

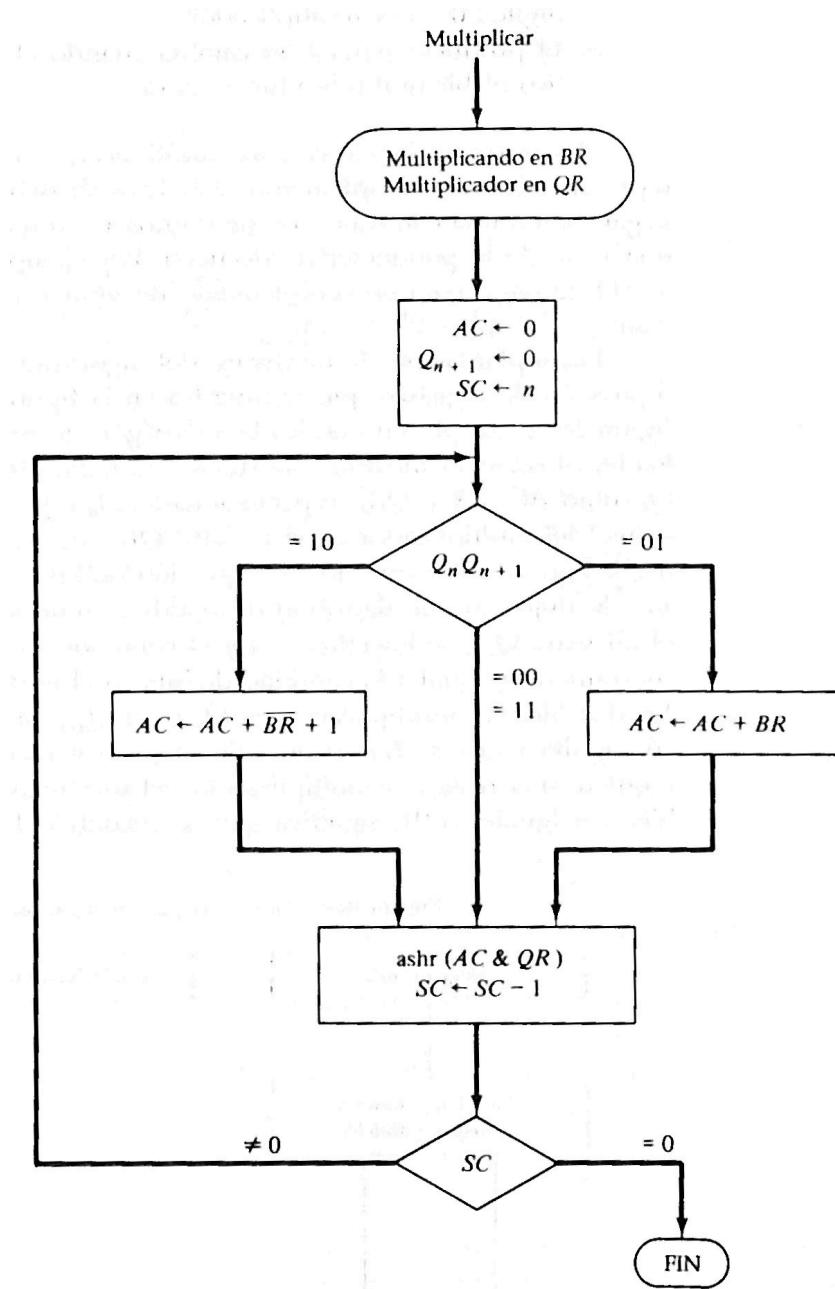
# Control cableado

## Algoritmo de Booth (MC2 m)

Condición	$\mu$ Operaciones	Siguiente
<b>Ciclo de búsqueda</b>		
t0	PC → MAR	SR+1 → SR
t1	M → GPR; PC+1 → PC	SR+1 → SR
t2	GPR(OP) → OPR	SR+1 → SR
<b>Ciclo de ejecución de MC2 m</b>		
i1·t3	GPR(AD) → MAR	SR+1 → SR
i1·t4	M → GPR	SR+1 → SR
i1·t5	AC ← 0; Qn+1 ← 0; SC ← n	SR+1 → SR
i1·t6	SC ← SC - 1	SR+1 → SR
i1·t6· $\bar{Q}_n \cdot Q_{n+1}$	AC ← AC+BR	-----
i1·t6· $Q_n \cdot \bar{Q}_{n+1}$	AC ← AC+ $\bar{B}R+1$	-----
i1·t7	Ashr (AC & QR)	-----
i1·t7·ZSC	-----	0 → SR
