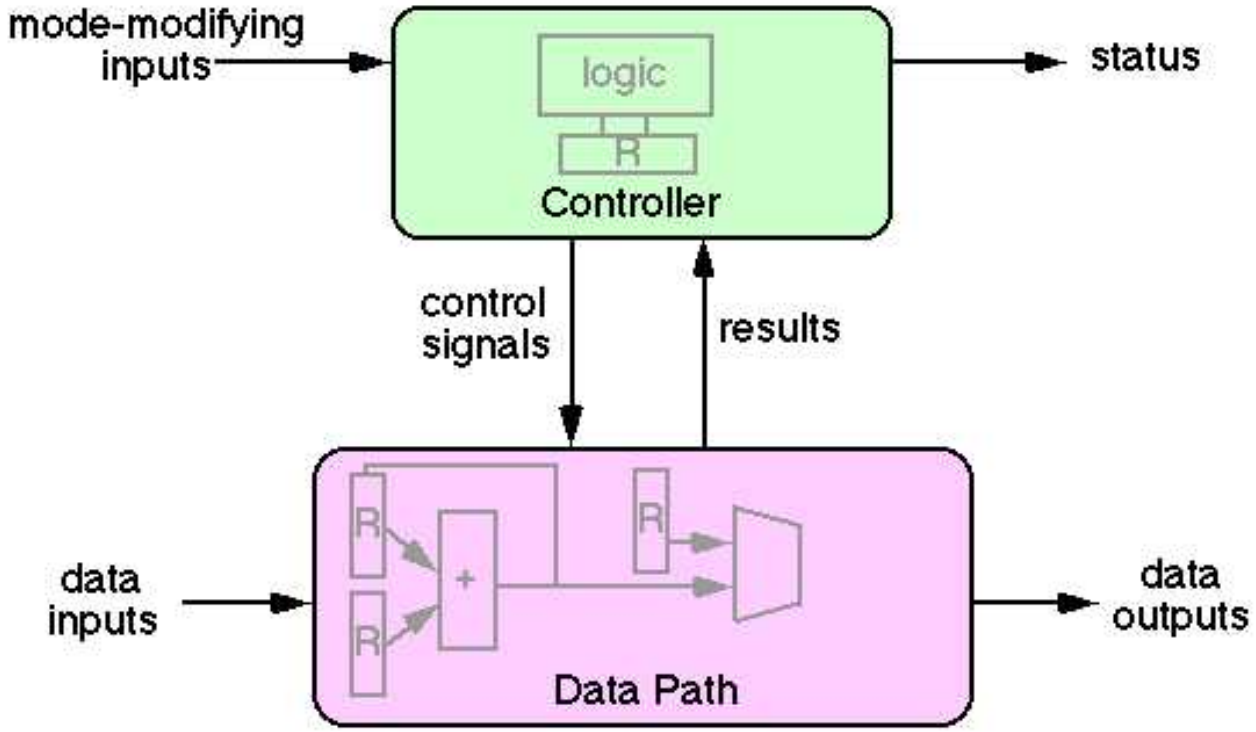
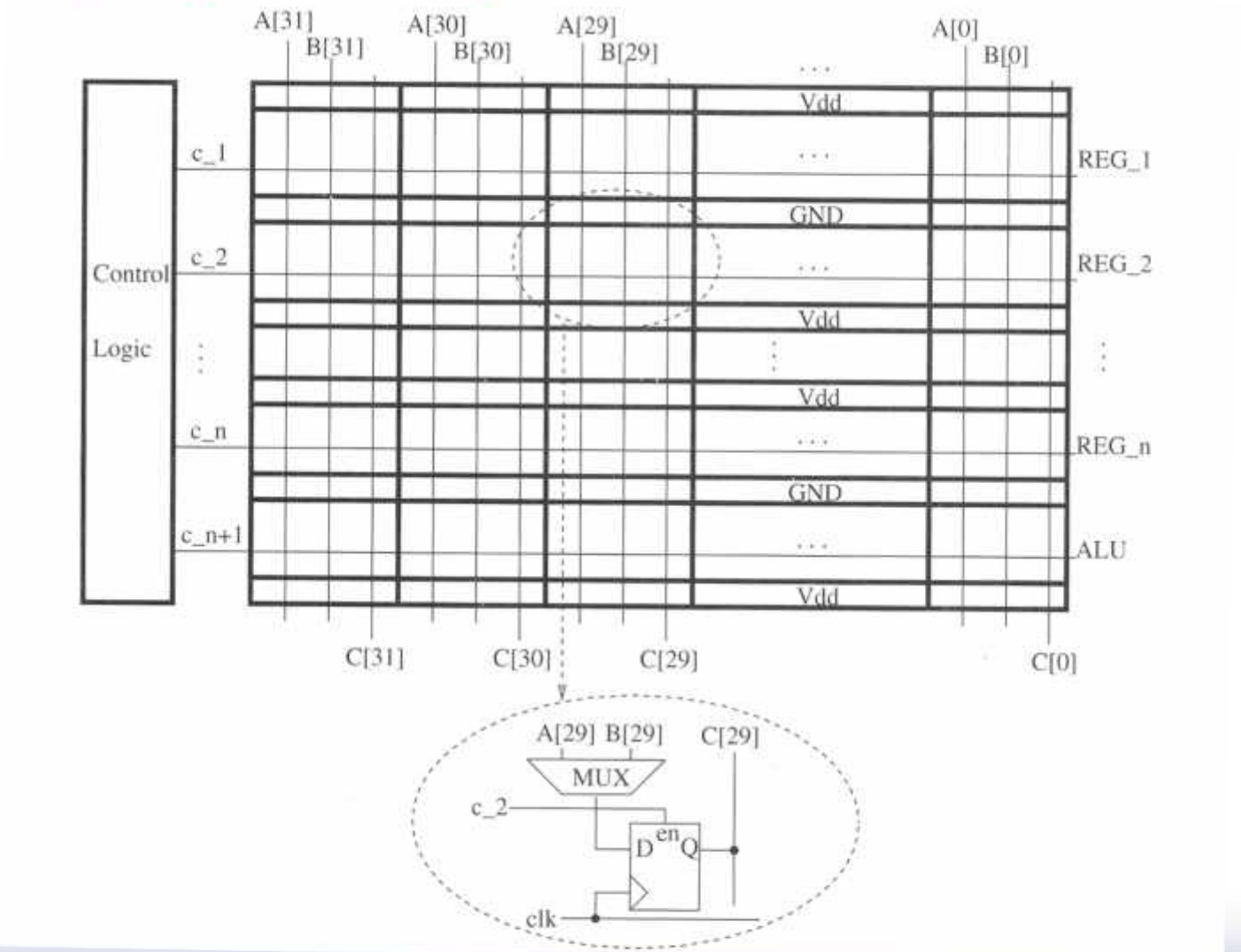
Circuitos Integrados
de Aplicación Específica

