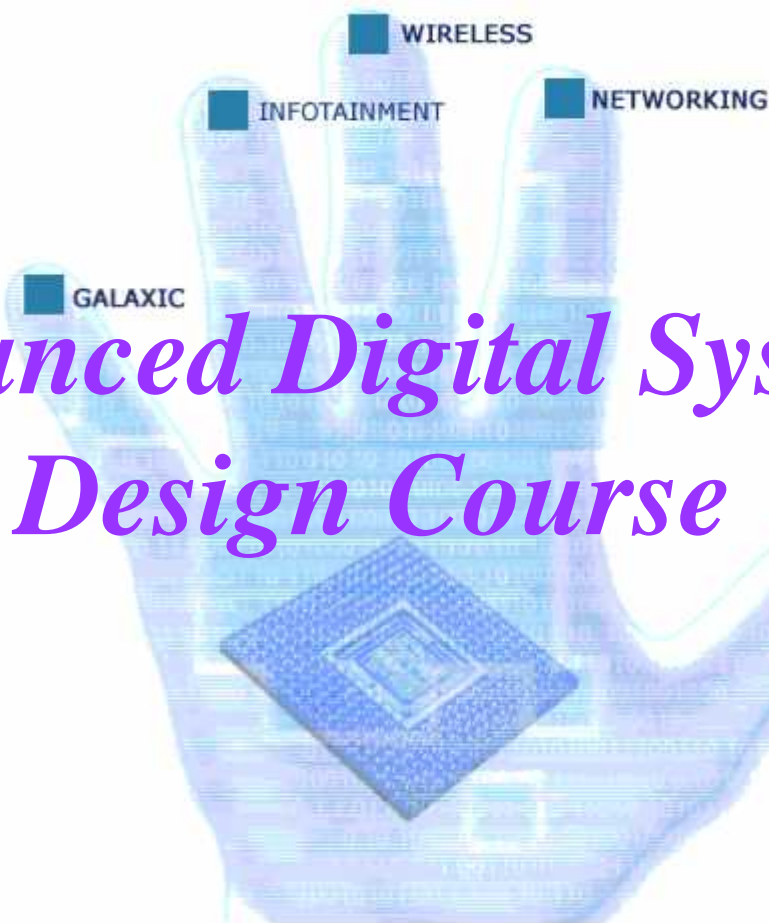




# *Advanced Digital System Design Course*



*Jacob Cardozo Londoño.  
0842004*

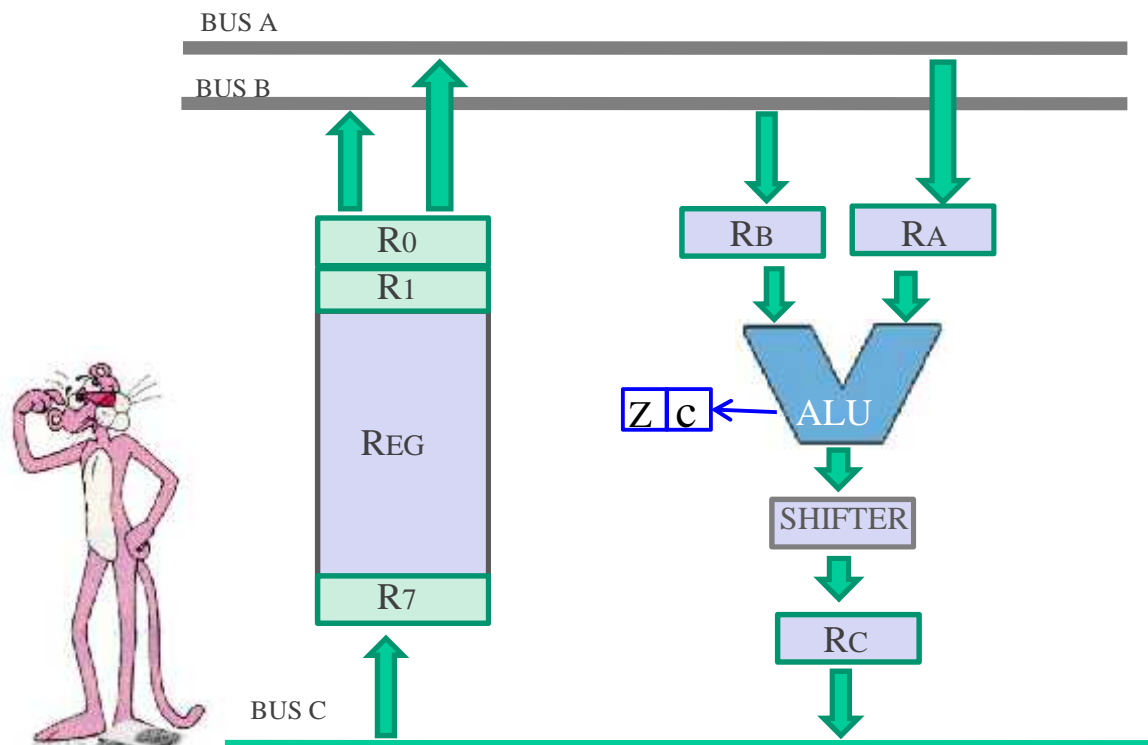


# *Advanced Digital System Design Course*



# División 2N/N con restauración

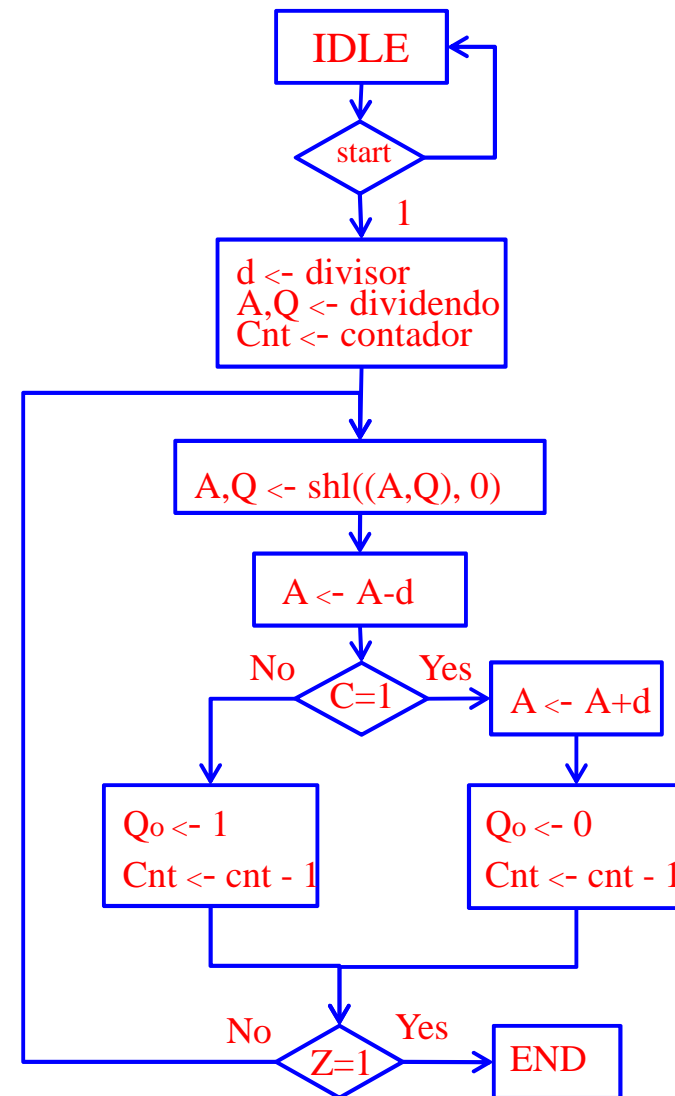
