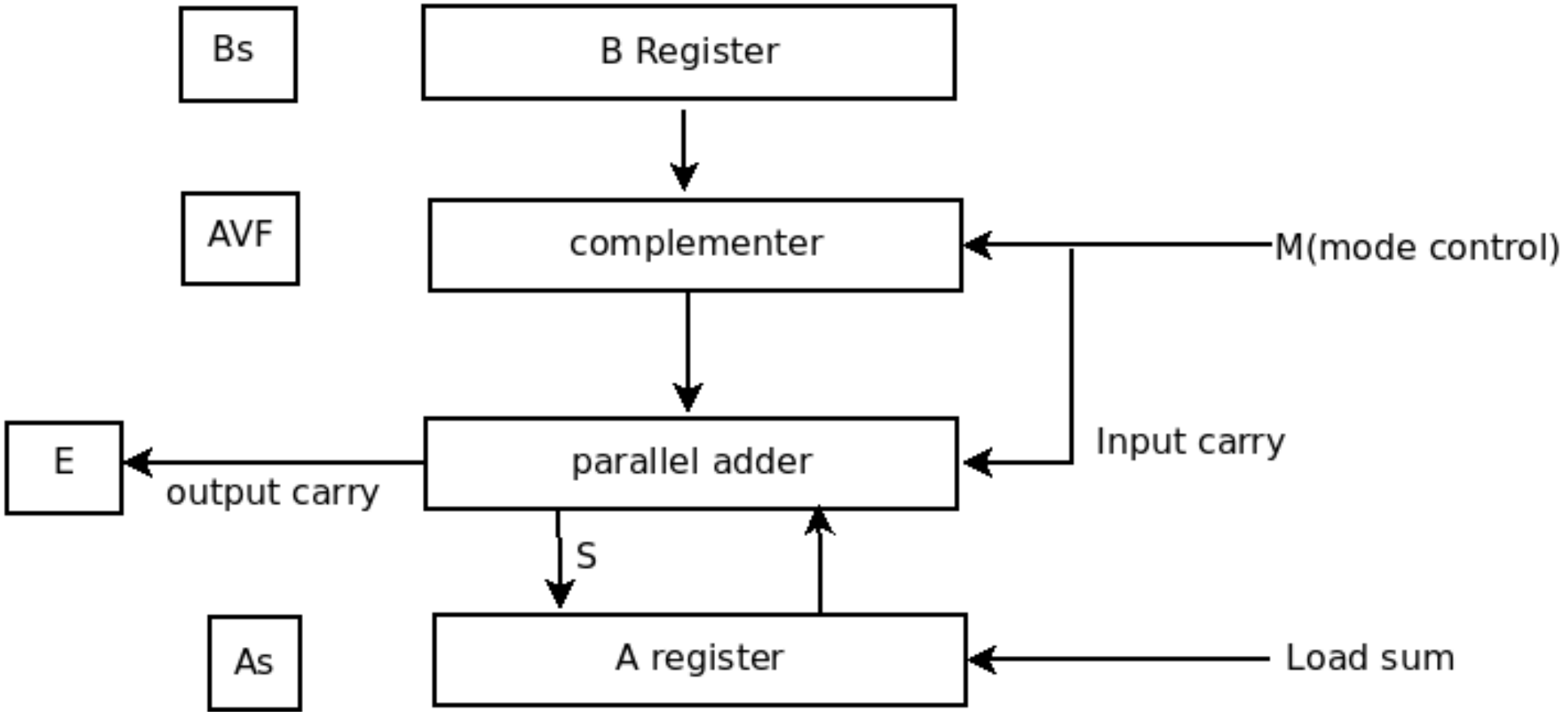
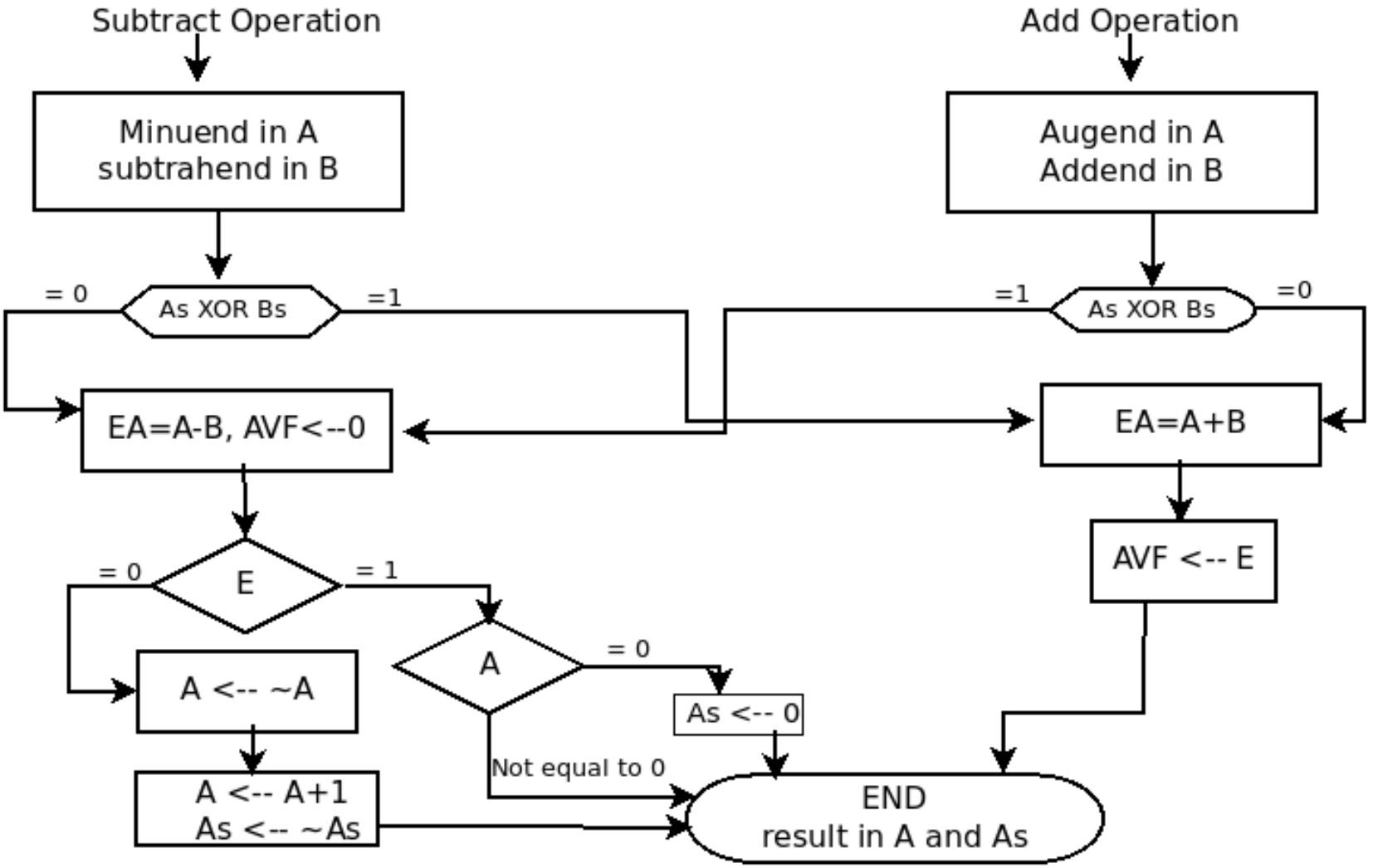
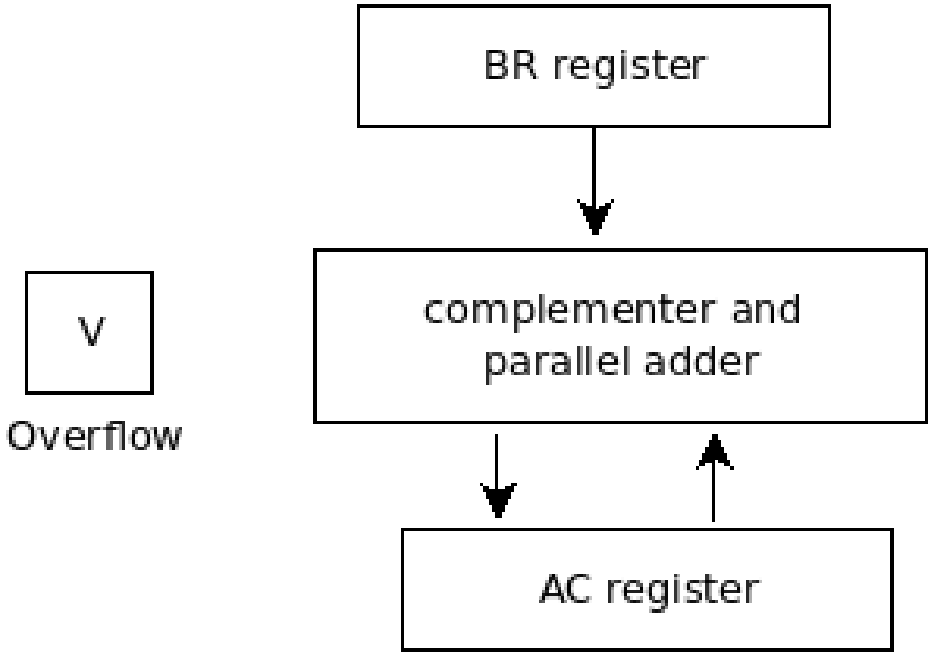
Signed magnitude addition or sub H/W Implementation