Método sistemático de Diseño Lógico

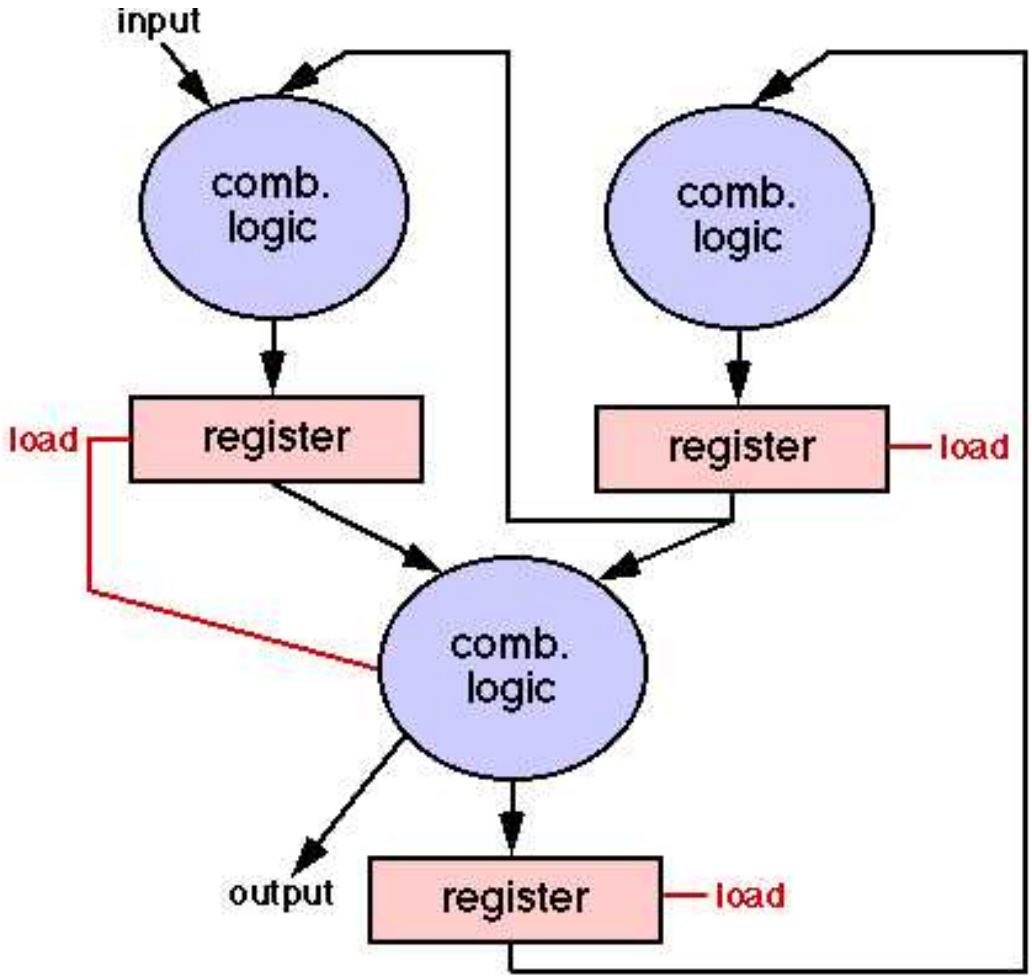
Estructura general de un circuito digital



Ejemplo de Layout de una ALU de 32 bits



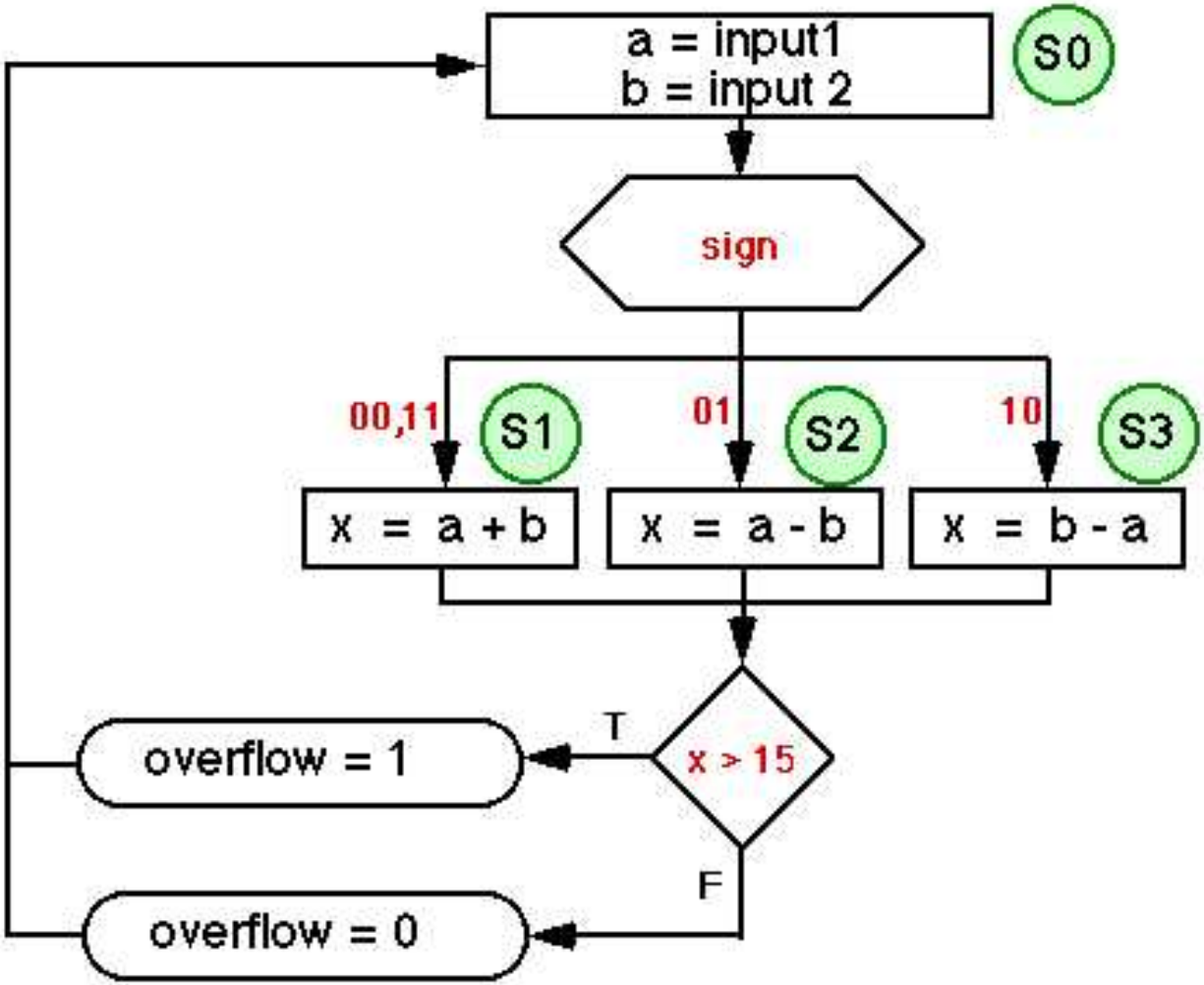
Especificación RTL



Register Transfer Notation (RTN)

Notation	Intended Operation
$X \leftarrow Y$	Transfer the contents of register Y to register X
$REG \leftarrow 0$	Clear the contents of register REG
$X \leftarrow$ all 1's	Set all bits of register X
$X \leftarrow 1$	Set the lsb of X and reset all other bits
$X\langle 31:0\rangle \leftarrow$ $0@X\langle 31:1\rangle$	one-bit right shift ($X >> 1$) with $X\langle 31\rangle \leftarrow 0$
$X\langle 31:0\rangle \leftarrow X\langle 3:0\rangle$ $@X\langle 31:4\rangle$	four-bit end-around right shift
$X \leftarrow M[1234H]$	$X \leftarrow$ contents of memory at address 1234H
$X \leftarrow M[Y]$	$X \leftarrow$ contents of memory pointed to by Y
$X \leftarrow Y \vee Z$	$X \leftarrow Y$ "OR" Z (bitwise operation)
$X \leftarrow Y \wedge Z$	$X \leftarrow Y$ "AND" Z (bitwise operation)
$X \leftarrow Y \oplus Z$	$X \leftarrow Y$ "EX-OR" Z (bitwise operation)
$X \leftarrow \overline{Y}$	$X \leftarrow$ 1s' complement of Y
$X \leftarrow -Y$	$X \leftarrow$ 2's complement of Y
$X \leftarrow Y + Z$	$X \leftarrow$ addition of Y and Z
$X \leftarrow Y - Z$	$X \leftarrow$ subtraction of Z from Y
$K{+}{+}$	Shorthand for $K \leftarrow K + 1$
$*$ If $(c = 1)X \leftarrow Y$	equivalent to $X \leftarrow (\overline{c} \wedge X) \vee (c \wedge Y)$ (refer to explanation in text)