i1·t7· $\bar{Z}SC$	-----	6 → SR

## Expresiones de control

$\mu$ Operación	Expresión de control
PC → MAR	t0
M → GPR	t1 + i1·t4
PC+1 → PC	t1
GPR(OP) → OPR	t2
GPR(AD) → MAR	i1·t3
AC ← 0	i1·t5
$Q_n+1 \leftarrow 0$	i1·t5
SC ← n	i1·t5
SC ← SC - 1	i1·t6
AC ← AC+BR	i1·t6· $\bar{Q}_n \cdot Q_{n+1}$
AC ← AC+ $\bar{B}R+1$	i1·t6· $Q_n \cdot \bar{Q}_{n+1}$
Ashr (AC & QR)	i1·t7
SR+1 → SR	t0 + t1 + t2 + i1·t3 + i1·t4 + i1·t5 + i1·t6
Load SR	i1·t7·ZSC + i1·t7· $\bar{Z}SC =$ i1·t7



# Control microprogramado

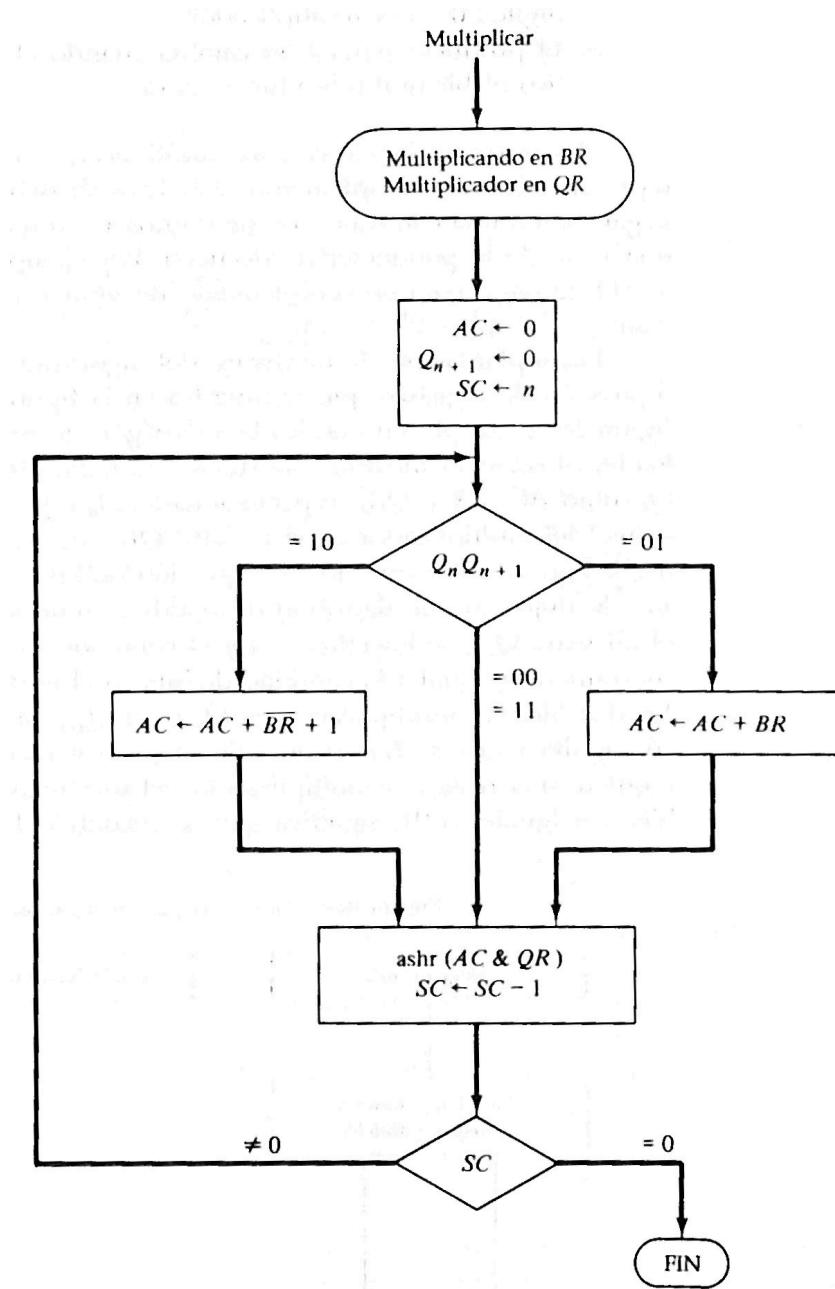
## Algoritmo de Booth (MC2 m)