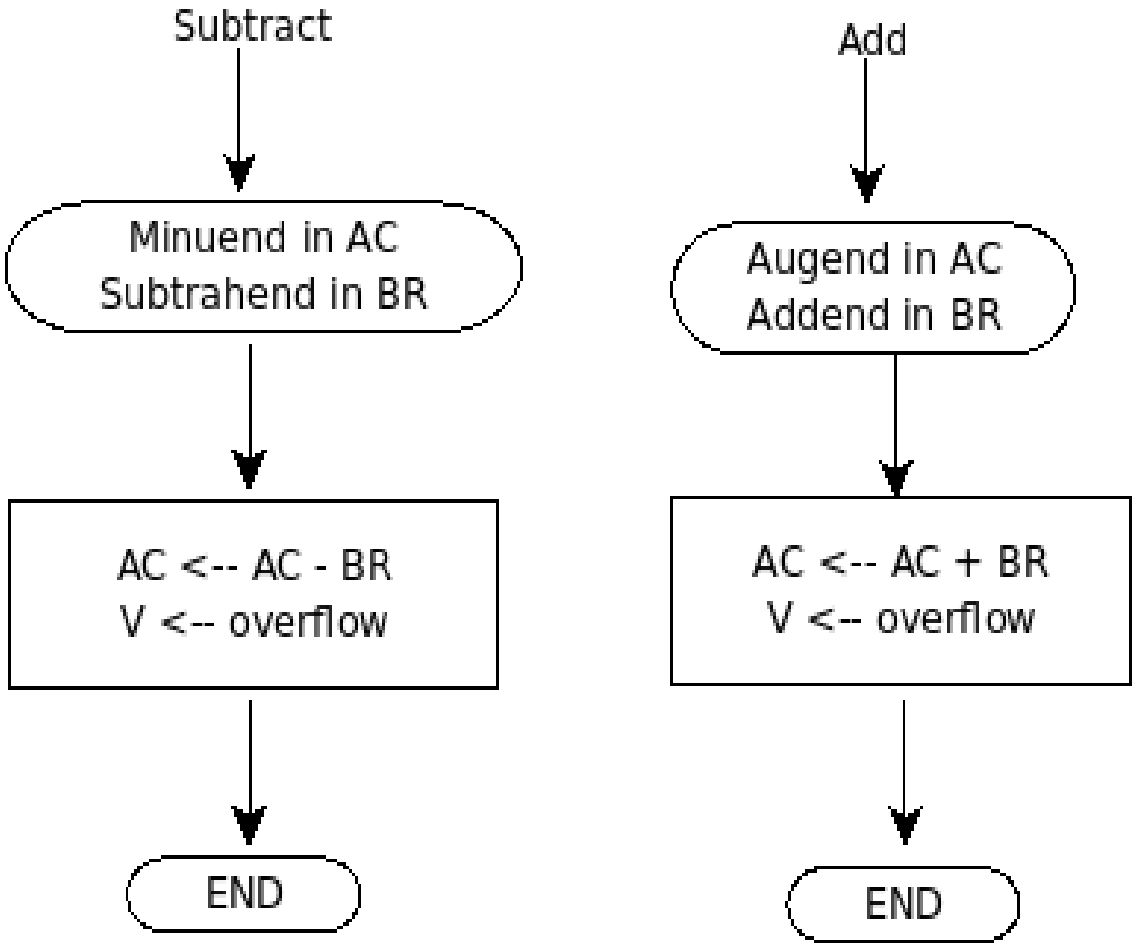


Signed magnitude addition or subtraction flowchart

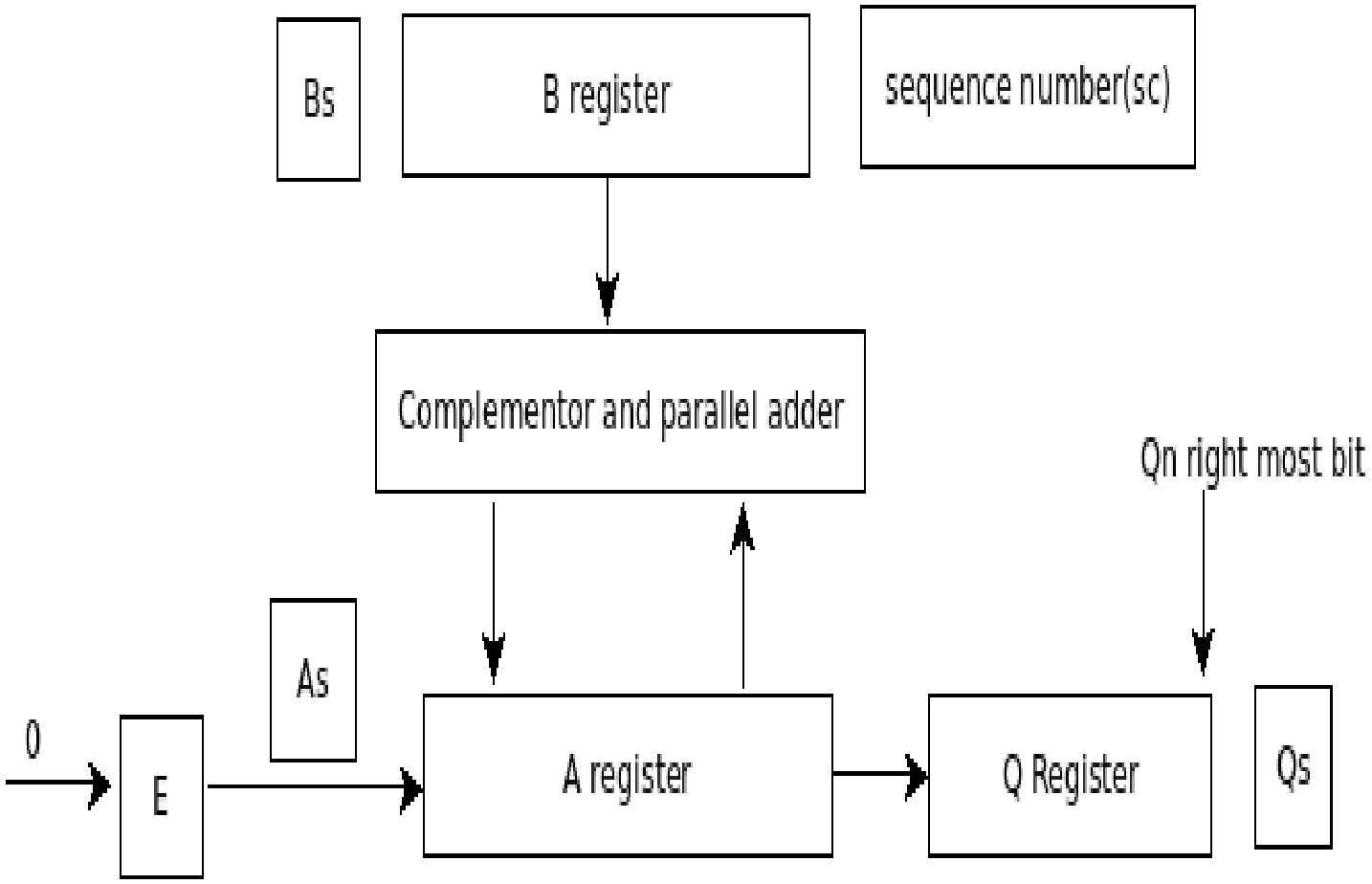


