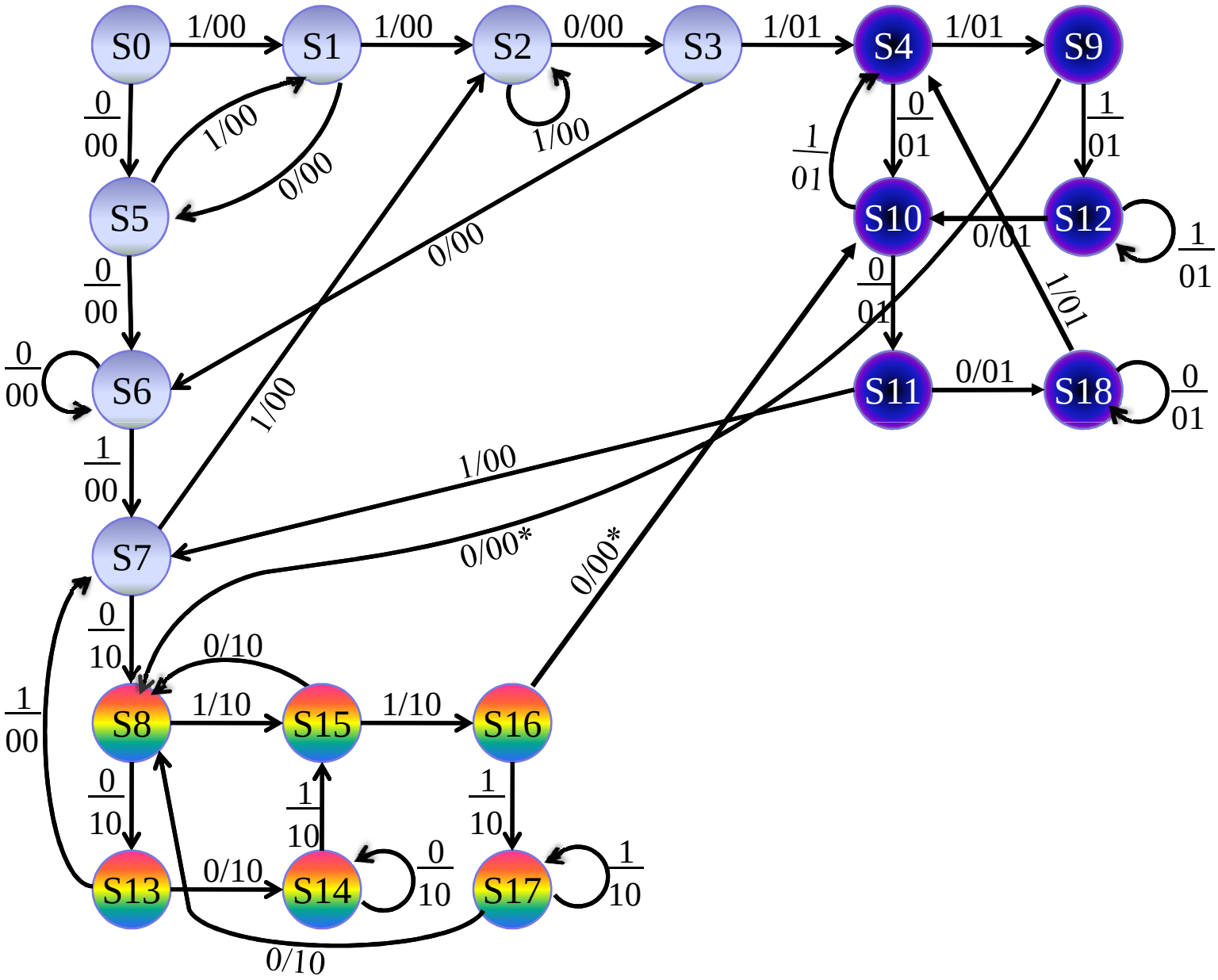
Digital System Design Course

FSM síncrona para controlar un 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.

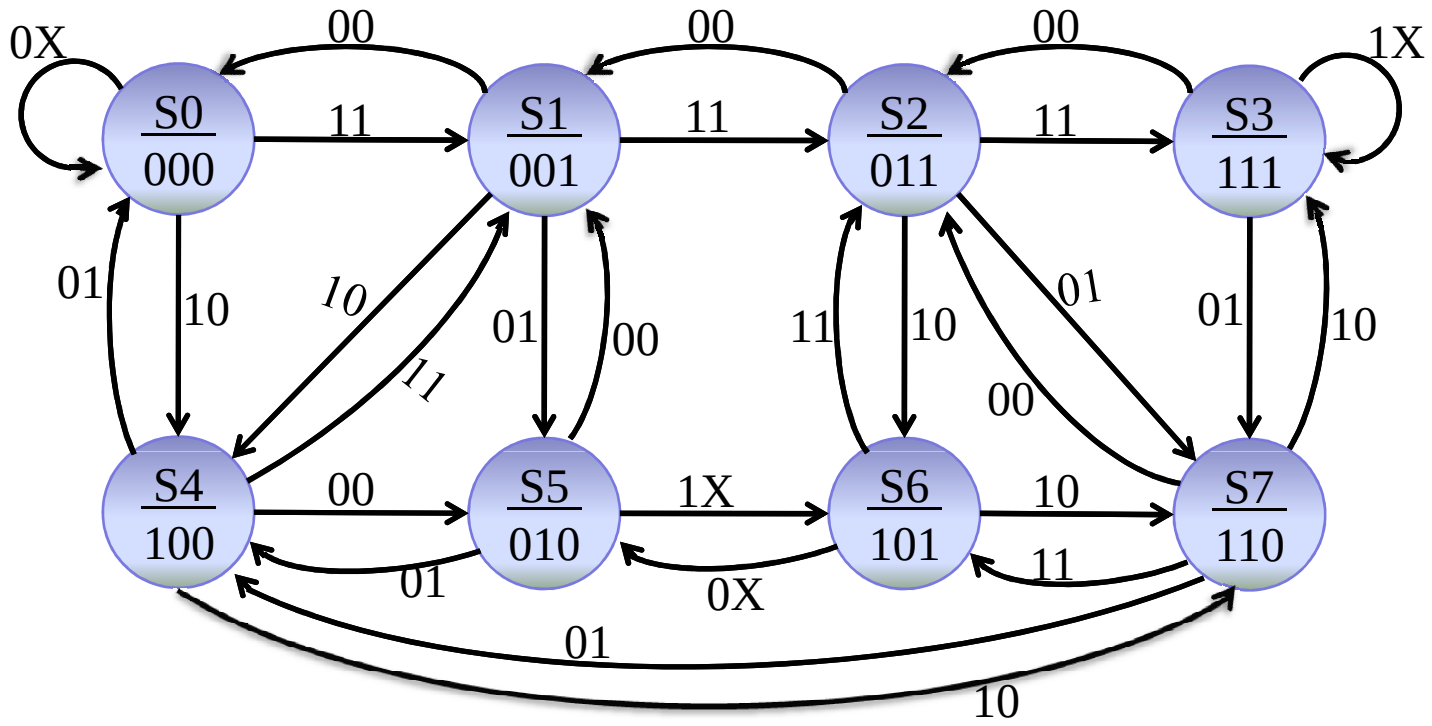
Diagrama de estados.