Tabla de la CROM

Dirección	μOperaciones	Bits LCB	μDir. salto
<b>Ciclo de búsqueda</b>			
ADDR(FETCH)	PC → MAR	0 0 0	-----
ADDR(FETCH)+1	M → GPR; PC+1 → PC	0 0 0	-----
ADDR(FETCH)+2	GPR(OP) → OPR	1 1 1	-----
<b>Ciclo de ejecución de MC2 m</b>			
ADDR(MC2)	GPR(AD) → MAR	0 0 0	-----
ADDR(MC2)+1	M → GPR	0 0 0	-----
ADDR(MC2)+2	AC ← 0; Qn+1 ← 0; SC ← n	0 0 0	-----
ADDR(MC2)+3	SC ← SC - 1	0 1 0	ADDR(MC2)+7
ADDR(MC2)+4	-----	0 1 1	ADDR(MC2)+6
ADDR(MC2)+5	AC ← AC+BR	0 0 1	ADDR(MC2)+7
ADDR(MC2)+6	AC ← AC+ $\overline{BR}$ +1	0 0 0	-----
ADDR(MC2)+7	ashr (AC & QR)	1 0 0	ADDR(MC2)+3
ADDR(MC2)+8	-----	0 0 1	ADDR(FETCH)

Tabla de la LCB

S2	S1	S0	Qn	Qn+1	ZSC	I	B	R
0	0	0	X	X	X	1	0	0
0	0	1	X	X	X	0	1	0
1	1	1	X	X	X	0	0	1
0	1	0	0	0	X	0	1	0
0	1	0	0	1	X	1	0	0
0	1	0	1	0	X	1	0	0
0	1	0	1	1	X	0	1	0
0	1	1	0	X	X	1	0	0
0	1	1	1	X	X	0	1	0
1	0	0	X	X	0	0	1	0
1	0	0	X	X	1	1	0	0