2’s complement addition or subtraction

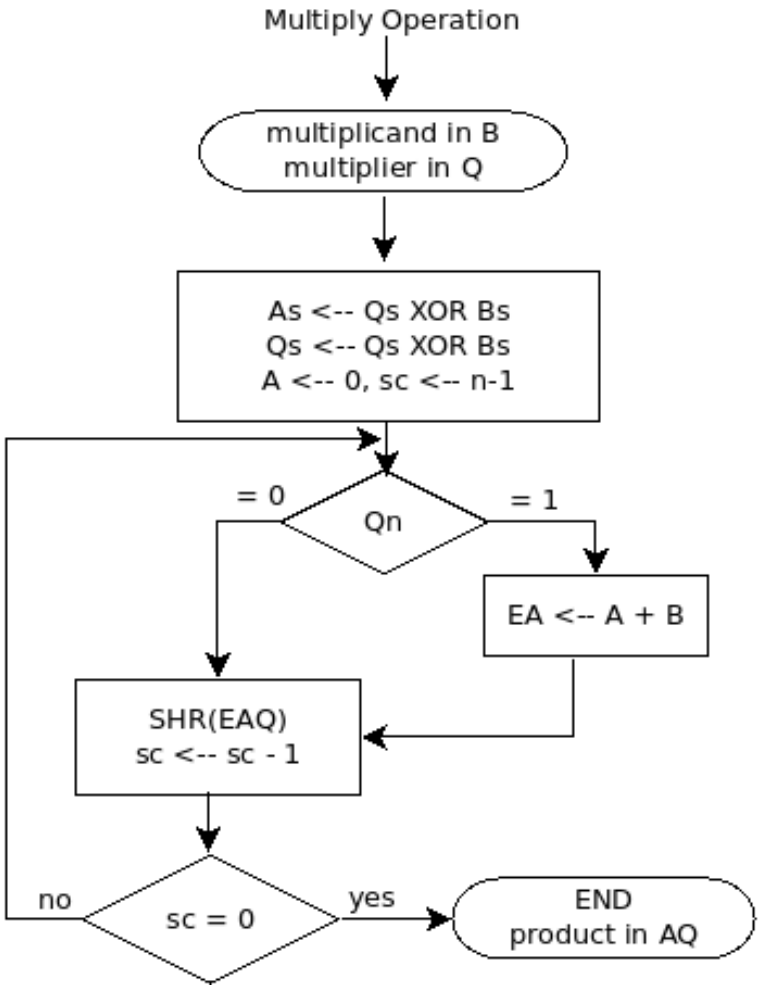



Subtract
Minuend in AC
Subtrahend in BR
AC <-- AC - BR
V <-- overflow
END
Add