Grafo ASM


a = input1
b = input 2
S0
sign
00,11
S1
01
S2
10
S3
x = a + b
x = a - b
x = b - a
T
x > 15
overflow = 1
F
overflow = 0

Especificación de estados

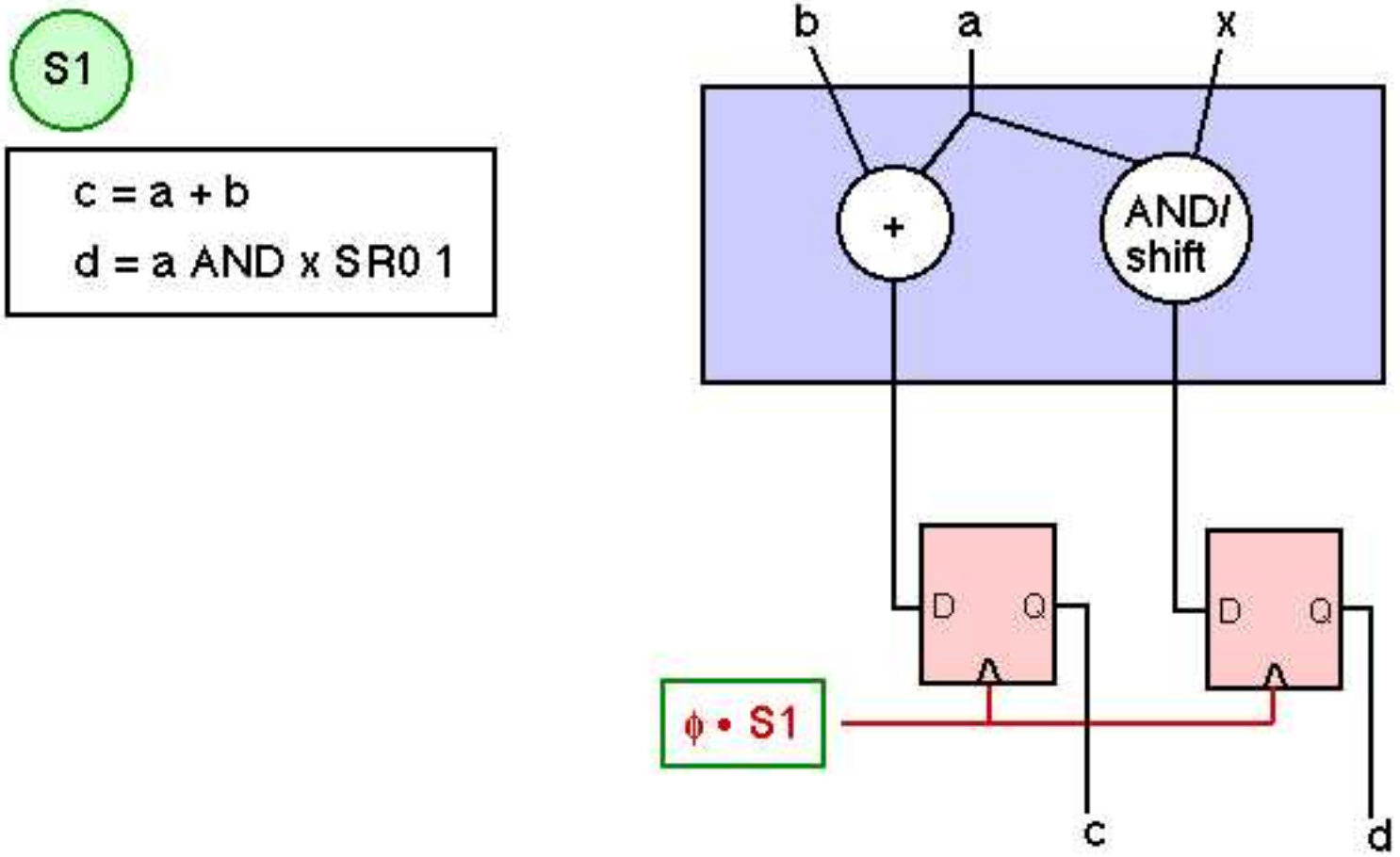
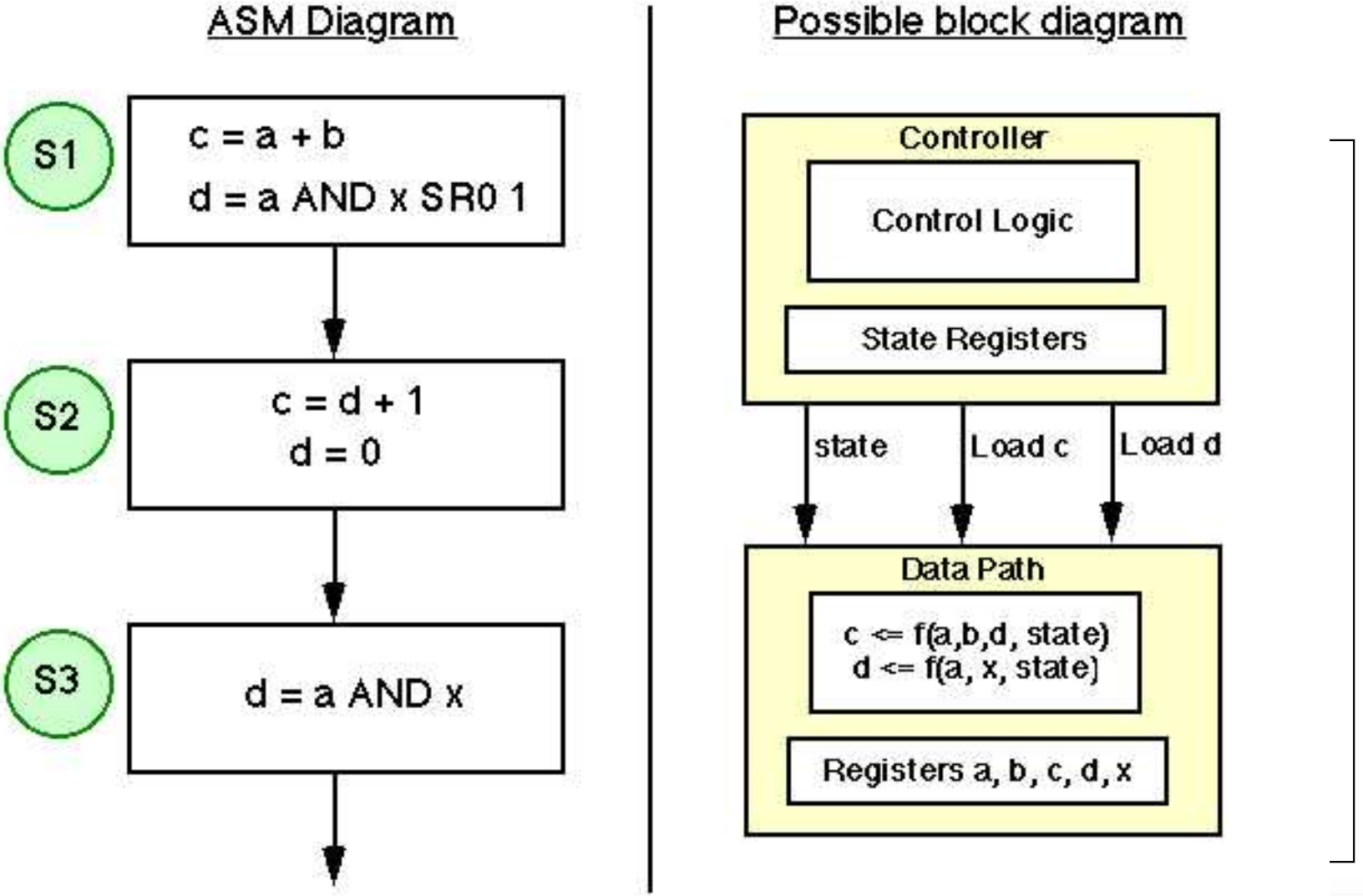
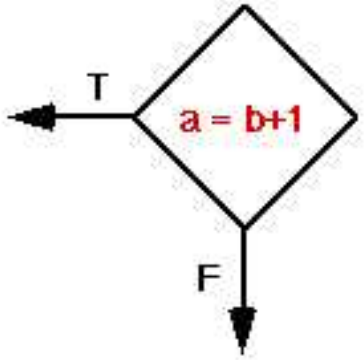