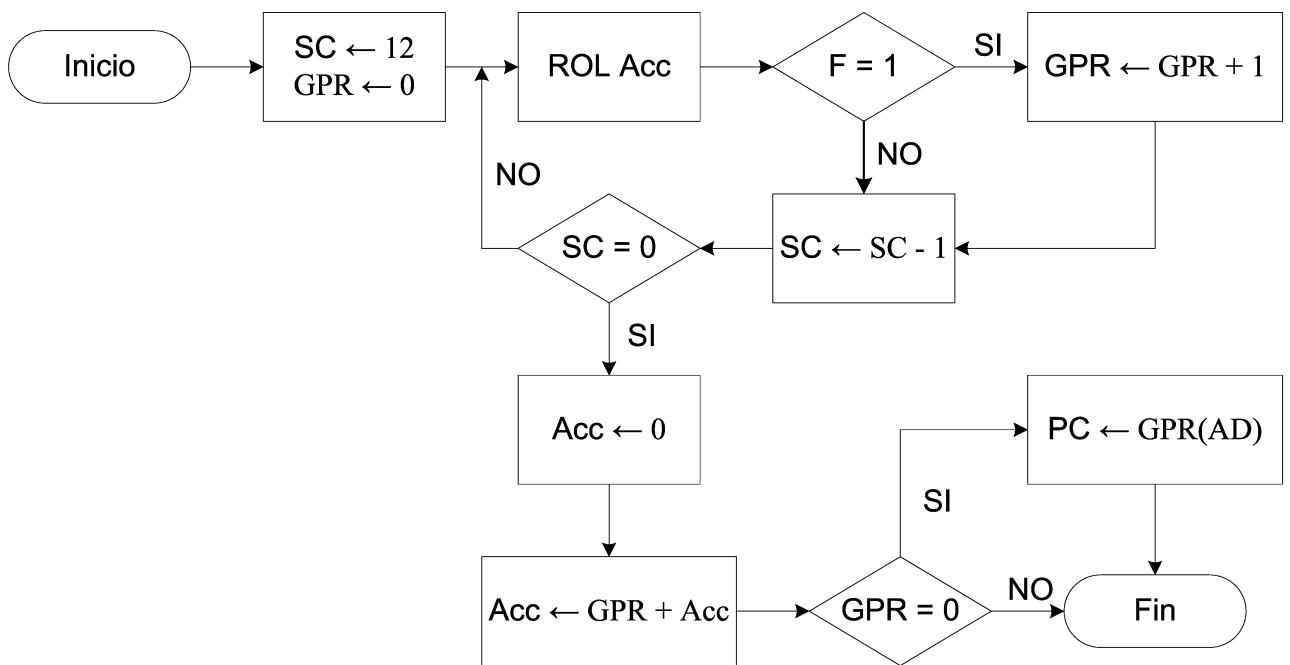
# Control cableado

## Instrucción CUA

Condición	$\mu$ Operaciones	Siguiente
<b>Ciclo de búsqueda</b>		
t0	PC → MAR	SR+1 → SR
t1	M → GPR; PC+1 → PC	SR+1 → SR
t2	GPR(OP) → OPR	SR+1 → SR
<b>Ciclo de ejecución de CUA</b>		
i1·t3	SC ← 12; GPR ← 0	SR+1 → SR
i1·t4	ROL Acc; SC ← SC - 1	SR+1 → SR
i1·t5·F	GPR ← GPR + 1	-----
i1·t5·ZSC	-----	SR+1 → SR
i1·t5·ZSC̄	-----	4 → SR
i1·t6	Acc ← 0	SR+1 → SR
i1·t7	Acc ← GPR + Acc	0 → SR
i1·t7·ZGPR	PC ← GPR(AD)	-----

## Expresiones de control

$\mu$ Operación	Expresión de control
PC → MAR	t0
M → GPR	t1 + i1·t4
PC+1 → PC	t1
GPR(OP) → OPR	t2
SC ← 12	i1·t3
GPR ← 0	i1·t3
ROL Acc	i1·t4
SC ← SC - 1	i1·t4
GPR ← GPR + 1	i1·t5·F
Acc ← 0	i1·t6
Acc ← GPR + Acc	i1·t7
PC ← GPR(AD)	i1·t7·ZGPR
SR+1 → SR	t0 + t1 + t2 + i1·t3 + i1·t4 + i1·t5·ZSC + i1·t6
Load SR	i1·t5·ZSC̄ + i1·t7