Augend in AC
Addend in BR
AC <-- AC + BR
V <-- overflow
END

Signed magnitude multiplication H/W implementation



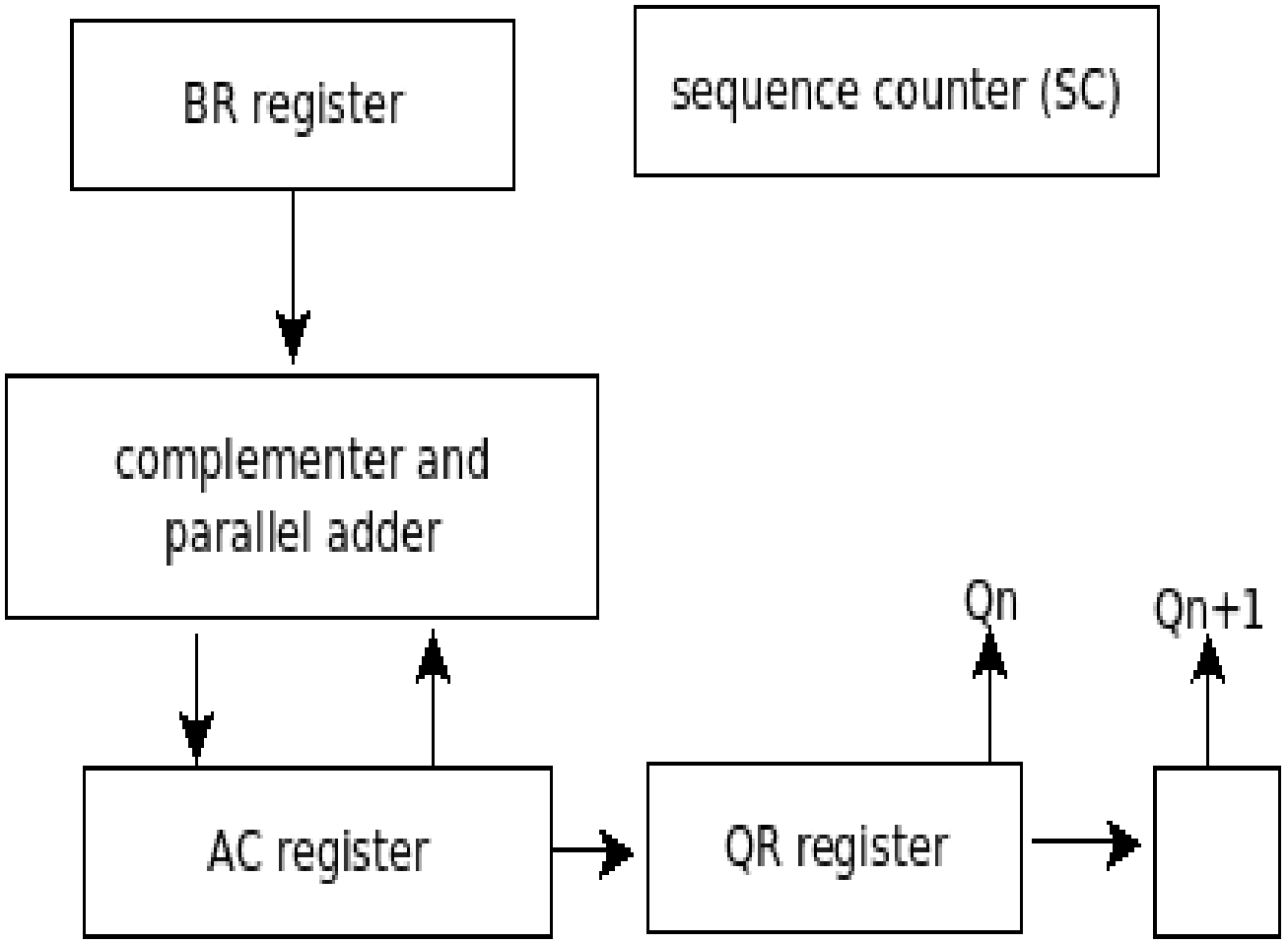
