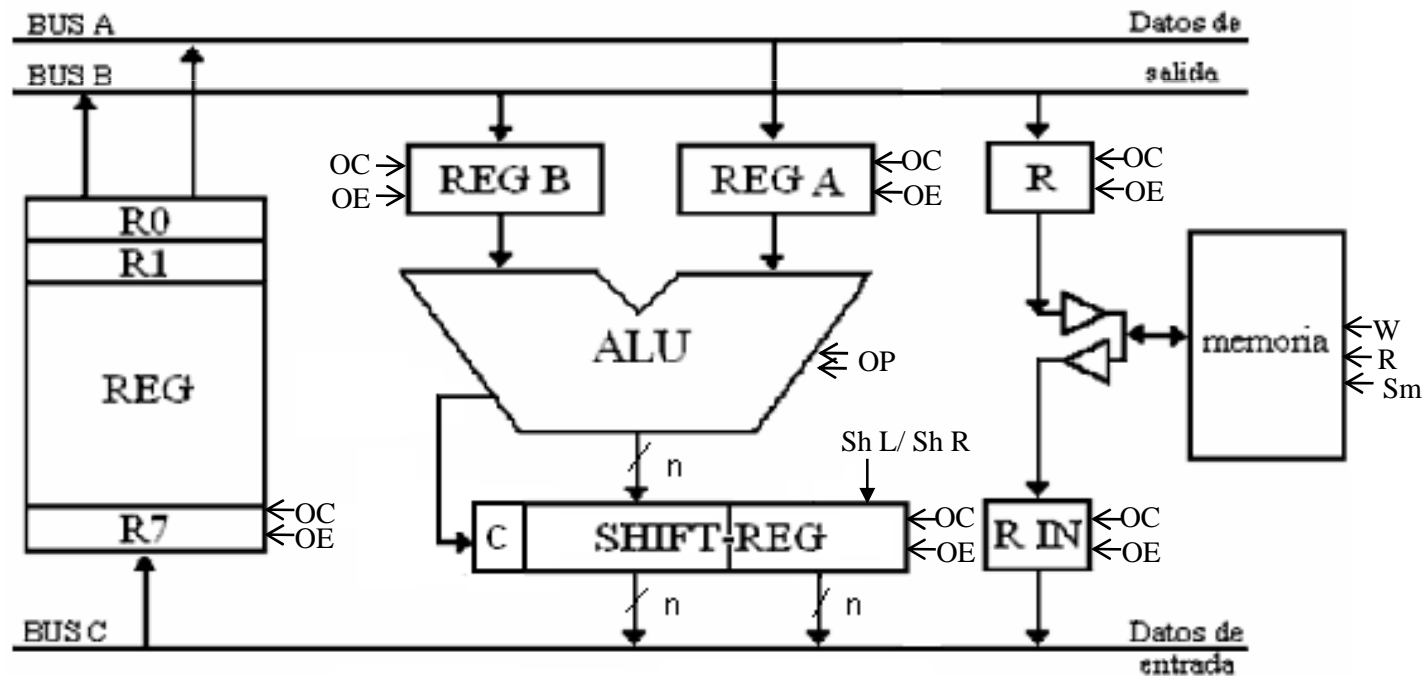


Diseño circuito controlador

➤ Diseñar un circuito controlador para realizar las operaciones en el data-path de la figura. Diagrama ASM, estados y verificar el diseño ($R3=1010$ y $R4=0011$).

$R1-R2 \leftarrow R3 * R4$ (usar R7 como registro auxiliar)

$M[100] - M[101] \leftarrow R3 / R4$



Consideraciones generales:

Shift-reg = (Shift-reg_high, Shift-reg_low)

Shift-reg = (Qh, Ql)

Cshift= contador de desplazamientos

Count= contador de numero de ciclos

Operaciones de la ALU: OP=00 Suma, OP=01 transparente RegA,
OP=10 transparente RegB, OP=11 Resta.

Para la multiplicación:

Qlsb = bit menos significativo de
Shif-reg

R3 = multiplicador

R4 = multiplicando

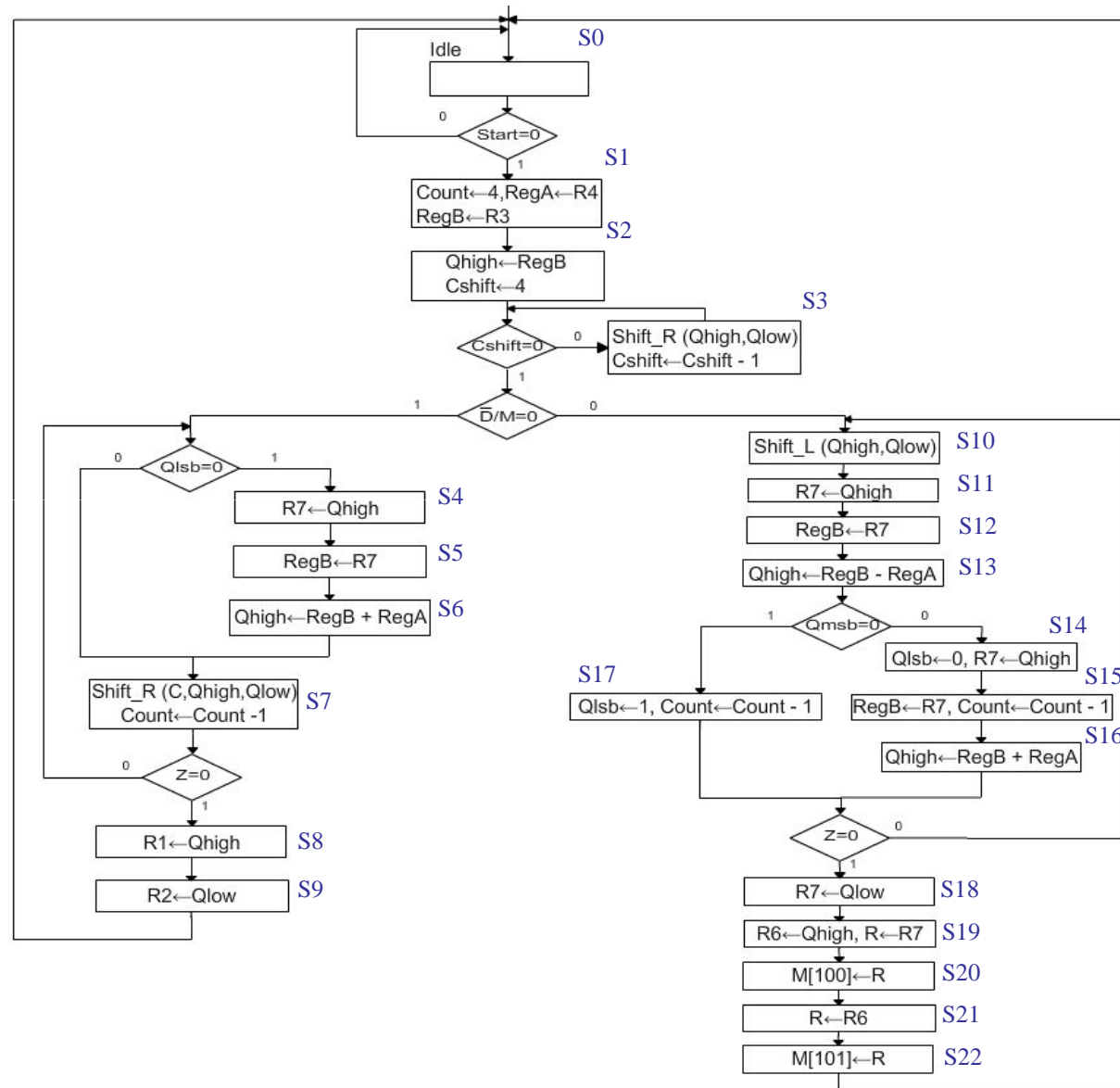
Para la división:

Qmsb=Cout de la ALU

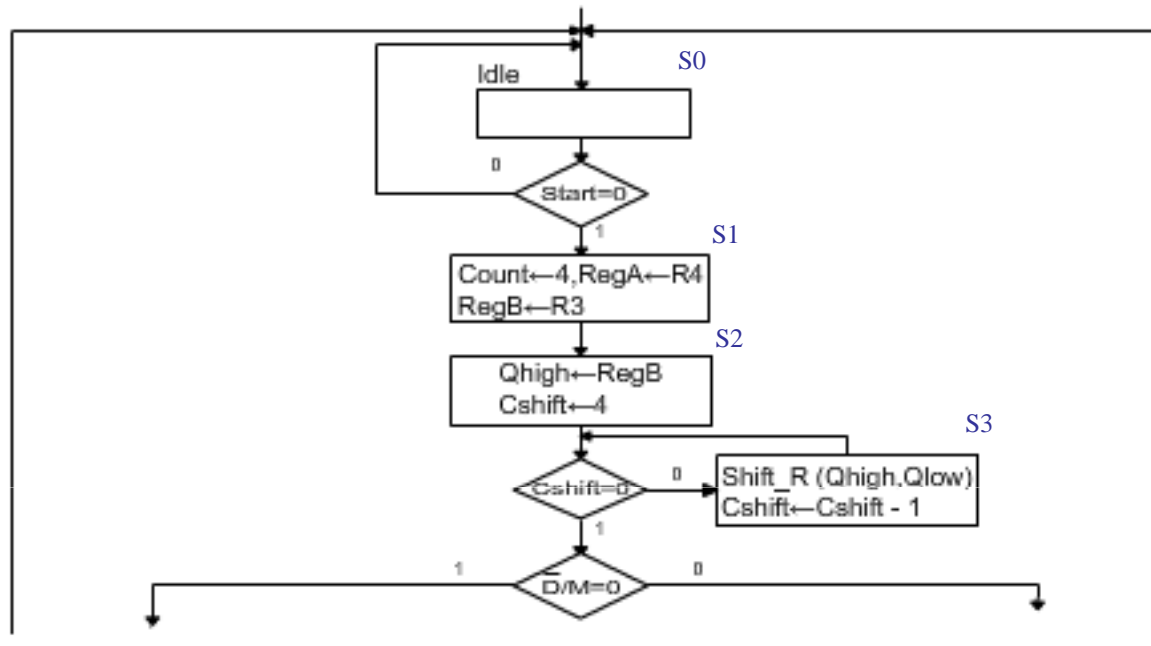
R3 = dividendo

R4 = divisor

Diagrama ASM



Prueba de funcionamiento



R3←1010 , R4←0011

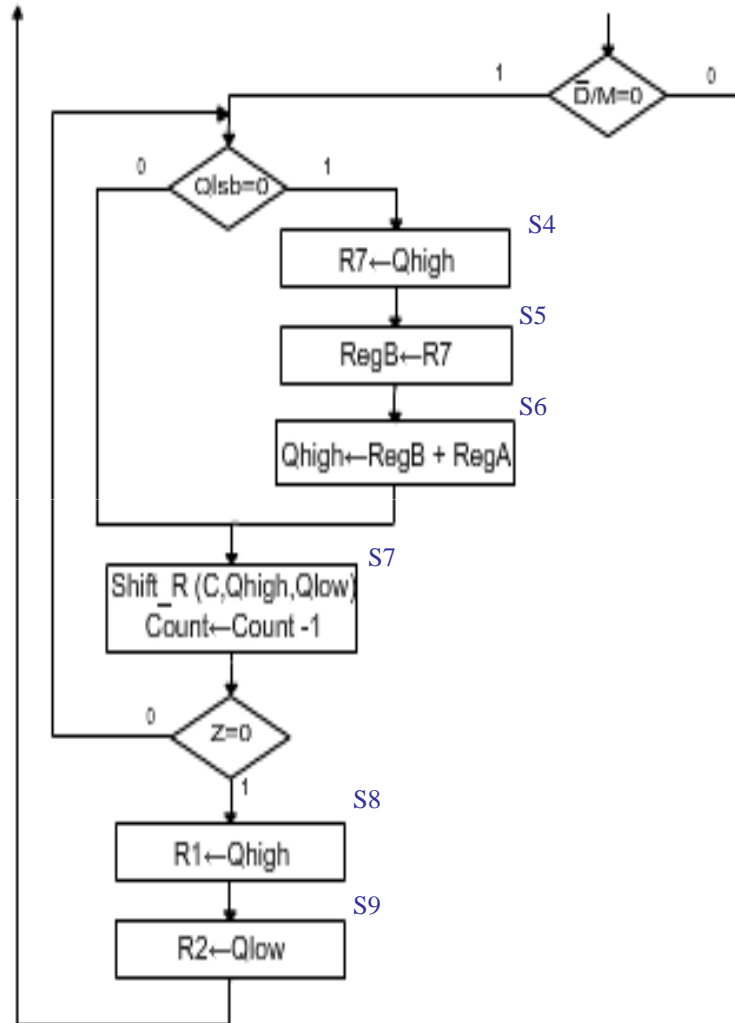
S1: OP←10, load Count, OER3,OER4,
OCRegA, OCRegB

S2: load Cshift, OERegB, OCQhigh

S3: Shift_R, Cshift←Cshift - 1

	C	Qhigh	Qlow
S2→	0	1010	0000
S3→	0	0101	0000
S3→	0	0010	1000
S3→	0	0001	0100
S3→	0	0000	1010