FSM Registro de desplazamiento

- 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



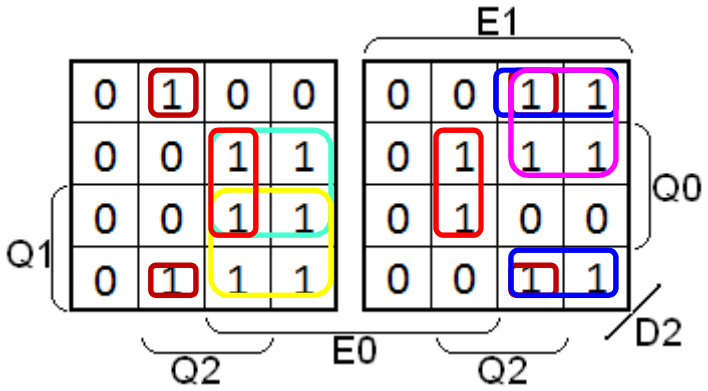
00 desplaza a la derecha con 0
01 desplaza a la izquierda con 0
10 desplaza a la derecha con 1
11 desplaza a la izquierda con 1

Tablas de transición de estados

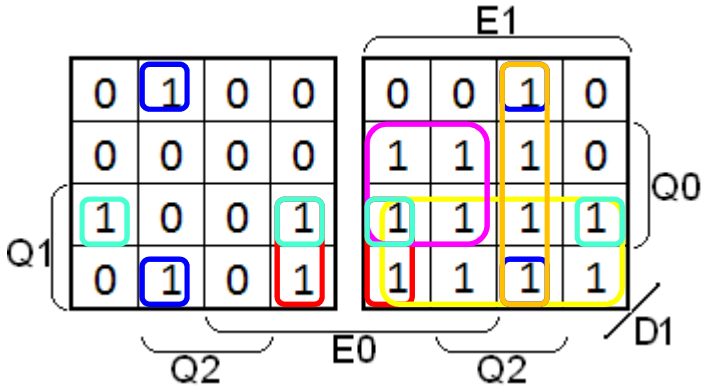
E1	E2	Q2	Q1	Q0	Q2+	Q1+	Q0+	D2	D1	D0	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0	0	1	0	1	1
0	0	0	1	1	0	1	0	0	1	0	1	1	1
0	0	1	0	0	1	0	1	1	0	1	1	0	0
0	0	1	0	1	0	0	1	0	0	1	0	1	0
0	0	1	1	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	0	1	0	0	1	0	1	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	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

E1	E2	Q2	Q1	Q0	Q2+	Q1+	Q0+	D2	D1	D0	S2	S1	S0
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

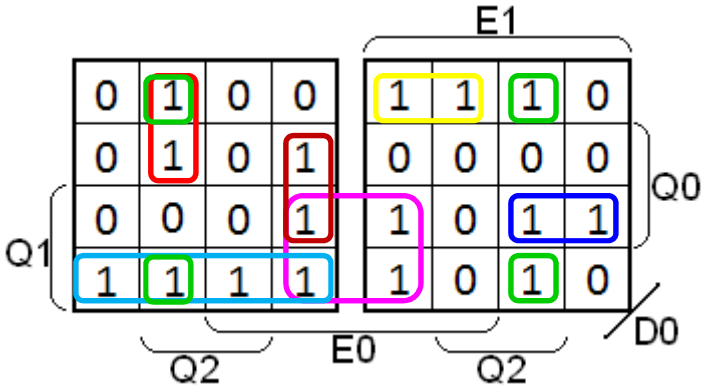
Mapas de Karnougth



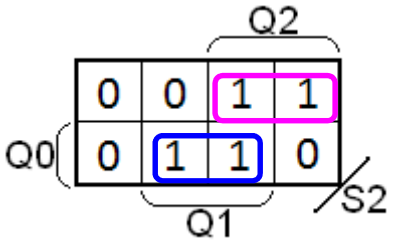
D2=(E0'*Q2*Q0')+(E1*E0'*Q0')+(E1*E0' Q1')
+(E1'*E0*Q0)+(E1'*E0*Q1)+(E0*Q2*Q0)