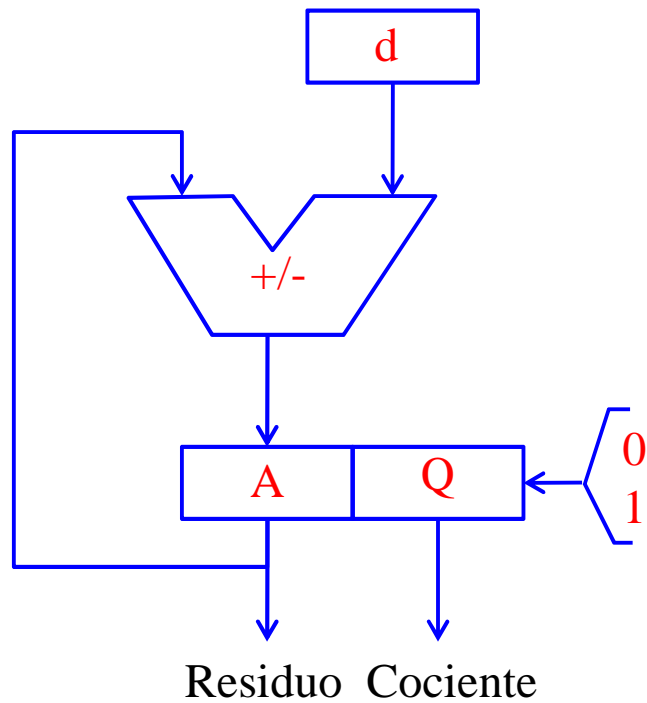
Diseñar una FSM para calcular la división 2N/N usando el procesador UV2009. Los datos son de 8 bits: dividendo en R0 y R1, divisor en R2, cociente en R3 y residuo en R4. Realizar el ASM.