# Control microprogramado

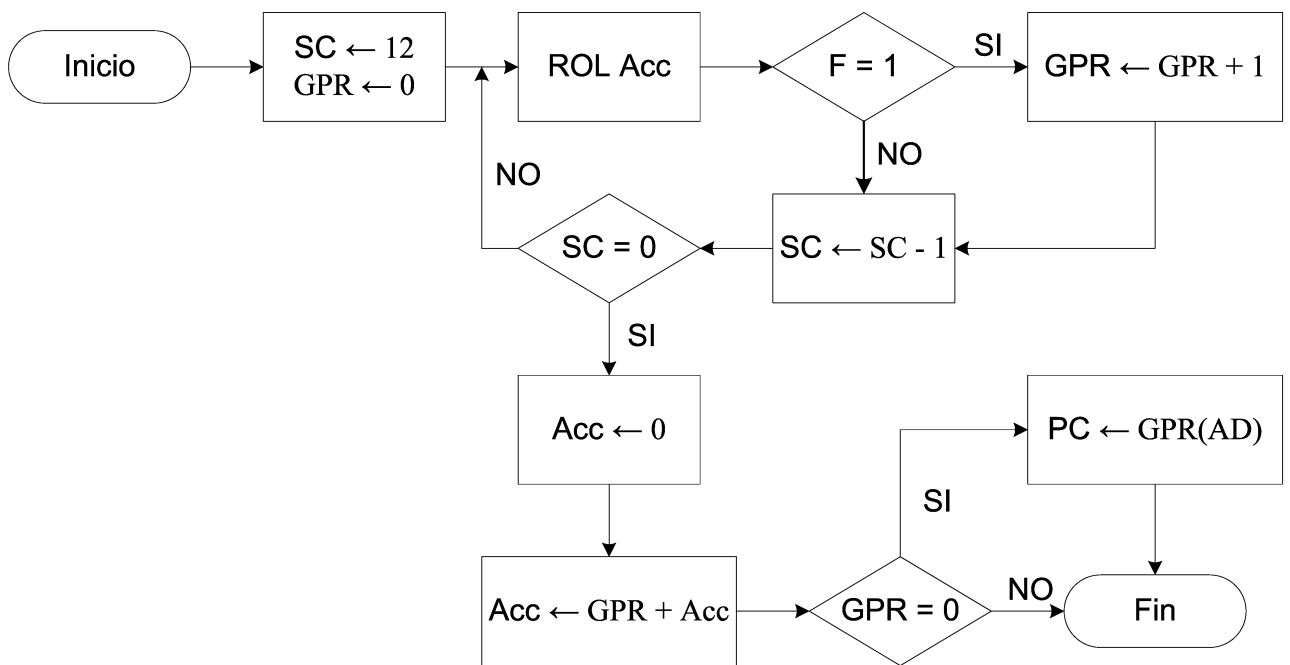
## Instrucción CUA

Tabla de la CROM

Dirección	μOperaciones	Bits LCB	μDir. salto
<b>Ciclo de búsqueda</b>			
ADDR(FETCH)	PC → MAR	0 0 0	-----
ADDR(FETCH)+1	M → GPR; PC+1 → PC	0 0 0	-----
ADDR(FETCH)+2	GPR(OP) → OPR	1 1 1	-----
<b>Ciclo de ejecución de CUA</b>			
ADDR(CUA)	SC ← 12; GPR ← 0	0 0 0	-----
ADDR(CUA)+1	ROL Acc; SC ← SC – 1	0 0 0	-----
ADDR(CUA)+2	-----	0 1 0	ADDR(CUA)+4
ADDR(CUA)+3	GPR ← GPR + 1	0 0 0	-----
ADDR(CUA)+4	-----	0 1 1	ADDR(CUA)+1
ADDR(CUA)+5	Acc ← 0	0 0 0	-----
ADDR(CUA)+6	Acc ← GPR + Acc	1 0 0	ADDR(FETCH)
ADDR(CUA)+7	PC ← GPR(AD)	0 0 1	ADDR(FETCH)