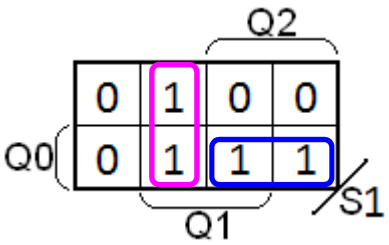
D1=(E0*Q2'*Q1)+(Q2'*Q1*Q0)+(E1*Q1)
+(E1*E0'*Q2)+(E1*E0*Q0)



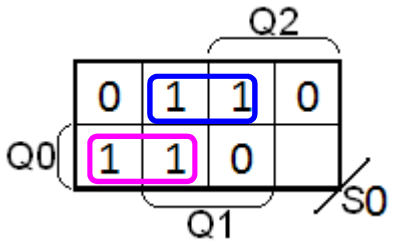
D0=(E1'*Q1*Q0')+(E1'*E0'*Q2*Q1')+
(E1'*E0*Q2'*Q0)+(E0*Q2'*Q1)+(E0'*Q2*Q0')
+(E1*E0'*Q1*Q0)+(E1*E0*Q1'*Q0')



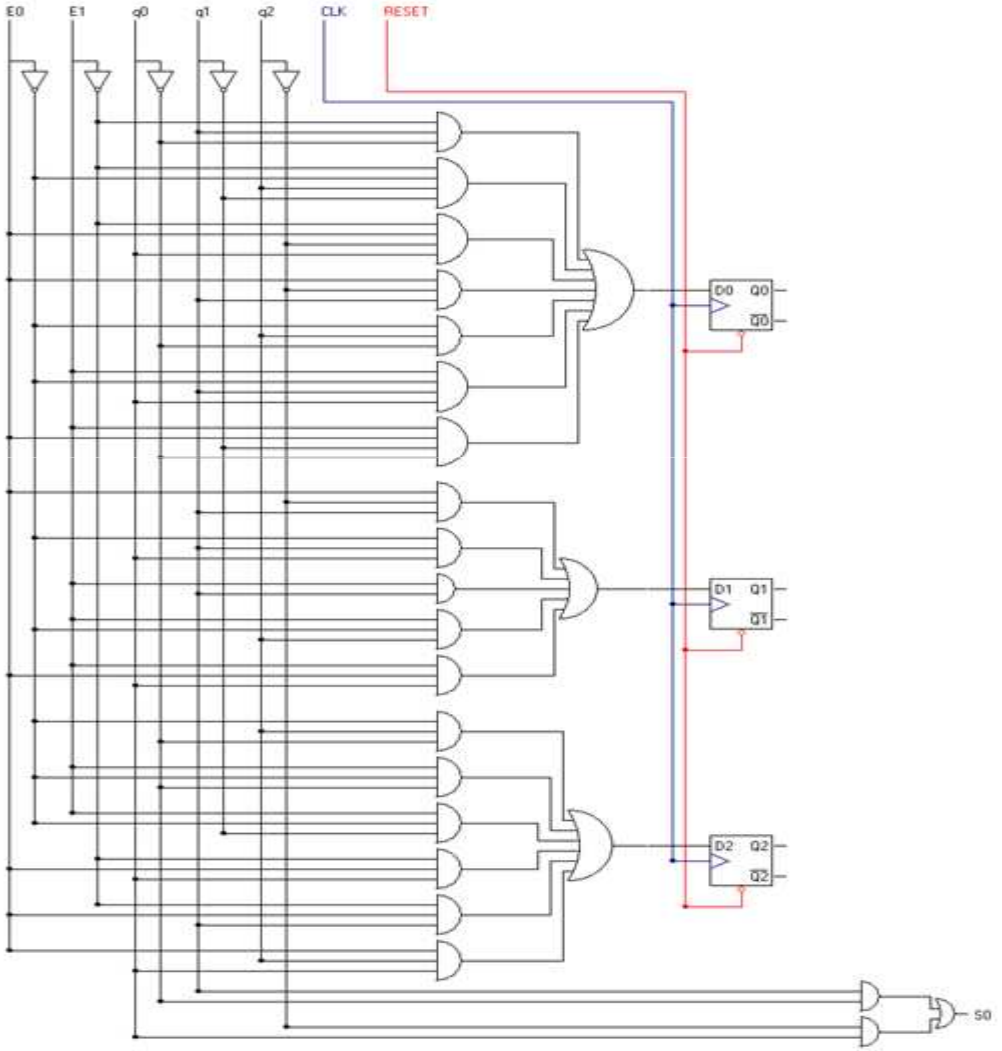
S2=(Q1*Q0)+(Q2*Q0')



S1=(Q2’*Q1)+(Q2*Q0)