La ALU ( $S_0S_1$ ): 00 = A or B;  
01 = A and B;  
10 = A + B;  
11 = A - B.

Shifter (S):  
1 desplaza lógico SHL;  
0 desplaza lógico SHR.

# ASM y Datapa original



# RTL división 2N/N

Primero se cargan los datos, se deben cargar dichos datos en los registros que van a interactuar en el algoritmo:

## RTL

## Control

1

```
1.1 RA  R1, RB  R3;
1.2 RC  RA+RB;
1.3 R3  RC;
1.4 RA  R4, RB  R0;
1.5 RC  RA+RB;
1.6 R4  RC;
      R5  0..1;
      R6 -- registro auxiliar;
      cnt 8;
```

```
RR1, WRA, RR2, WRB;
RRA, RRB, S0 $\overline{S_1}$ , WRC;
RRC, WR3;
RR4, WRA, RR0, WRB;
RRA, RRB, S0 $\overline{S_1}$ , WRC;
RRC, WR4;
WR5;
```

ya que cnt = nbits;

Se realiza el desplazamiento del acumulador y el cociente, se testea el bit mas significativo de R3 para saber si el LSB de R4 al ser desplazado es 1 o 0:

2

```
2.1 RA  R3, RB  R3;
2.2 RC  RA+RB;
```

```
RR1, WRA, ZRB;
RRA, RRB, S0 $\overline{S_1}$ , WRC;
```

If C= 1 then {2a} else {2b}

Si el carry es 1 el MBS de R3 es 1, de lo contrario MBS es 0, y se debe hacer el respectivo desplazamiento del dividendo.

# RTL división 2N/N

En este macro estado se hace el desplazamiento de del dividendo donde el LSB de R4 es igual a 1 y posteriormente se realiza el desplazamiento de R3:

2a

2a.1 RA	R4, RB	0;	RR4, WRA, ZRB;
2a.2 RC	shl ([RA+RB],0);		RRA, RRB, $S_0\bar{S}_1$ , S, WRC
2a.3 R4	RC;		RRC, WR4;
2a.4 RA	R4, RB	R5;	RR4, WRA, RR5, WRB;
2a.5 RC	RA+RB;		RRA, RRB, $S_0\bar{S}_1$ , WRC;
2a.6 R4	RC, RA	R3, RB	0;
2a.7 RC	shl ([RA+RB],0);		RRC, WR4, RR3, WRA; ZRB;
2a.8 R3	RC;		RRA, RRB, $S_0\bar{S}_1$ , S, WRC;
			RRC, WR3;

En este otro macro estado se hace el desplazamiento cuando el LSB de R4 es igual a 0 y se desplaza R3.

2b

2b.1 RA	R4, RB	0;	RR4, WRA, ZRB;
2b.2 RC	shl ([RA+RB],0);		RRA, RRB, $S_0\bar{S}_1$ , S, WRC;
2b.3 R4	RC, RA	R3, RB	0;
2b.4 RC	shl ([RA+RB],0);		RRC, WR4, RR3, WRA, ZRB;
2b.5 R3	RC;		RRA, RRB, $S_0\bar{S}_1$ , S, WRC;
			RRC, WR3;

# RTL división 2N/N

Luego se prosede a realizar el macro estado donde se ejecuta la resta:

3

3.1 RA	R4, RB	R2;	RR4, WRA, RR2, WRB;
3.2 RC	RA-RB;		RRA, RRB, $S_0S_1$ , WRC;
3.3 R4	RC;		RRC, WR4;

If C=1 then {5} else {4}

Si el carry es 1 indica que la resta no se pudo hacer y debemos restaurar (macro estado 5), en caso contrario la resta se pudo realizar y se continua con el proceso (macro estado 4).

Este es el macro estado donde la resta fue posible, se agrega 1 al cociente y disminuimos en uno el contador:

4

4.1 RA	R3, RB	R5;	RR3, WRA, RR5, WRB;
4.2 RC	RA+RB;		RRA, RRB, $S_0\bar{S}_1$ , WRC;
4.3 R3	RC;		RRC, WR4;
cnt	cnt -1		

# RTL división 2N/N

Este es el macro estado donde la resta no fue posible, se restaura el acumulador, el cociente queda con 0 y se disminuye el contador:

5

5.1 RA	R4, RB	R2;	RR4, WRA, RR2, WRB;
5.2 RC	RA+RB;		RRA, RRB, $S_0\bar{S}_1$ , WRC;
5.3 R4	RC;		RRC, WR4;
	cnt	cnt -1	

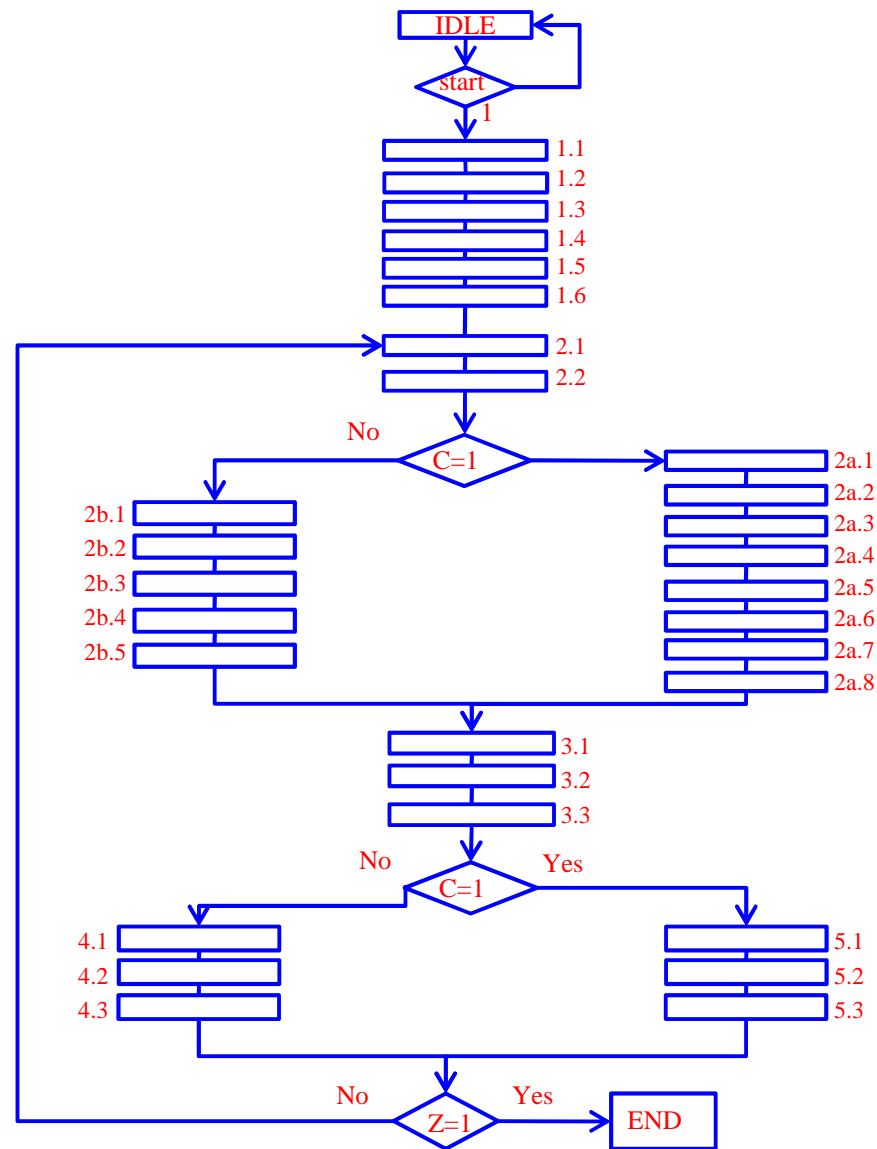
Por ultimo se realiza la comparación final, donde se observamos si el contador ya se encuentra en cero o aun nos hace falta interacciones:

If Z=1 then {END} else {2}

Si la señal Z se activa quiere decir que el contador ya esta en cero y terminar el proceso de lo contrario se debe regresar al macro estado 2.