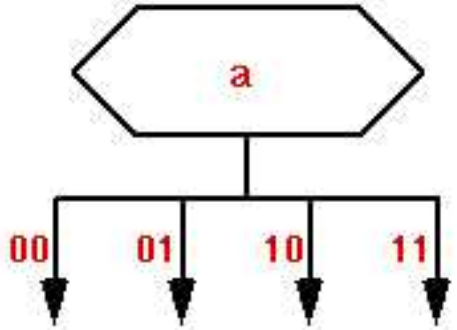
Diagrama de flujo de control ASM



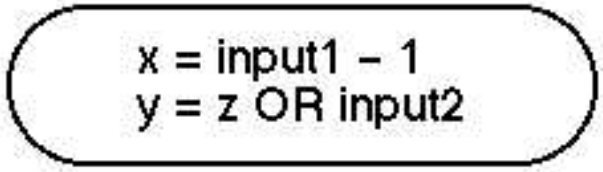
Símbolos ASM

Conditionals

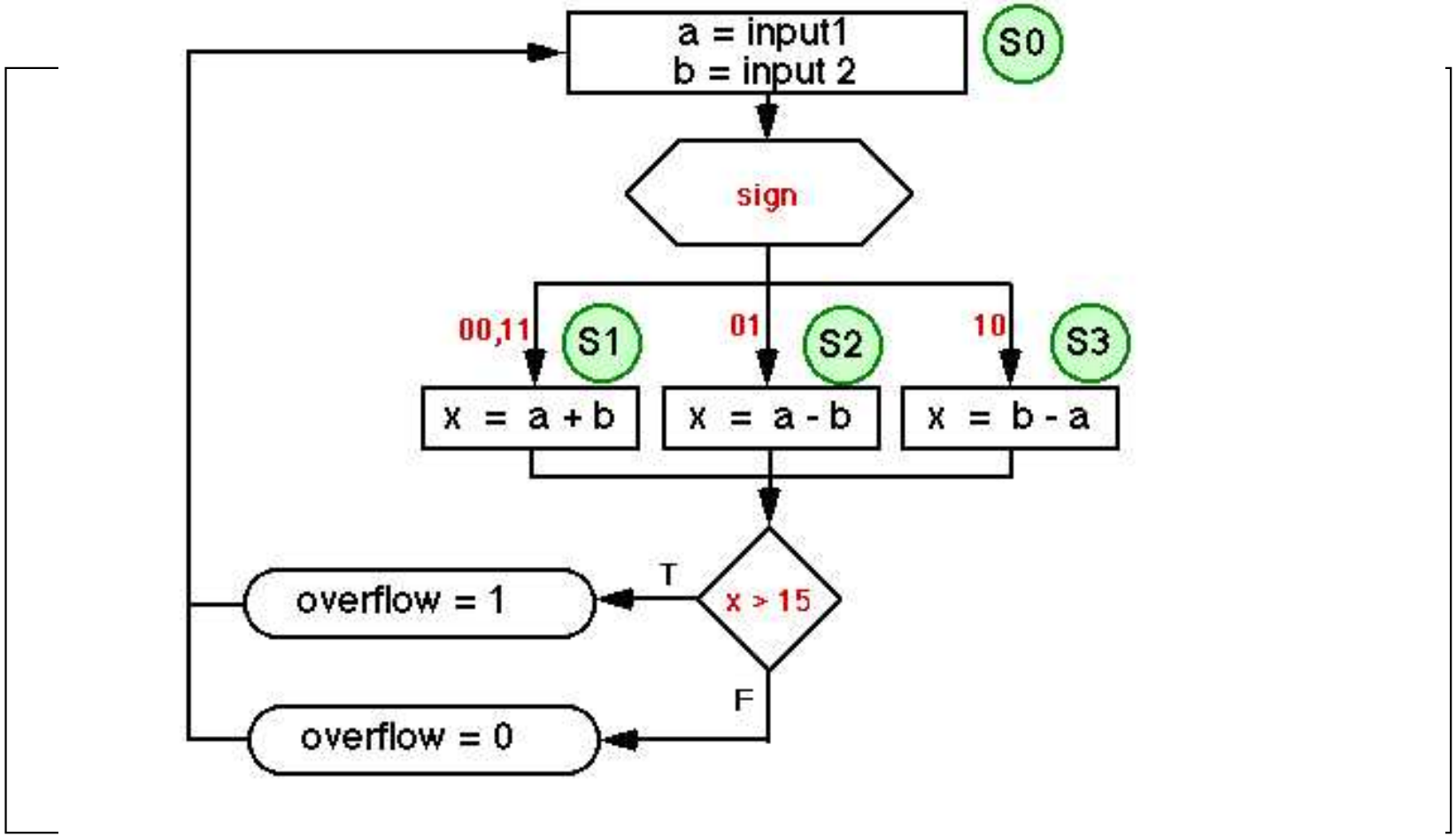




Conditional output (unclocked logic)