Signed magnitude multiplication flowchart



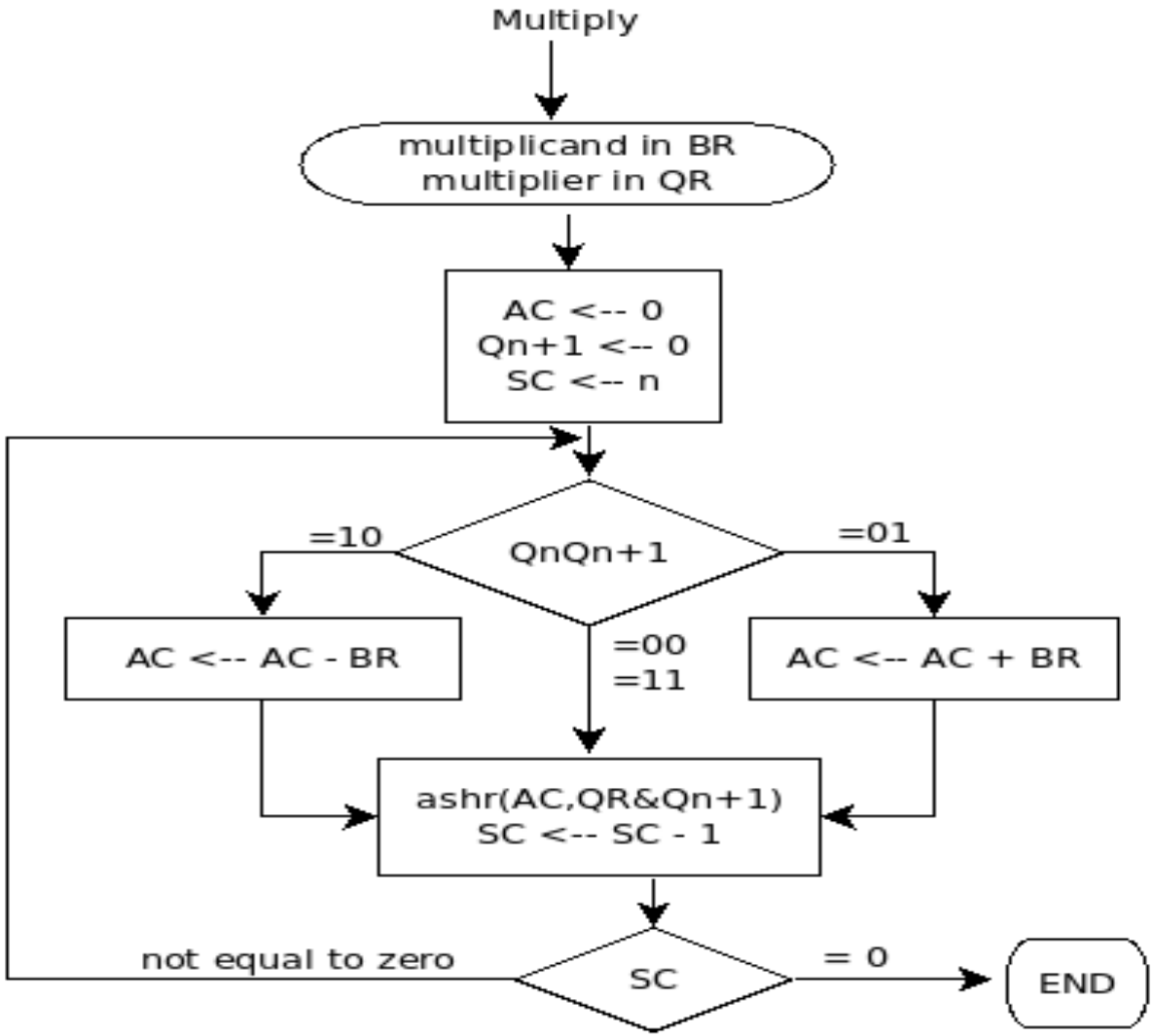
signed magnitude numbers multiplication example (-6*3)

Multiplicand in B=0110	E	A	Q	SC