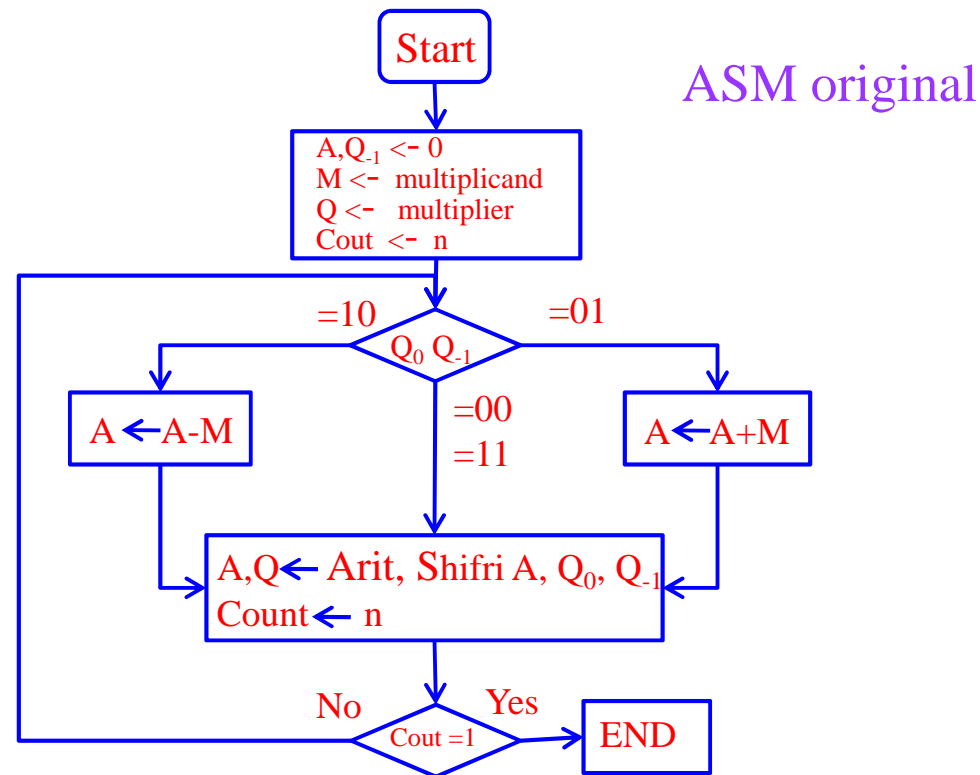


# ASM división 2N/N en el UV2009



# ASM y RTL para el algoritmo de Booth

Diseñar una FSM para calcular la multiplicación usando el algoritmo de Booth y el procesador UV2009. Los datos son de 8 bits; Multiplicando en R0, Multiplicador en R1, Producto en R2 y R3.



Primero cargamos los datos. Si los datos se encontraran en una memoria primero habría que cargarlos en los registros correspondientes ejemplo:

```
RIN    MRAM [M]
R0     RIN
RMRAM, WRIN;
RRIN, WR0;
```

Registros auxiliares:

R4 - en este registro el bit mas significativo (LSB) corresponde a  $Q_{-1}$   
 R5, R6 0  
 R7 1..0

Cargamos los datos:

```
1.1 R3 0;
1.2 RA R1, RB R3;
1.3 RC RA+RB;
1.4 R3 RC;
1.5 R2 0, R4 0, R5 0, R6 0;
1.6 R7 1..0;
```

Control:

```
ZR3;
RR1, WRA, RR3, WRB;
RRA, RRB,  $S_0\bar{S}_1$ , WRC;
RRC, WR3;
ZR2, ZR4, ZR5, ZR6;
WR7;
```

El siguiente paso es testear a  $Q_0$   $Q_{-1}$ . El tener a  $Q_{-1}$  en R4 como MSB ayuda a realizar todas las comparaciones seguidas y enviarlos a los macro estados correspondientes facilitando un poco el proceso.

Testeo el LSB de R3 que corresponde a  $Q_0$ , ya que este se cargo con los datos del multiplicador ( $Q=R1$ );

2

2.1 RA	R3, RB 0;	RR3, WRA, ZRB;
2.2 RC	shr ([RA+RB],0);	RRA, RRB, $S_0\bar{S}_1, \bar{S}$ , WRC
2.3 R5	RC;	RRC, WR5;
2.4 RA	R5, RB 0;	RR5, WRA, ZRB;
2.5 RC	shl ([RA+RB],0);	RRA, RRB, $S_0\bar{S}_1, S$ , WRC;
2.6 R5	RC;	RRC, WR5;
2.7 RA	R5, RB R3;	RR5, WRA, RR3, WRB;
2.8 RC	RA-RB;	RRA, RRB, $S_0S_1$ , WRC;

If C=1 then {3} else {4}

Si el carry es 1 entonces  $Q_0$  es uno, en caso contrario será 0, cada macro estado a que se dirige se testea  $Q_{-1}$  y con ello se podrá obtener las condiciones deseadas (00, 01, 10, 11).

Testeo el MSB de R4 que corresponde a  $Q_{-1}$ , ( $Q_0$  es igual a 1):

3

3.1 RA R4, RB R4;

RR4, WRA, WRB;

3.2 RC RA+RB;

RRA, RRB,  $S_0\bar{S}_1$ , WRC;

If C=1 then {7} else {5}

Si el carry es igual a 1 significa que  $Q_{-1}$  es igual a 1, de lo contrario será 0. Si  $Q_{-1}$  es 1 se tiene la condición 11 y se realiza el desplazamiento aritmético(macro estado 7) , de lo contrario se tendrá la condición 10 y se realiza la resta (macro estado 5).

Se testea de nuevo el MSB de R4 que corresponde a  $Q_{-1}$ , pero en este macro estado  $Q_0$  es igual a 0:

4

4.1 RA R4, RB R4;

RR4, WRA, WRB;

4.2 RC RA+RB;

RRA, RRB,  $S_0\bar{S}_1$ , WRC;

If C=1 then {6} else {7}

Si el carry es igual a 1  $Q_{-1}$  es igual a 1, se tiene la condición 01 donde se realiza la suma (macro estado 6), en caso contrario  $Q_{-1}$  es 0 y se tiene como condición 00 donde se realiza el desplazamiento (macro estado 7).