Ejemplo: Sumador con signo



Grafo ASM en forma de tabla

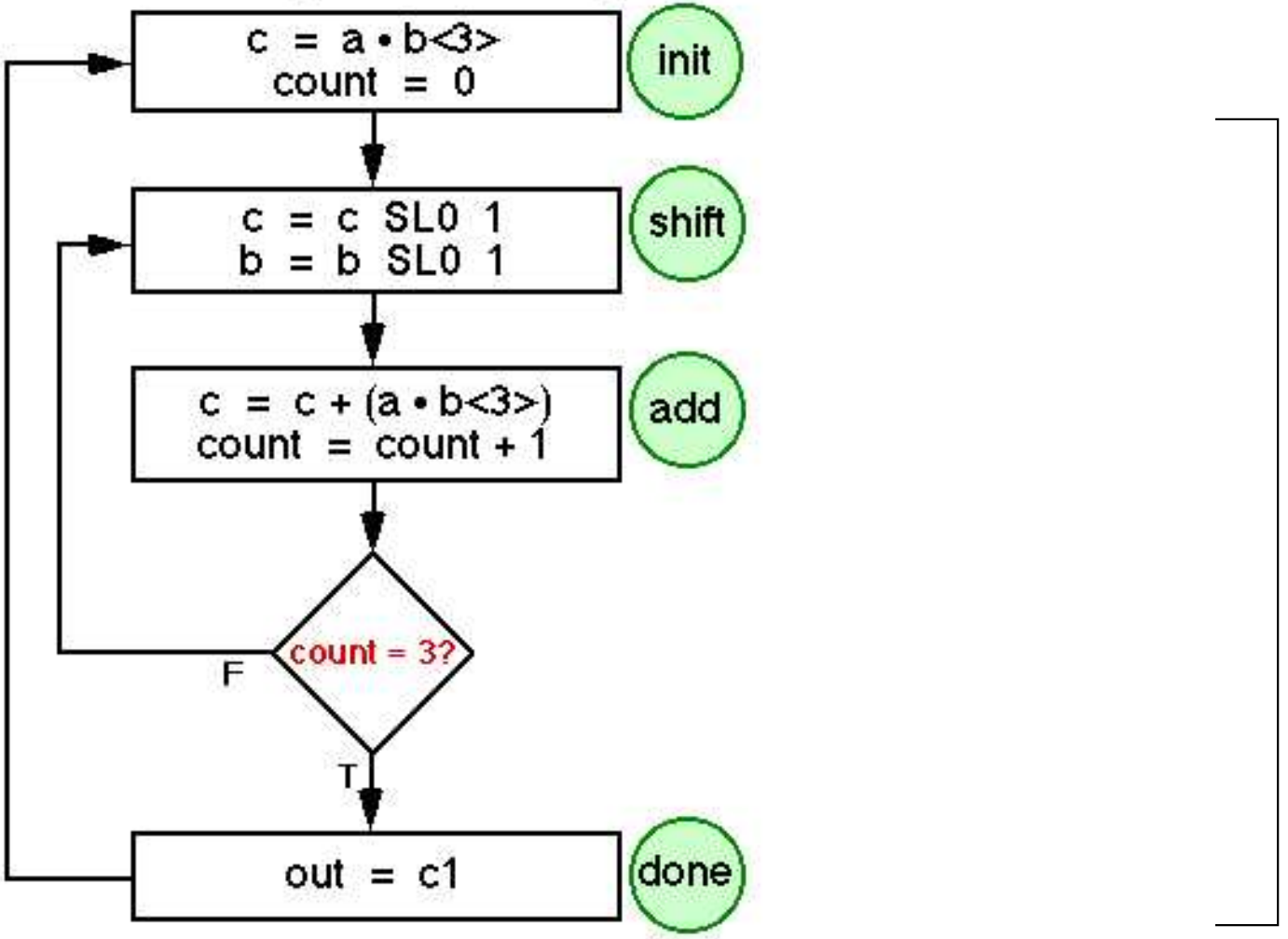
ESTADO S0	CONDICIÓN C0	ESTADO S1	ESTADO S2	ESTADO S3	CONDICIÓN C2

a ← input1;
b ← input2;
	If ((sign = 00) OR (sign =11)
	(goto S1);
	If (sign = 01)
	(goto S2);
	
	If (sign = 10)
	(goto S3);
		x ← a+b;
		(goto C2);
			x ← a-b;
			(goto C2);
				x ← b-a;
				(goto C2);
					If (x>15) then
					overflow ← 1;
					Else
					overflow ← 1;
					(goto S0);

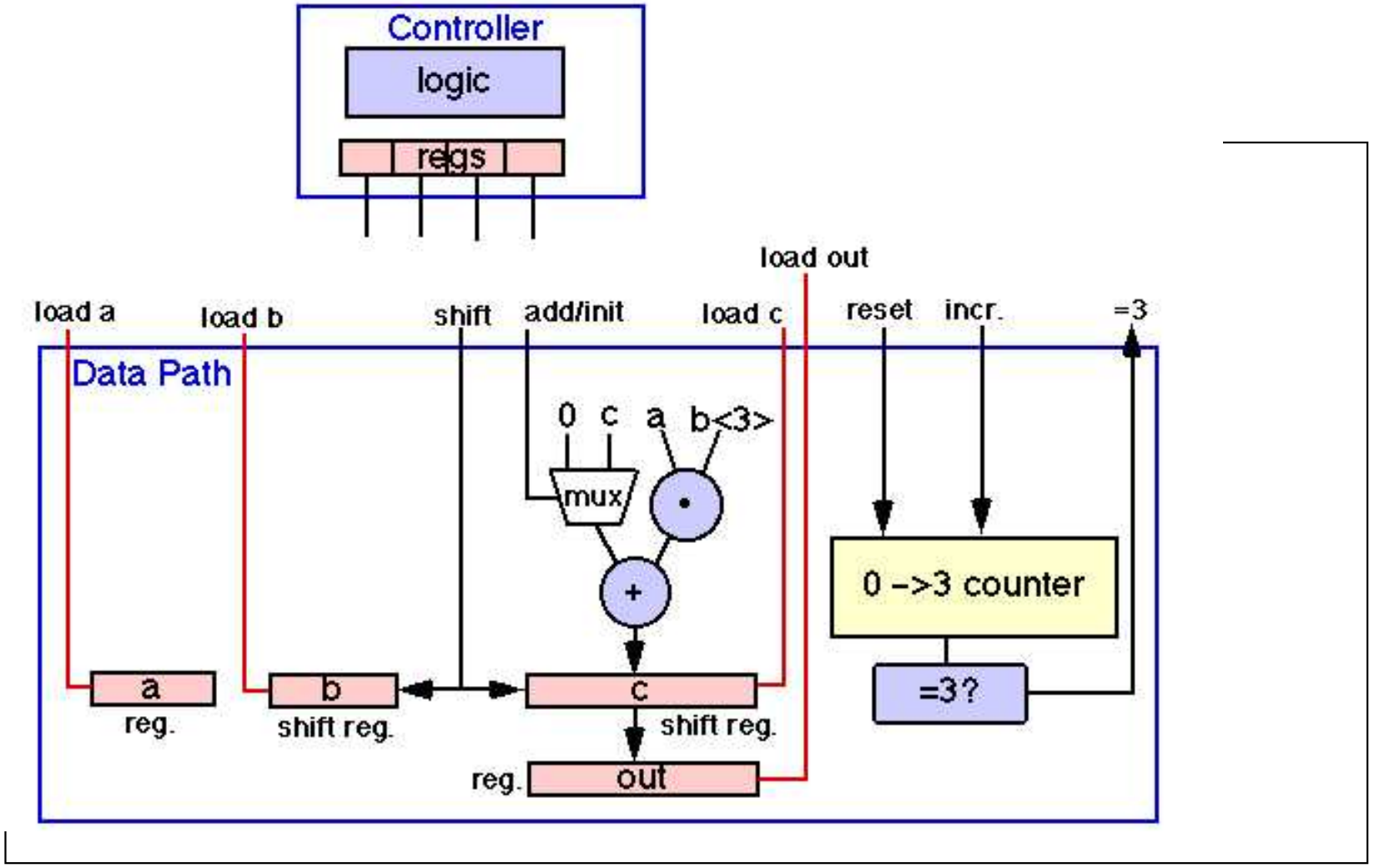
Grafo ASM en forma de texto

```
S0.     a ← input1;
        b ← input2;
C0.     If ((sign = 00) OR (sign =11) (goto S1);
        If (sign = 01) (goto S2);
        If (sign = 10) (goto S3);
S1.     x ← a+b;
        (goto C2);
S2.     x ← a-b;
        (goto C2);
S3.     x ← a-b;
C2.      If (x>15) then overflow  ← 1;
        Else overflow  ← 1;
        (goto S0);
```