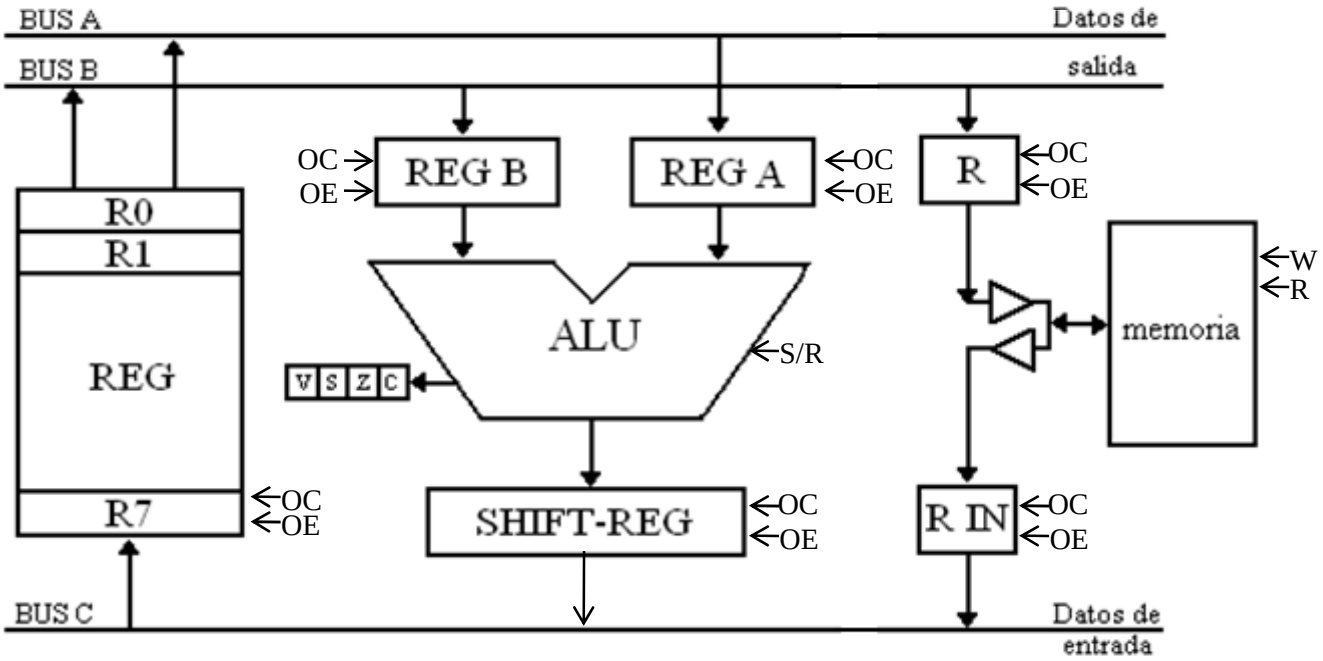
S0=(Q1*Q0’)+(Q2’*Q0)



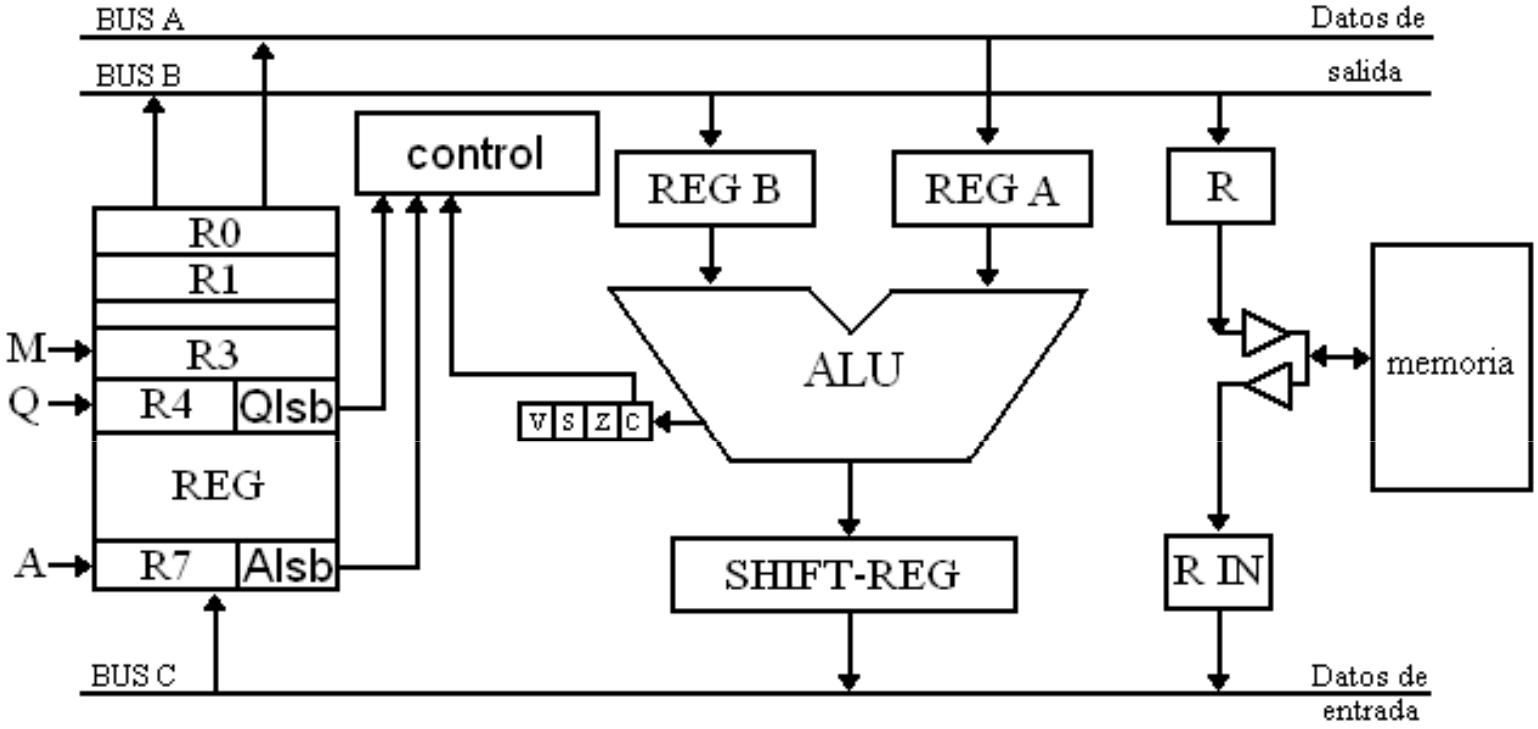
Diseño de un circuito controlador.