Tabla de la LCB

S2	S1	S0	F	ZGPR	ZSC	I	B	R
0	0	0	X	X	X	1	0	0
0	0	1	X	X	X	0	1	0
1	1	1	X	X	X	0	0	1
0	1	0	0	X	X	0	1	0
0	1	0	1	X	X	1	0	0
0	1	1	X	X	0	0	1	0
0	1	1	X	X	1	1	0	0
1	0	0	X	0	X	0	1	0
1	0	0	X	1	X	1	0	0



# Control cableado

## Instrucción SUB

Condición	$\mu$ Operaciones	Siguiente
<b>Ciclo de búsqueda</b>		
t0	PC → MAR	SR+1 → SR
t1	M → GPR; PC+1 → PC	SR+1 → SR
t2	GPR(OP) → OPR	SR+1 → SR
<b>Ciclo de ejecución de SUB</b>		
i1·t3	GPR(AD) → MAR; Acc ← $\overline{\text{Acc}}$	SR+1 → SR
i1·t4	M → GPR; Acc ← Acc + 1	SR+1 → SR
i1·t5	Acc ← GPR + Acc	SR+1 → SR
i1·t6	Acc ← $\overline{\text{Acc}}$	SR+1 → SR
i1·t7	Acc ← Acc + 1	0 → SR

## Expresiones de control

$\mu$ Operación	Expresión de control
PC → MAR	t0
M → GPR	t1 + i1·t4
PC+1 → PC	t1
GPR(OP) → OPR	t2
GPR(AD) → MAR	i1·t3
Acc ← $\overline{\text{Acc}}$	i1·t3 + i1·t6
Acc ← Acc + 1	i1·t4 + i1·t7
Acc ← GPR + Acc	i1·t5
SR+1 → SR	t0 + t1 + t2 + i1·t3 + i1·t4 + i1·t5 + i1·t6
Load SR	i1·t7

# Control microprogramado

## Instrucción SUB

### Tabla de la CROM

Dirección	μOperaciones	Bits LCB	μDir. salto
<b>Ciclo de búsqueda</b>			
ADDR(FETCH)	PC → MAR	0 0 0	-----
ADDR(FETCH)+1	M → GPR; PC+1 → PC	0 0 0	-----
ADDR(FETCH)+2	GPR(OP) → OPR	1 1 1	-----
<b>Ciclo de ejecución de SUB</b>			
ADDR(SUB)	GPR(AD) → MAR; Acc ← $\overline{\text{Acc}}$	0 0 0	-----
ADDR(SUB)+1	M → GPR; Acc ← Acc + 1	0 0 0	-----
ADDR(SUB)+2	Acc ← GPR + Acc	0 0 0	-----
ADDR(SUB)+3	Acc ← $\overline{\text{Acc}}$	0 0 0	-----
ADDR(SUB)+4	Acc ← Acc + 1	0 0 1	ADDR(FETCH)