Macro estado donde se realiza la resta de multiplicando y el acumulador:

5

5.1 RA	R2, RB	R0;	RR2, WRA, RR0, WRB;
5.2 RC	RA-RB;		RRA, RRB, $S_0S_1$ , WRC;
5.3 R2	RC;		RRC, WR4;

Macro estado donde se realiza la suma del multiplicando y el acumulador:

6

6.1 RA	R2, RB	R0;	RR2, WRA, RR0, WRB;
6.2 RC	RA+RB;		RRA, RRB, $S_0\bar{S}_1$ , WRC;
6.3 R2	RC;		RRC, WR4;

Macro estado donde se realiza el desplazamiento aritmético, primero se desplaza R4 que contiene a  $Q_{-1}$ , luego R3 que contiene el multiplicando y por ultimo el acumulador R2, esto facilita el desplazamiento aritmético:

7

7.1 RA	R3, RB	0;	RR3, WRA, ZRB;
7.2 RC	shr ([RA+RB],0);		RRA, RRB, $S_0S_1$ , S, WRC
7.3 R5	RC;		RRC, WR5; $\bar{\phantom{x}} \bar{\phantom{x}}$
7.4 RA	R5, RB	0;	RR5, WRA, ZRB;
7.5 RC	shl ([RA+RB],0);		RRA, RRB, $S_0\bar{S}_1$ , S, WRC;
7.6 R5	RC;		RRC, WR5;
7.7 RA	R5, RB	R2;	RR5, WRA, RR3, WRB;
7.8 RC	RA-RB;		RRA, RRB, $S_0S_1$ , WRC;

If C= 1 then{8} else {9}

Si el carry es 1 entonces  $Q_0$  es igual a 1 y se debe desplazar R4 con un 1 para que  $Q_{-1}$  tome ese nuevo valor (macro estado 8), de lo contrario se será 0 y se desplaza con 0 (macro estado 9).

Desplazamiento de R4:

8

8.1 RA R4, RB 0;  
8.2 RC shr ([RA+RB],0);  
8.3 R4 RC;  
8.4 RA R4, RB R7;  
8.5 RC RA+RB;  
8.6 R4 RC;

RR4, WRA, ZRB;  
RRA, RRB,  $S_0\bar{S}_1, \bar{S}$ , WRC;  
RRC, WR4;  
RR4, WRA, RR7, WRB;  
RRA, RRB,  $S_0\bar{S}_1$ , WRC;  
RRC, WR4;

9

9.1 RA R4, RB 0;  
9.2 RC shr ([RA+RB],0);  
9.3 R4 RC;

RR4, WRA, ZRB;  
RRA, RRB,  $S_0\bar{S}_1, \bar{S}$ , WRC;  
RRC, WR4;

Luego se testea al LSB de R2 (acumulador) para saber con que bit desplazo a R3:

10

10.1 RA	R2, RB	0;	RR2, WRA, ZRB;
10.2 RC	shr ([RA+RB],0);		RRA, RRB, $S_0\bar{S}_1, \bar{S}$ , WRC
10.3 R5	RC;		RRC, WR5;
10.4 RA	R5, RB	0;	RR5, WRA, ZRB;
10.5 RC	shl ([RA+RB],0);		RRA, RRB, $S_0\bar{S}_1, S$ , WRC;
10.6 R5	RC;		RRC, WR5;
10.7 RA	R5, RB	R2;	RR5, WRA, RR3, WRB;
10.8 RC	RA-RB;		RRA, RRB, $S_0S_1$ , WRC;

If C= 1 then{ 11} else { 12}

Si el carry es 1 el LSB de R2 es 1, se debe desplazar R3 con un 1 (macro estado 11), de lo contrario se será 0 y se desplaza con 0 (macro estado 12).

Desplazamiento de R3:

11

11.1 RA	R3, RB	0;	RR3, WRA, ZRB;
11.2 RC	shr ([RA+RB],0);		RRA, RRB, $S_0\bar{S}_1, \bar{S}$ , WRC;
11.3 R4	RC;		RRC, WR4;
11.4 RA	R3, RB	R7;	RR4, WRA, RR7, WRB;
11.5 RC	RA+RB;		RRA, RRB, $S_0\bar{S}_1$ , WRC;
11.6 R3	RC;		RRC, WR4;



12

12.1 RA R3, RB 0;  
12.2 RC shr ([RA+RB],0);  
12.3 R3 RC;

RR3, WRA, ZRB;  
RRA, RRB,  $S_0\bar{S}_1, \bar{S}$ , WRC;  
RRC, WR3;

Hasta el momento ya se desplazo Q y a  $Q_{-1}$ , por lo tanto solo hace falta desplazar el acumulador R2, como es aritmético hay que testear el MSB para saber si desplaza con 1 ó 0:

13

13.1 RA R2, RB R2;  
13.2 RC RA+RB;

RR2, WRA, WRB;  
RRA, RRB,  $S_0\bar{S}_1, \bar{S}$ , WRC;

If C= 1 then{ 14} else { 15}

Si el carry es 1 el MSB de R2 es 1, se deberá hacer el desplazamiento con 1 (macro estado 14), de lo contrario se desplaza con 0 (macro estado 15).