- 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 ⟵ R3*R4(usar R7 como registro auxiliar)
- M[100] ⟵ R3/R4



Multiplicador

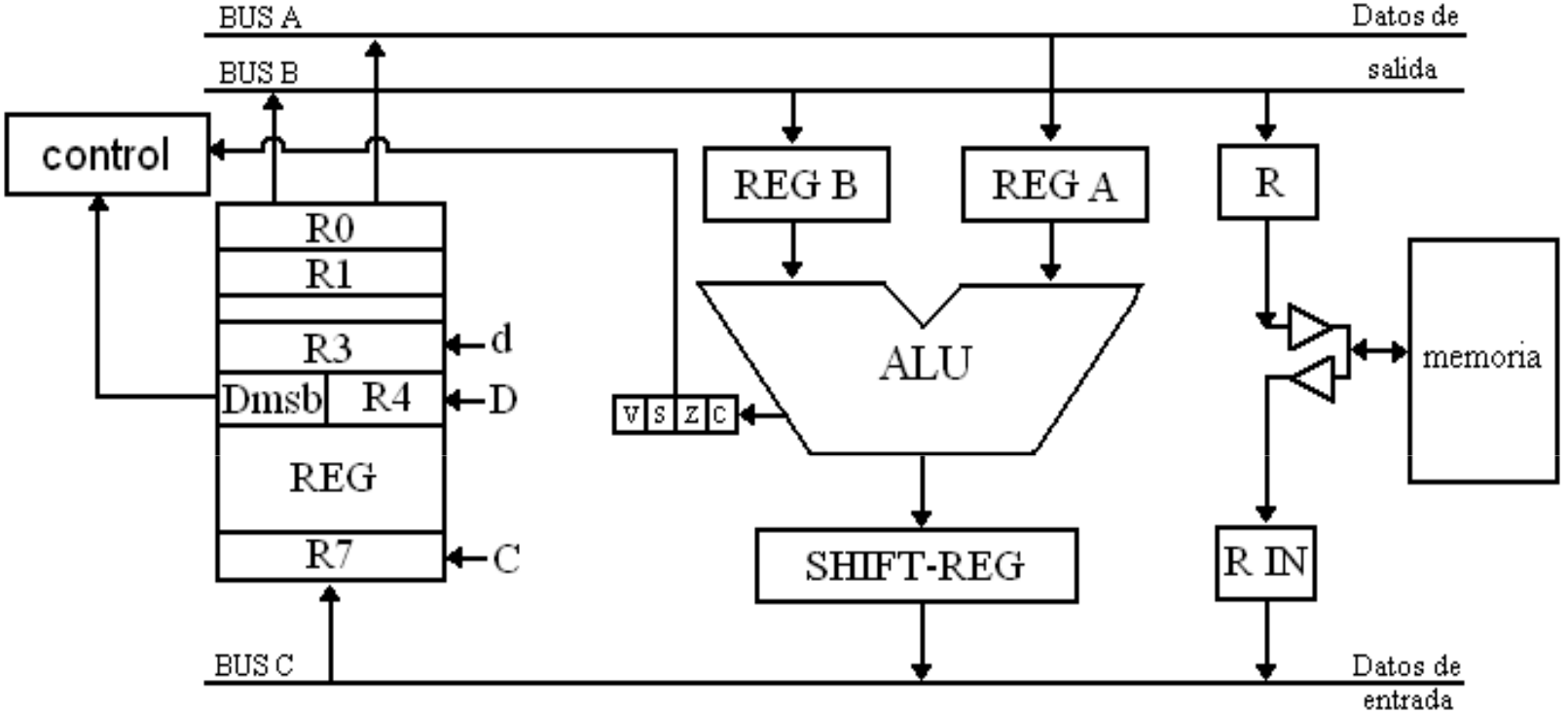


M = multiplicando

Q = multiplicador

A = acumulador

Divisor



d = dividendo

D = divisor

C = cociente

Especificaciones del diseño

SHIFT-REG	
no desplaza	Shift-Reg(000)
desplaza a la derecha con 0	Shift-Reg(001)
desplaza a la derecha con 1	Shift-Reg(010)
desplaza a la izquierda con 0	Shift-Reg(011)
desplaza a la izquierda con 1	Shift-Reg(100)

m = decisión “0” división
“1” multiplicación

Para la multiplicación:

Qlsb = bit menos significativo de R4

C = la bandera de la ALU, carry

Para la división:

Alsb = bit menos significativo de R7

Dmsb = bit mas significativo de R4

S = la bandera de la ALU

ASM

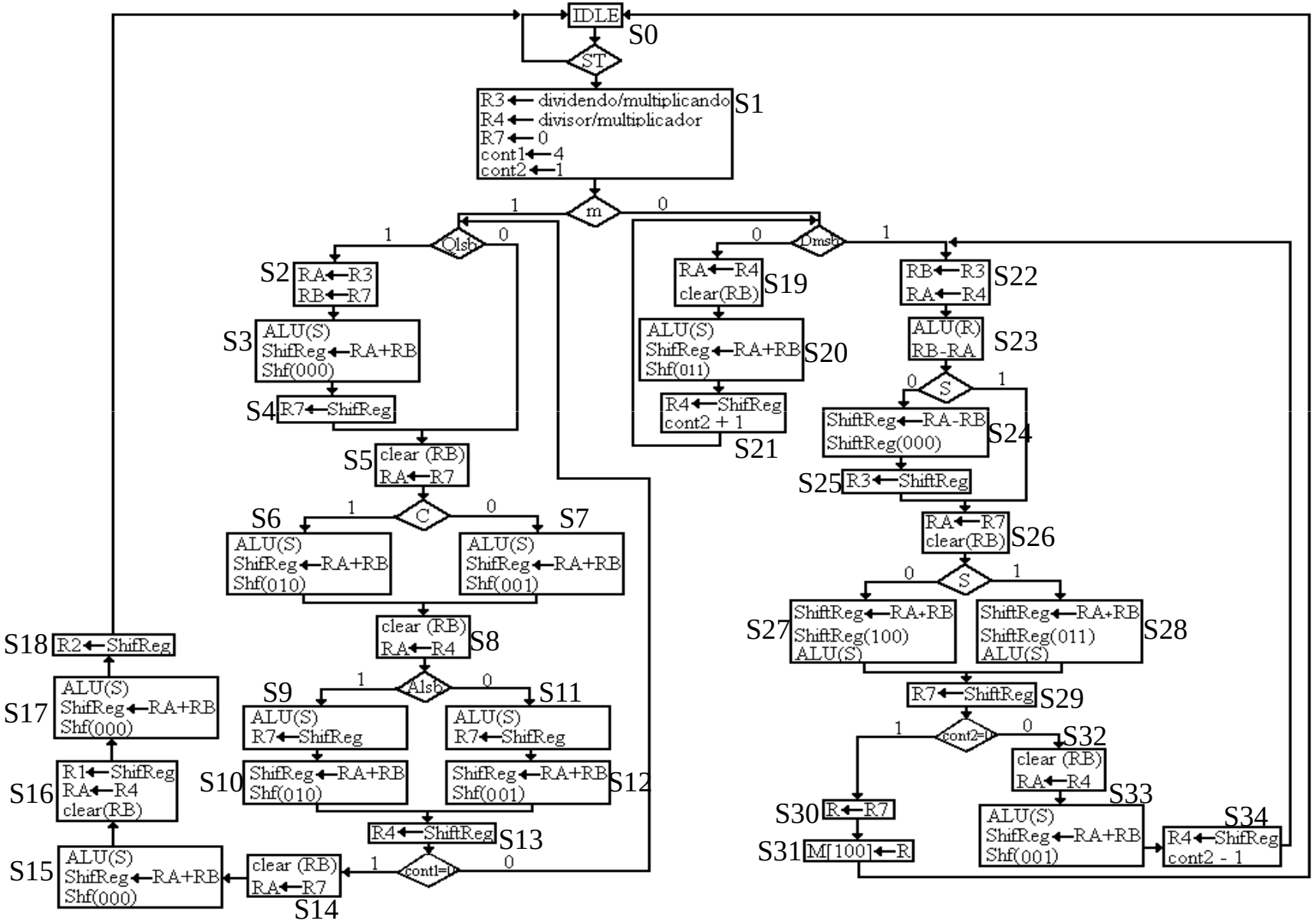
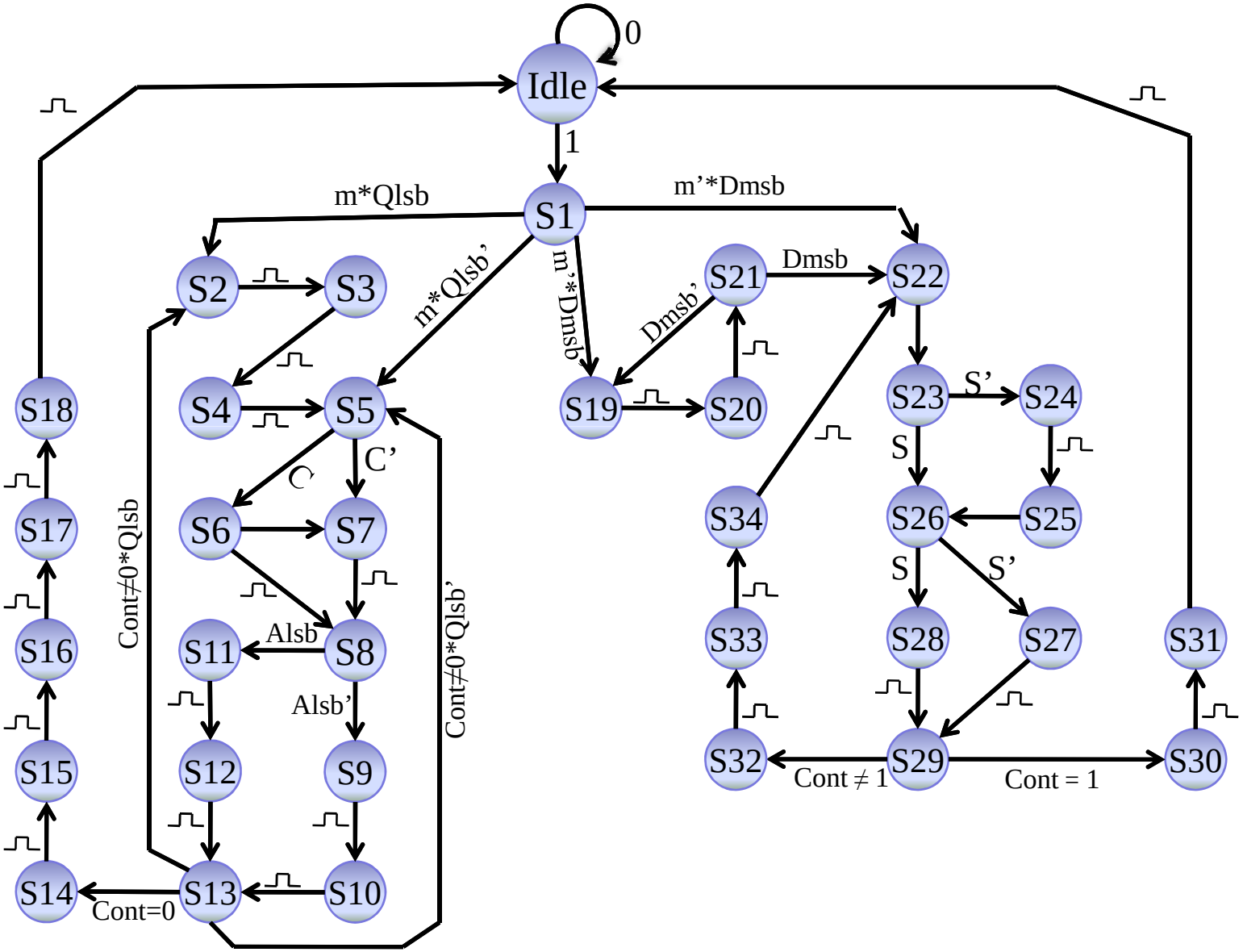
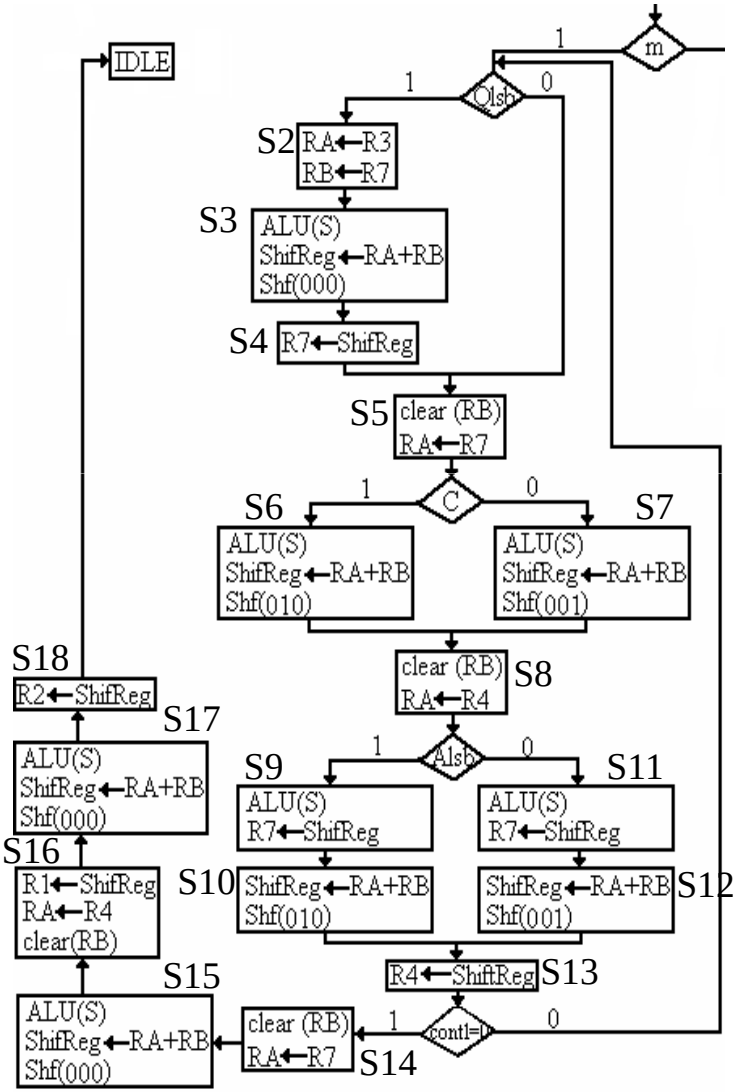