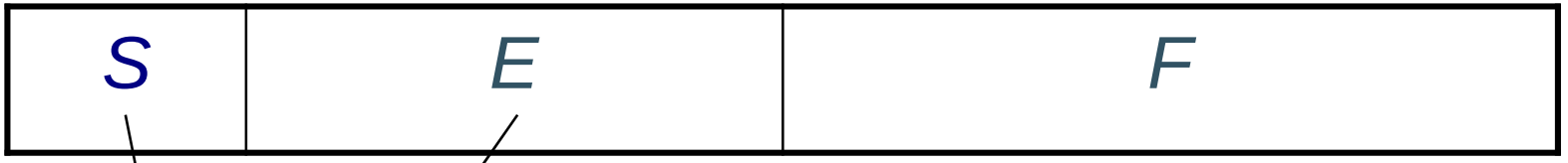
Ejemplo: Multiplicador de 4 bits



Arquitectura



Práctica nº1 : SUMADOR DE PUNTO FLOTANTE DE PRECISIÓN SIMPLE



Signo(1 BIT) ***Exponente (8 BITS)*** ***Mantisa (23 BITS)***

Interpretación : $(-1)^S \times 2^{E-127} \times (1.F)$

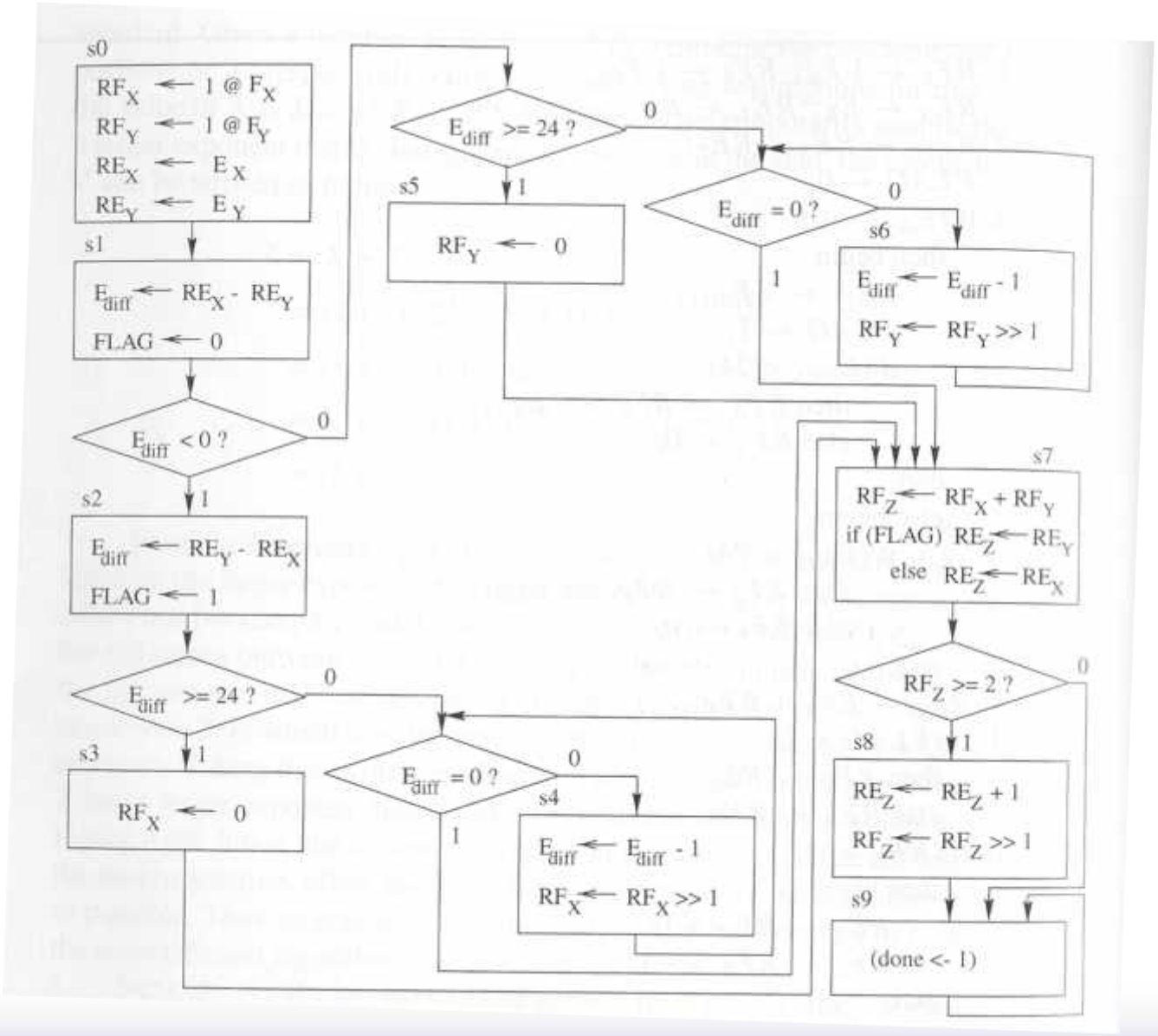
1. ALGORITMO

$$
\begin{aligned}
Z &= X + Y \\
&= (1.F_X) \cdot 2^{E_X - 127} + (1.F_Y) \cdot 2^{E_Y - 127} \\
&= (1.F_X + 1.F_Y \cdot 2^{-((E_X-127)-(E_Y-127))}) \cdot 2^{E_X-127} \\
&= (1.F_X + (1.F_Y) \cdot 2^{-|E_X - E_Y|}) \cdot 2^{E_X-127} \\
&= (1.F_Z) \cdot 2^{E_Z-127}
\end{aligned}
$$