Desplazamiento aritmético de R2:

14

14.1 RA R2, RB 0;  
14.2 RC shr ([RA+RB],0);  
14.3 R2 RC;  
14.4 RA R2, RB R7;  
14.5 RC RA+RB;  
14.6 R2 RC;

RR2, WRA, ZRB;  
RRA, RRB,  $S_0\bar{S}_1, \bar{S}$ , WRC;  
RRC, WR2;  
RR2, WRA, RR7, WRB;  
RRA, RRB,  $S_0\bar{S}_1, \bar{S}$ , WRC;  
RRC, WR2;

15

15.1 RA R2, RB 0;

15.2 RC shr ([RA+RB],0);

15.3 R2 RC;

RR2, WRA, ZRB;

RRA, RRB, S<sub>0</sub>S<sub>1</sub>, S, WRC;

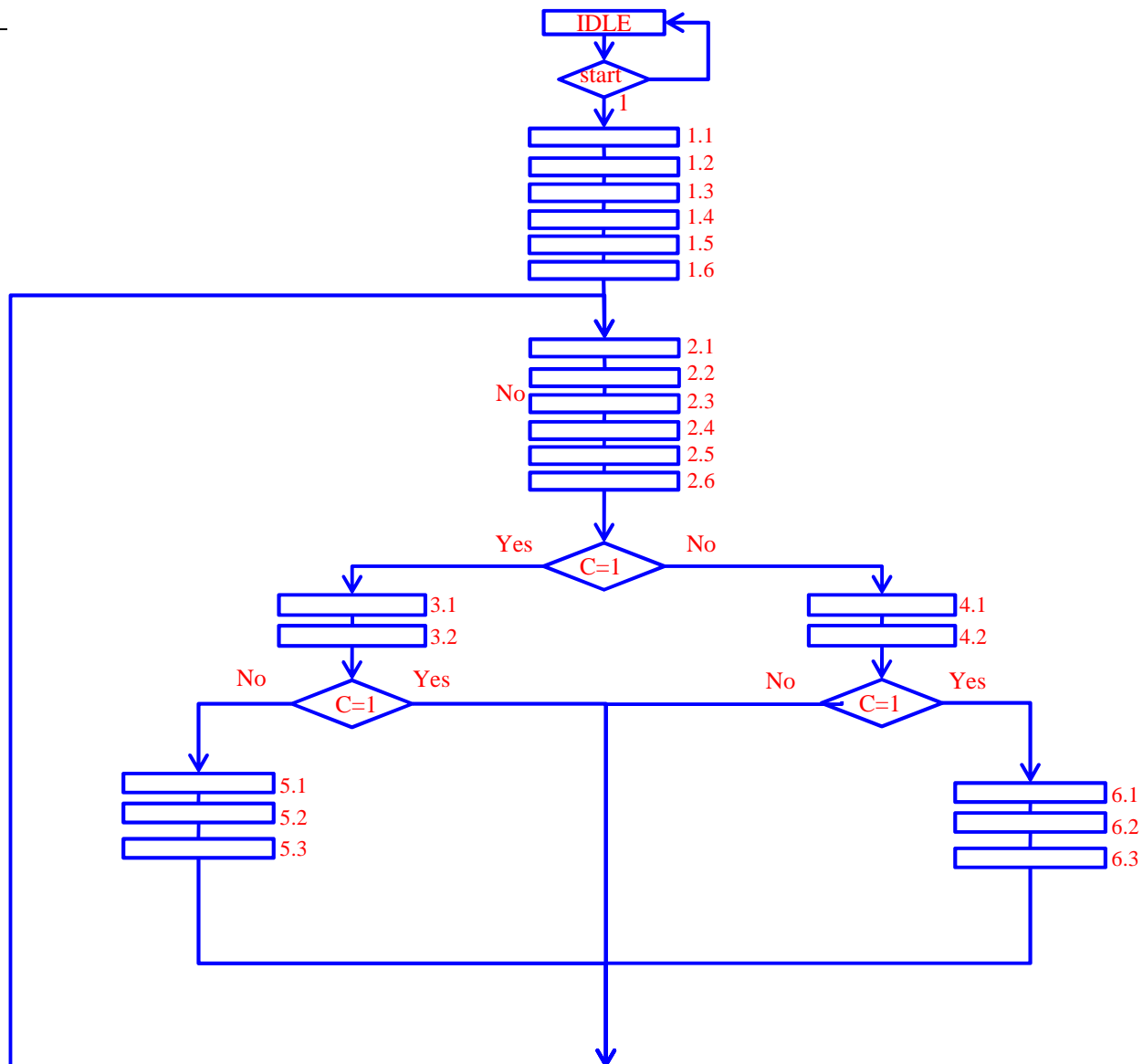
RRC, WR2;