### Tabla de la LCB

S2	S1	S0	I	B	R
0	0	0	1	0	0
0	0	1	0	1	0
1	1	1	0	0	1

# Control cableado

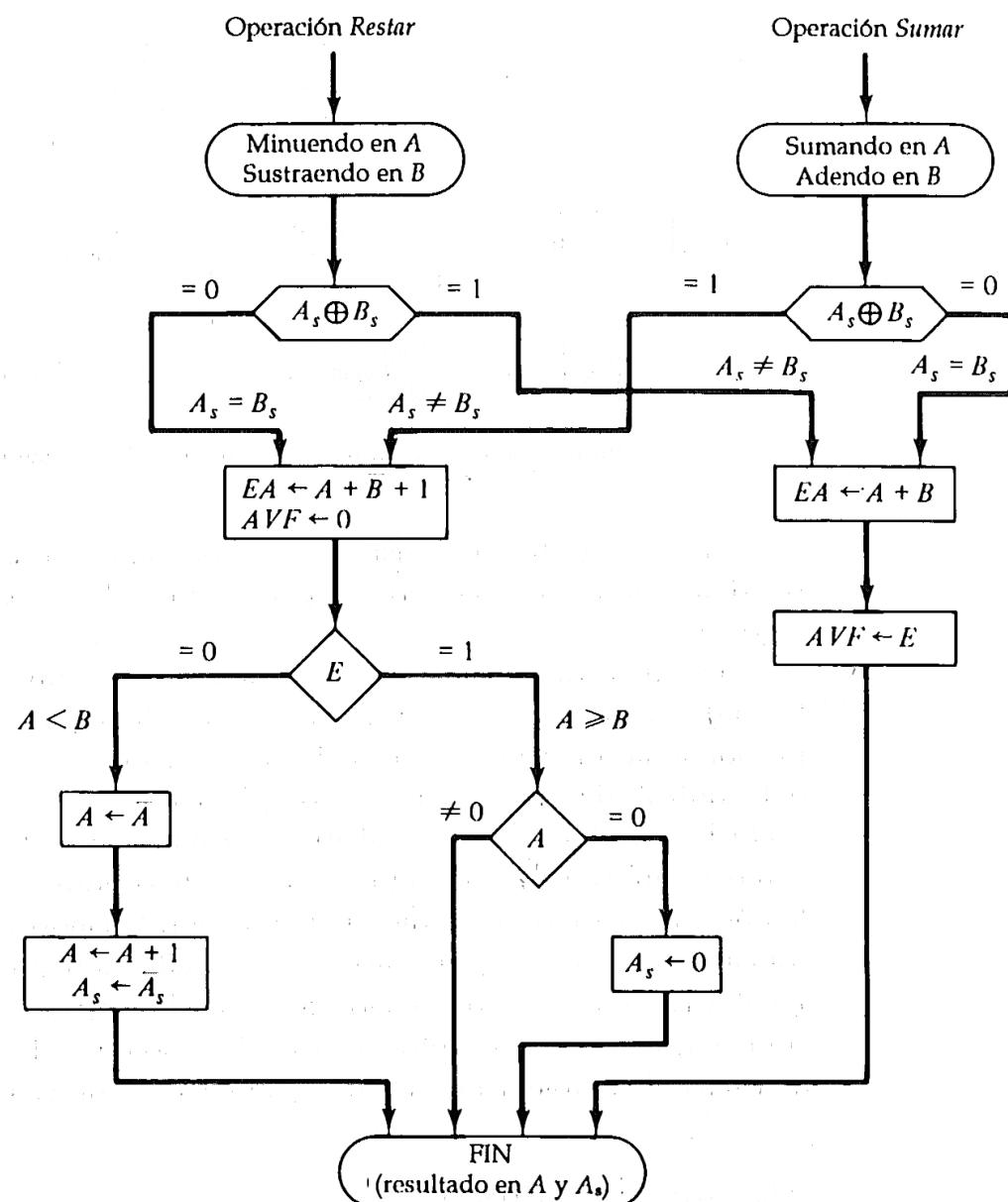
## Algoritmo de suma/resta en signo-magnitud (SRSM m)

Condición	$\mu$ Operaciones	Siguiente
<b>Ciclo de búsqueda</b>		
t0	PC → MAR	SR+1 → SR
t1	M → GPR; PC+1 → PC	SR+1 → SR
t2	GPR(OP) → OPR	SR+1 → SR
<b>Ciclo de ejecución de SRSM m</b>		
i1·t3	GPR(AD) → MAR	SR+1 → SR
i1·t4	M → GPR	SR+1 → SR
i1·t5·( $\overline{OP \oplus RS}$ )	EA ← A+B	SR+1 → SR
i1·t5·( $OP \oplus RS$ )	EA ← A+ $\overline{B}$ +1; AVF ← 0	7 → SR
i1·t6	AVF ← E	0 → SR
i1·t7·E	-----	0 → SR
i1·t7·E·ZA	AS ← 0	-----
i1·t7· $\overline{E}$	A ← $\overline{A}$	SR+1 → SR
i1·t8	A ← A+1; AS ← $\overline{AS}$	0 → SR