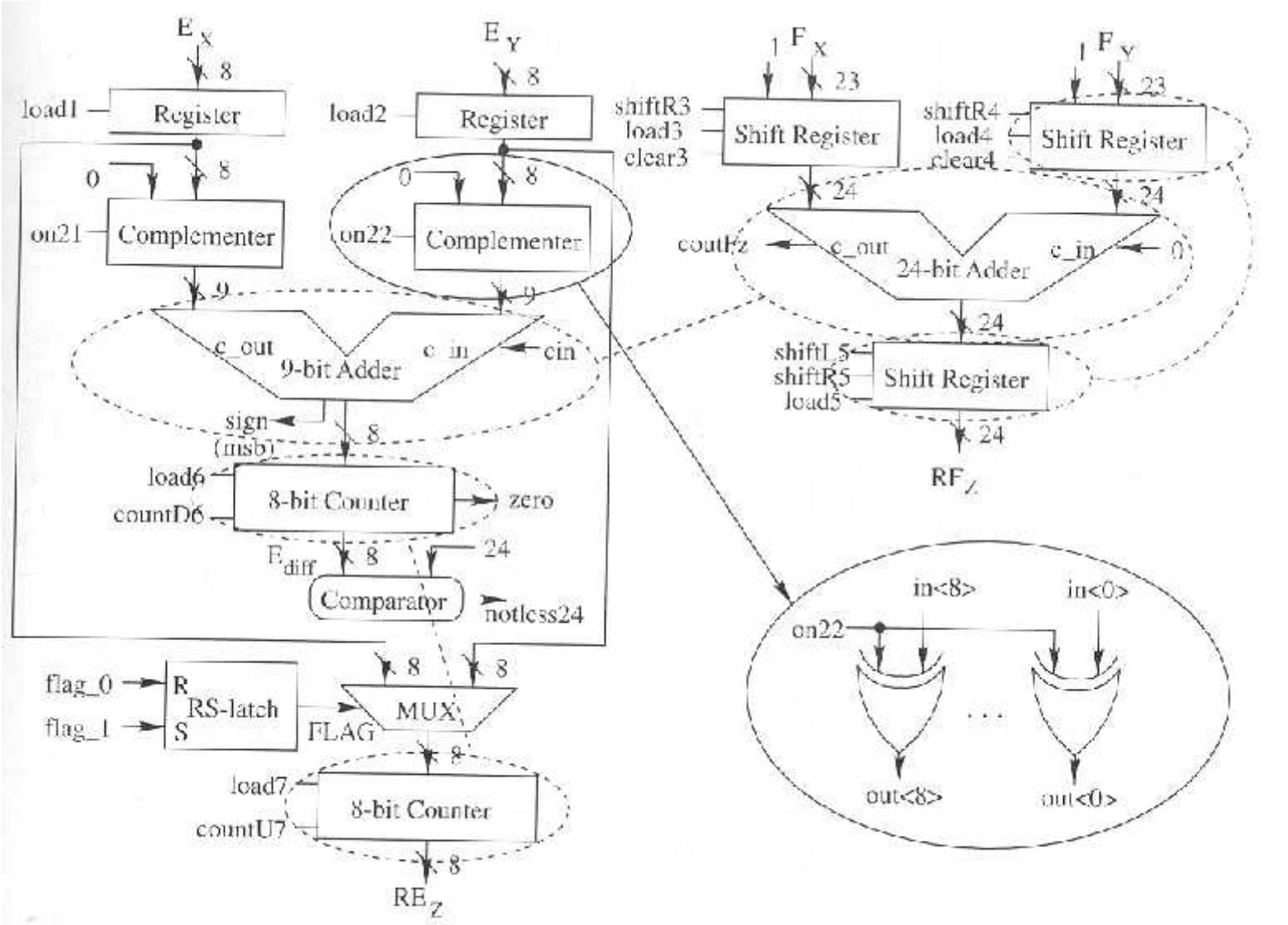
Desplazamiento a la izquierda

1. $RF_X \leftarrow 1.F_X$; $RF_Y \leftarrow 1.F_Y$;
   $RE_X \leftarrow E_X$; $RE_Y \leftarrow E_Y$;
2. $E_{diff} \leftarrow RE_X - RE_Y$;
   $FLAG \leftarrow 0$;
3. If $(E_{diff} < 0)$
   then begin
   $E_{diff} \leftarrow -E_{diff}$;
   $FLAG \leftarrow 1$;
   if $(E_{diff} < 24)$
   then $RF_X \leftarrow RF_X >> E_{diff}$;
   else $RF_X \leftarrow 0$;
   end;
   else begin
   if $(E_{diff} < 24)$
   then $RF_Y \leftarrow RF_Y >> E_{diff}$;
   else $RF_Y \leftarrow 0$;
   end;
4. $RF_Z \leftarrow RF_X + RF_Y$;
5. If $(FLAG = 1)$
   then $RE_Z \leftarrow RE_Y$;
   else $RE_Z \leftarrow RE_X$;
6. If $(RF_Z \geq 1)$
   then begin
   $RE_Z \leftarrow RE_Z + 1$;
   $RF_Z \leftarrow RF_Z >> 1$;
   end;

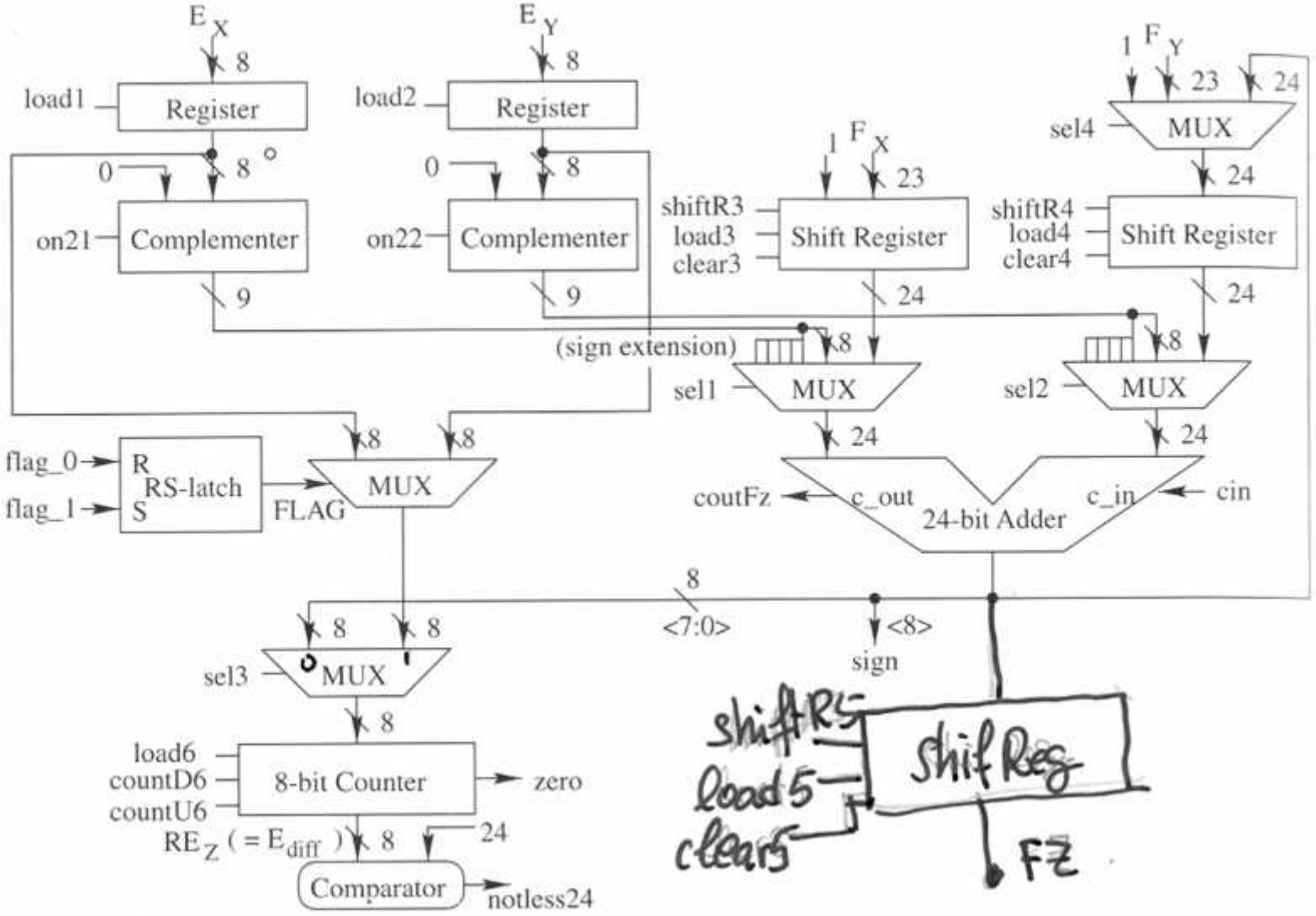
2.- Grafo ASM


s0
RF_X ← 1 @ F_X
RF_Y ← 1 @ F_Y
RE_X ← E_X
RE_Y ← E_Y
s1
E_diff ← RE_X - RE_Y
FLAG ← 0
E_diff < 0 ?
0
1
s2
E_diff ← RE_Y - RE_X
FLAG ← 1
E_diff >= 24 ?
0
1
s3
RF_X ← 0
E_diff = 0 ?
0
1
s4
E_diff ← E_diff - 1
RF_X ← RF_X >> 1
E_diff >= 24 ?
0
1
s5
RF_Y ← 0
E_diff = 0 ?
0
1
s6
E_diff ← E_diff - 1
RF_Y ← RF_Y >> 1
s7
RF_Z ← RF_X + RF_Y
if (FLAG) RE_Z ← RE_Y
else RE_Z ← RE_X
RF_Z >= 2 ?
0
1
s8
RE_Z ← RE_Z + 1
RF_Z ← RF_Z >> 1
s9
(done <- 1)