Por ultimo se realiza la condicional que verifica si el contador ya esta encero y el proceso termina:

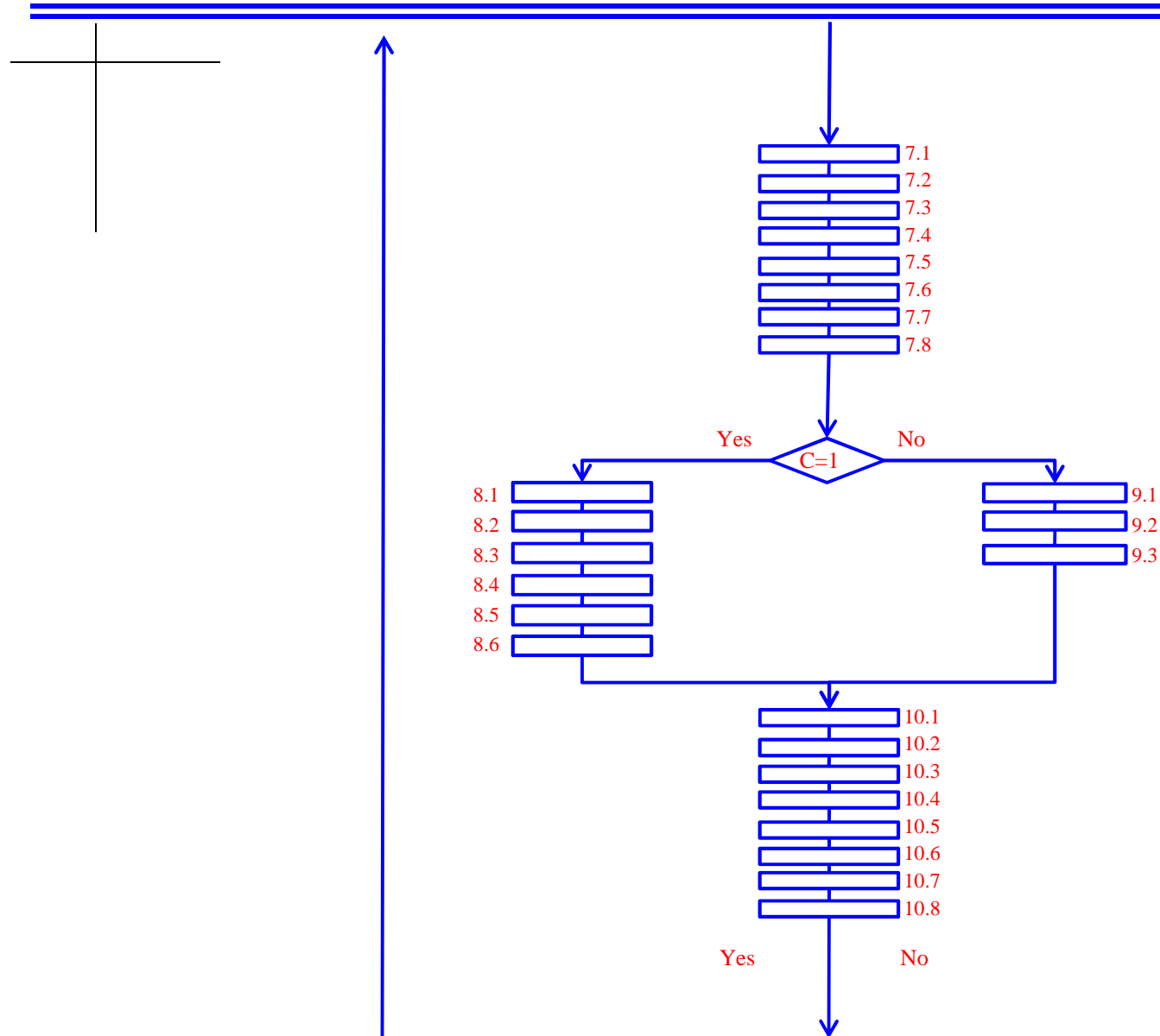
If Z= 1 then{END} else {2}

Si la señal Z se activa quiere decir que el contador llevo a cero y termina el proceso. El producto en R2 y R3, si por el contrario Z es 0 se regresa al magro estado 2.

# ASM



# ASM



# ASM