RS ← AS  $\oplus$  BS  
OP (0 → Suma; 1 → Resta)

## Expresiones de control

$\mu$ Operación	Expresión de control
PC → MAR	t0
M → GPR	t1 + i1·t4
PC+1 → PC	t1
GPR(OP) → OPR	t2
GPR(AD) → MAR	i1·t3
EA ← A+B	i1·t5·( $\overline{OP \oplus RS}$ )
EA ← A+ $\overline{B}$ +1	i1·t5·( $OP \oplus RS$ )
AVF ← 0	i1·t5·( $OP \oplus RS$ )
AVF ← E	i1·t6
AS ← 0	i1·t7·E·ZA
A ← $\overline{A}$	i1·t7· $\overline{E}$
A ← A+1	i1·t8
AS ← $\overline{AS}$	i1·t8
SR+1 → SR	t0 + t1 + t2 + i1·t3 + i1·t4 + i1·t5·( $\overline{OP \oplus RS}$ ) + i1·t7· $\overline{E}$
Load SR	i1·t5·( $OP \oplus RS$ ) + i1·t6 + i1·t7·E + i1·t8



# Control microprogramado

## Algoritmo de suma/resta en signo-magnitud (SRSM m)

Tabla de la CROM

Dirección	μOperaciones	Bits LCB	μDir. salto
<b>Ciclo de búsqueda</b>			
ADDR(FETCH)	PC → MAR	0 0 0	-----
ADDR(FETCH)+1	M → GPR; PC+1 → PC	0 0 0	-----
ADDR(FETCH)+2	GPR(OP) → OPR	1 1 1	-----
<b>Ciclo de ejecución de SRSM m</b>			
ADDR(SRSM)	GPR(AD) → MAR	0 0 0	-----
ADDR(SRSM)+1	M → GPR	0 0 0	-----
ADDR(SRSM)+2	-----	0 1 0	ADDR(SRSM)+5
ADDR(SRSM)+3	EA ← A + B	0 0 0	-----
ADDR(SRSM)+4	AVF ← E	0 0 1	ADDR(FETCH)
ADDR(SRSM)+5	EA ← A + $\bar{B}$ + 1; AVF ← 0	0 0 0	-----
ADDR(SRSM)+6	-----	0 1 1	ADDR(SRSM)+9
ADDR(SRSM)+7	A ← $\bar{A}$	0 0 0	-----
ADDR(SRSM)+8	A ← A+1; AS ← $\bar{AS}$	0 0 1	ADDR(FETCH)
ADDR(SRSM)+9	-----	1 0 0	ADDR(FETCH)
ADDR(SRSM)+10	AS ← 0	0 0 1	ADDR(FETCH)

Tabla de la LCB

S2	S1	S0	OP	RS	E	ZA	I	B	R
0	0	0	X	X	X	X	1	0	0
0	0	1	X	X	X	X	0	1	0
1	1	1	X	X	X	X	0	0	1
0	1	0	0	0	X	X	1	0	0
0	1	0	0	1	X	X	0	1	0
0	1	0	1	0	X	X	0	1	0
0	1	0	1	1	X	X	1	0	0
0	1	1	X	X	0	X	1	0	0
0	1	1	X	X	1	X	0	1	0
1	0	0	X	X	X	0	0	1	0
1	0	0	X	X	X	1	1	0	0

RS ← AS  $\oplus$  BS  
 OP (0 → Suma; 1 → Resta)