IDLE
S0
ST
R3 ← dividendo/multiplicando
R4 ← divisor/multiplicador
R7 ← 0
cont1 ← 4
cont2 ← 1
S1
m
1
0
Qlsb
S2
RA ← R3
RB ← R7
S3
ALU(S)
ShifReg ← RA+RB
Shf(000)
S4
R7 ← ShifReg
S5
clear (RB)
RA ← R7
C
S6
ALU(S)
ShifReg ← RA+RB
Shf(010)
S7
ALU(S)
ShifReg ← RA+RB
Shf(001)
clear (RB)
RA ← R4
S8
Alsb
S9
ALU(S)
R7 ← ShifReg
S11
ALU(S)
R7 ← ShifReg
S10
ShifReg ← RA+RB
Shf(010)
ShifReg ← RA+RB
Shf(001)
S12
R4 ← ShifReg
S13
cont1=0
clear (RB)
RA ← R7
S14
S15
ALU(S)
ShifReg ← RA+RB
Shf(000)
S16
R1 ← ShifReg
RA ← R4
clear(RB)
S17
ALU(S)
ShifReg ← RA+RB
Shf(000)
S18
R2 ← ShifReg
Dmsb
RA ← R4
clear(RB)
S19
ALU(S)
ShifReg ← RA+RB
Shf(011)
S20
R4 ← ShifReg
cont2 + 1
S21
RB ← R3
RA ← R4
S22
ALU(R)
RB-RA
S23
S
ShiftReg ← RA-RB
ShiftReg(000)
S24
R3 ← ShiftReg
S25
RA ← R7
clear(RB)
S26
S
ShiftReg ← RA+RB
ShiftReg(100)
ALU(S)
S27
ShiftReg ← RA+RB
ShiftReg(011)
ALU(S)
S28
R7 ← ShiftReg
S29
cont2=0
clear (RB)
RA ← R4
S32
S33
R ← R7
S30
M[100] ← R
S31
ALU(S)
ShifReg ← RA+RB
Shf(001)
R4 ← ShifReg
cont2 - 1
S34