Prueba de multiplicación



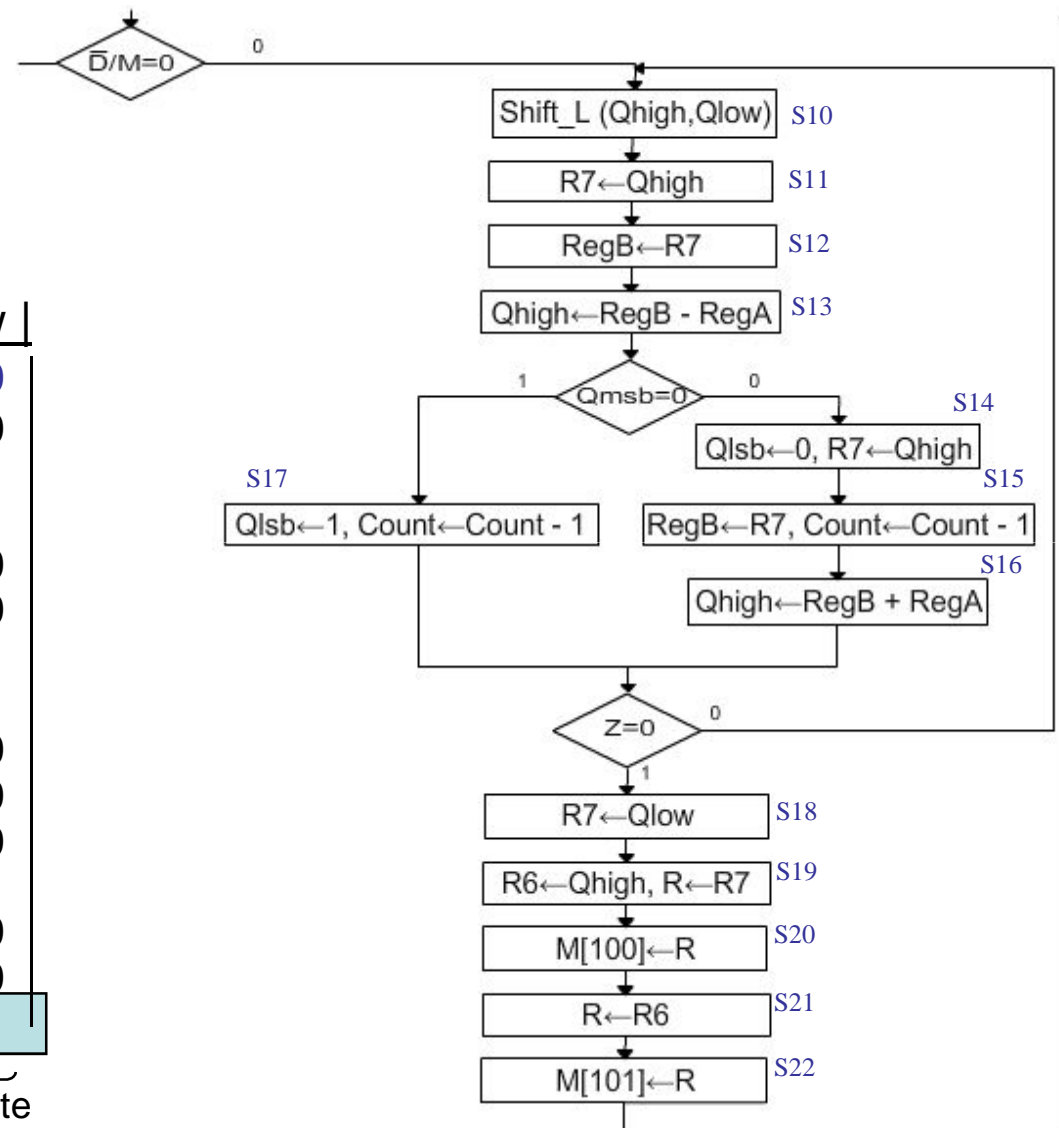
RegA ← 0011

	C	Qhigh	Qlow
	0	0000	1010
S7 →	0	0000	0101
S4, S5, S6 →	0	0011	0101
S7 →	0	0001	1010
S7 →	0	0000	1101
S4, S5, S6 →	0	0011	1101
S7 →	0	0001	1110

Prueba de división

RegA ← 0011

	C	Qhigh	Qlow
	0	0000	1010
S10 →	0	0001	0100
S11,S12,S13 →	1	1110	0100
S14 →	1	1110	0100
S15,S16 →	0	0001	0100
S10 →	0	0010	1000
S11,S12,S13 →	1	1111	1000
S14 →	1	1111	1000
S15,S16 →	0	0010	1000
S10 →	0	0101	0000
S11,S12,S13 →	0	0010	0000
S17 →	0	0010	0001
S10 →	0	0100	0010
S11,S12,S13 →	0	0001	0010
S17 →	0	0001	0011
		Residuo	Cociente



*ANDRES FERNANDO
RESTREPO ALVAREZ*

COD: 0511013