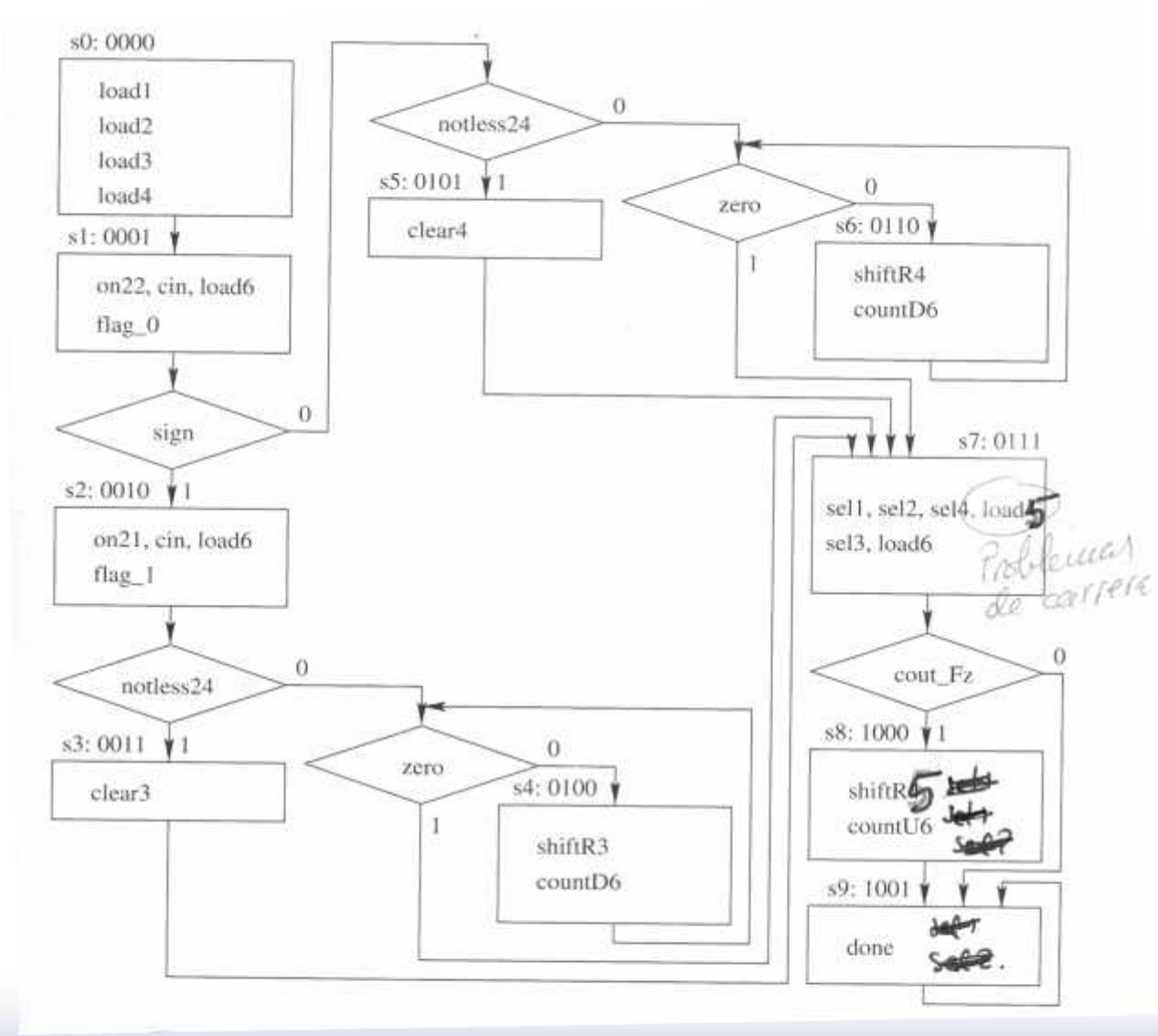
Data Path del Sumador



Otro Datapath del sumador



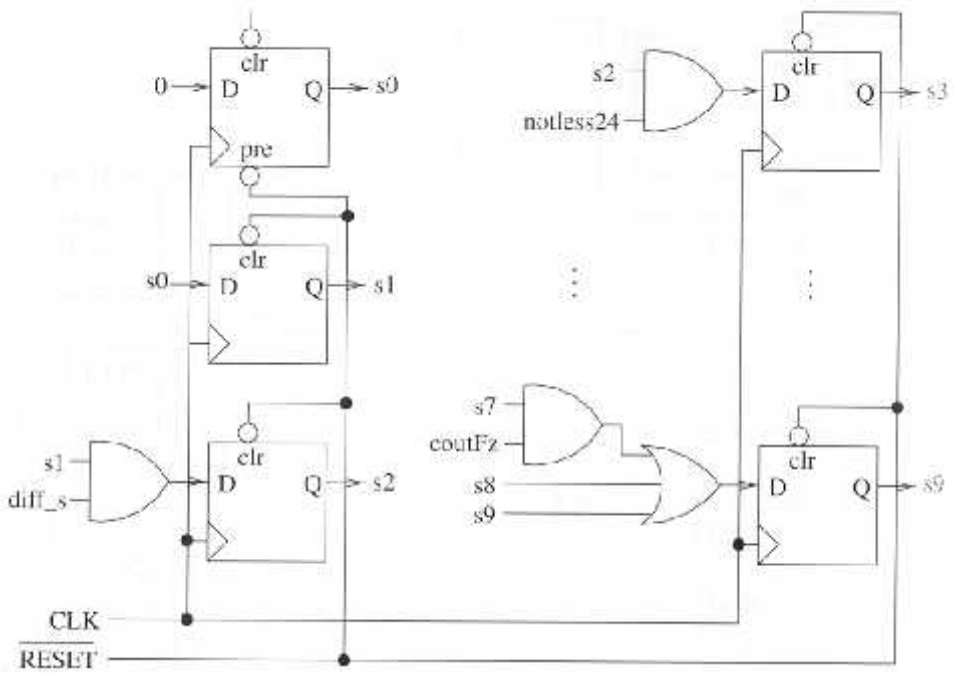
Grafo ASM del Controlador



Realización del controlador:
1- Un Flip Flop por estado

TABLE 4.7 THE INPUT CONDITION STATE TABLE FOR THE FLOATING-POINT ADDER.

Current State $C_3C_2C_1C_0$	Next State $N_3N_2N_1N_0$	Input Conditions	Outputs load1	load2	cin	...
0 0 0 0	0 0 0 1	1	1	1	0	...
0 0 0 1	0 0 1 0	$\overline{\text{sign}}$	0	0	1	...
0 0 0 1	0 1 0 1	$\overline{\text{sign}} \cdot \text{notless24}$	0	0	1	...
0 0 0 1	0 1 1 0	$\overline{\text{sign}} \cdot \overline{\text{notless24}} \cdot \overline{\text{zero}}$	0	0	1	...
0 0 0 1	0 1 1 1	$\overline{\text{sign}} \cdot \overline{\text{notless24}} \cdot \text{zero}$	0	0	1	...
0 0 1 0	0 0 1 1	notless24	0	0	1	...
0 0 1 0	0 1 0 0	$\overline{\text{notless24}} \cdot \overline{\text{zero}}$	0	0	1	...
0 0 1 0	0 1 1 1	$\overline{\text{notless24}} \cdot \text{zero}$	0	0	1	...
⋮	⋮	⋮	⋮			



Realización del controlador:
2- Utilización de estructuras PLD

TABLE 4.7 THE INPUT CONDITION STATE TABLE FOR THE FLOATING-POINT ADDER.

Current State $C_3C_2C_1C_0$	Next State $N_3N_2N_1N_0$	Input Conditions	Outputs load1	load2	cin	...
0 0 0 0	0 0 0 1	1	1	1	0	...
0 0 0 1	0 0 1 0	$\overline{\text{sign}}$	0	0	1	...
0 0 0 1	0 1 0 1	$\overline{\text{sign}} \cdot$ notless24	0	0	1	...
0 0 0 1	0 1 1 0	$\overline{\text{sign}} \cdot \overline{\text{notless24}} \cdot \overline{\text{zero}}$	0	0	1	...
0 0 0 1	0 1 1 1	$\overline{\text{sign}} \cdot \overline{\text{notless24}} \cdot$ zero	0	0	1	...
0 0 1 0	0 0 1 1	$\overline{\text{notless24}}$	0	0	1	...
0 0 1 0	0 1 0 0	$\overline{\text{notless24}} \cdot \overline{\text{zero}}$	0	0	1	...
0 0 1 0	0 1 1 1	$\overline{\text{notless24}} \cdot$ zero	0	0	1	...
⋮	⋮	⋮	⋮