Diagrama de Estados

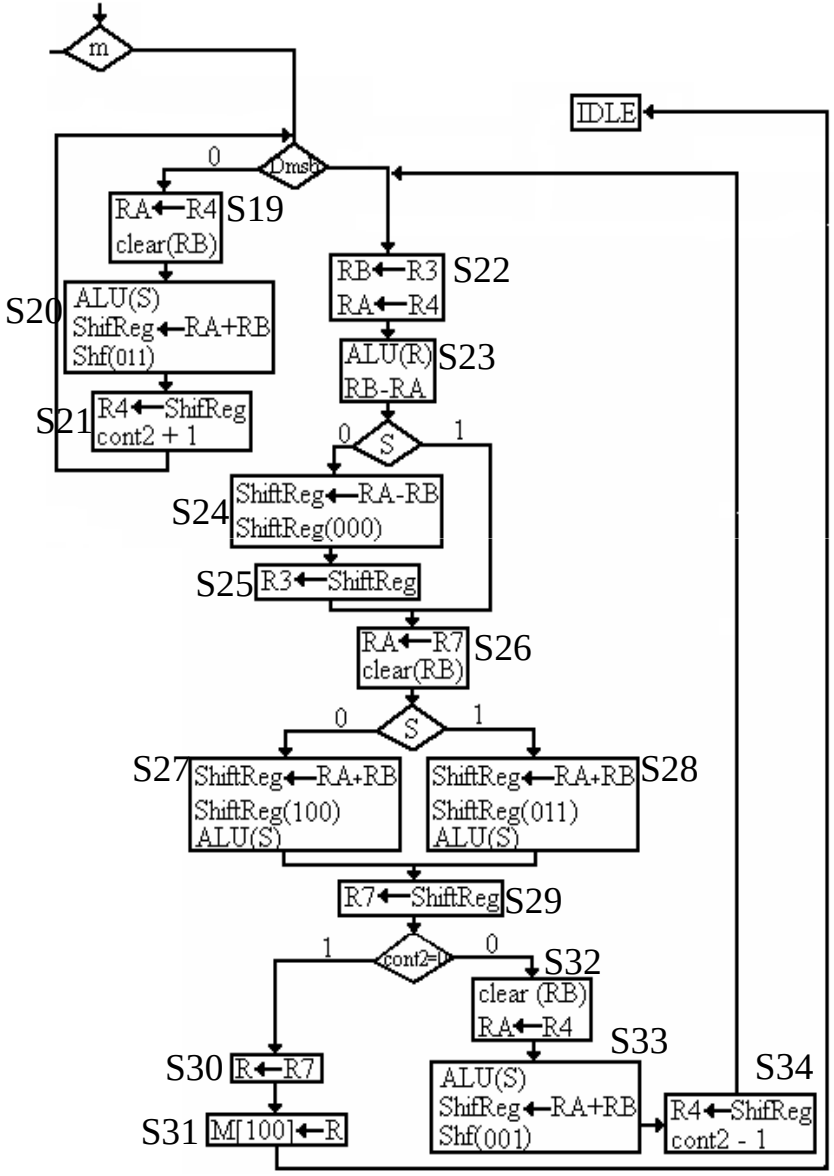


Demostración De Multiplicación



- S1: R3←1010 (multiplicando)
  R4←0011 (multiplicador)
  Cont1=4
- S2: RA←1010
  RB←0000
- S3: ShiftReg←1010
- S4: R7←1010
- S5: RA←1010
  RB←0000
- S7: ShiftReg←0101
- S8: RA←0011
  RB←0000
- S11: R7←0101
- S12: ShiftReg←0001
- S13: R4←0001
  Cont1=3
- S2: RA←1010
  RB=0101
- S3: ShiftReg←1111
- S4: R7←1111
- S5: RA←1111
  RB←0000
- S7: ShiftReg←0111
- S8: RA←0001
  RB←0000
- S9: R7←0111
- S10: ShiftReg←1000
- S13: R4←1000
  Cont1=2
- S5: RA←0111
  RB←0000
- S7: ShiftReg←0011
- S8: RA←1000
  RB←0000
- S9: R7←0011
- S10: ShiftReg←1100
- S13: R4←1100
  Cont1=1
- S5: RA←0011
  RB←0000
- S7: ShiftReg←0001
- S8: RA←1100
  RB←0000
- S9: R7←0001
- S10: ShiftReg←1110
- S13: R4←1110
  Cont1=0
- S14: RA←0001
  RB←0000
- S15: ShiftReg←0001
- S16: **R1← 0001**
  RA←1110
  RB←0000
- S17: ShiftReg←1110
- **S18: R2←1110**

Demostración De División



- S1: R3←1010 (dividendo)
  R4←0011 (divisor)
  Cont2←1
- S19: RA←0011
  RB←0000
- S20: ShiftReg ←0110
- S21: R4 ←0110
  cont2=2
- S19: RA ←0110
  RB ←0000
- S20: ShiftReg ←1100
- S21: R4 ←1100
  cont2=3
- S22: RB ←1010
  RA ←1100
- S23: 1110
- S26:RA ←0000
  RB ←0000
- S28: ShiftReg ←0000
- S29: R7 ←0000
- S32: RA←1100
  RB←0000
- S33: ShiftReg←0110
- S34: R4←0110
  Cont2=2
- S22: RB ←1010
  RA ←0110
- S23: 0100
- S24: ShiftReg←0100
- S25: R3 ←0100
- S26:RA ←0000
  RB ←0000
- S27: ShiftReg←0001
- S29: R7 ←0001
- S32: RA←0110
  RB←0000
- S33: ShiftReg←0011
- S34: R4←0011
  Cont2=1
- S22: RB ←0100
  RA ←0011
- S23: 0001
- S24: ShiftReg←0001
- S25: R3 ←0001
- S26:RA ←0001
  RB ←0000
- S27: ShiftReg←0011
- S29: R7 ←0011
- S30: R ←0011
- **S31: M[100] ←0011**

Melani Viveros Moreno
cod. 0425565