Multiplier in Q	0	0000	0011	4
Qn = 1; add B to A		0110		
First partial product in A	0	0110	0011	
Shift right(EAQ)	0	0011	0001	3
Qn=1; add B to partial product A		0110		
Partial product in A	0	1001	0001	
Shift right (EAQ)	0	0100	1000	2
Qn=0; shift right(EAQ)	0	0010	0100	1
Qn=0; shift right(EAQ)	0	0001	0010	0

Booth's multiplication H/W implementation



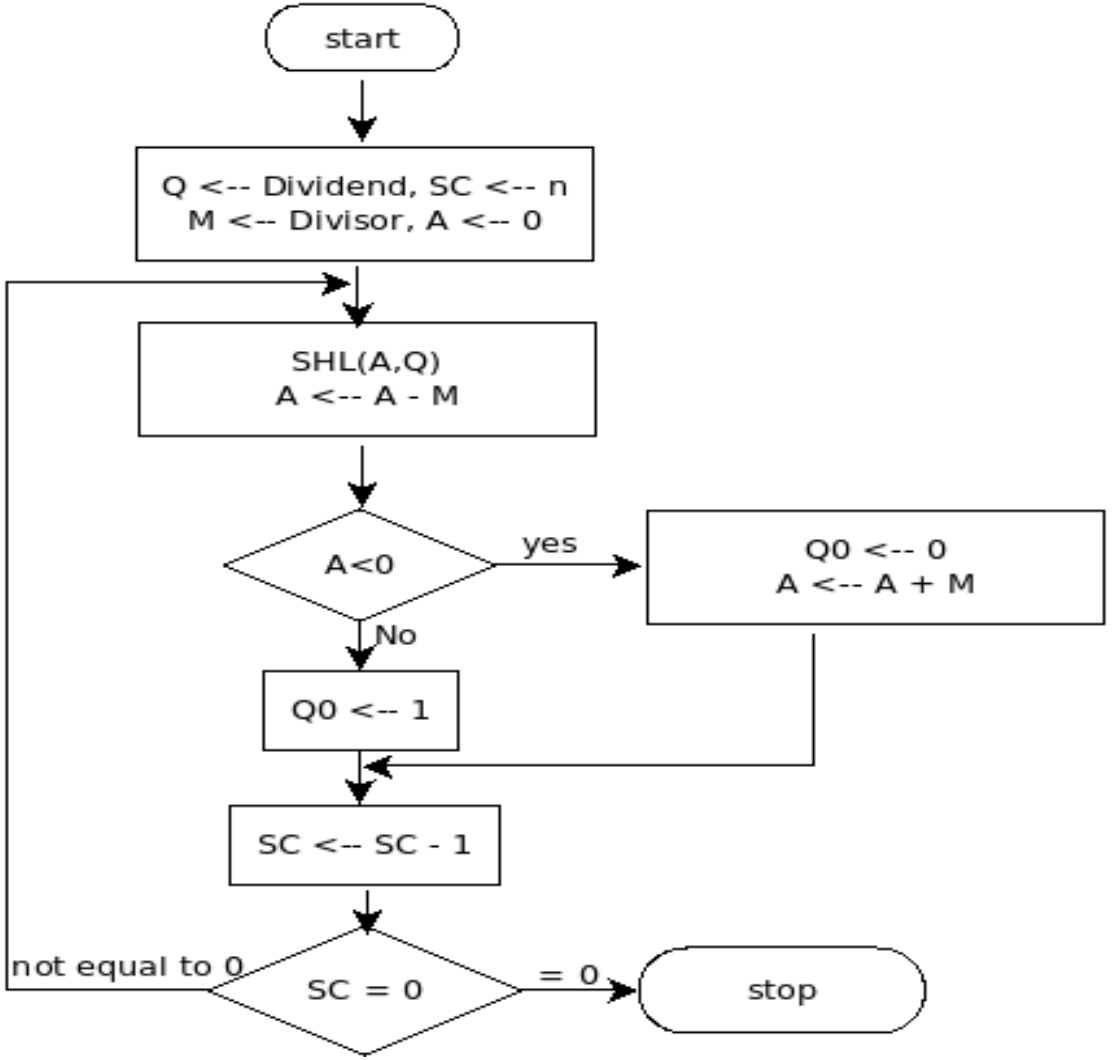
Booth’s multiplication flowchart



Booth's multiplication example (-5*-4)

QnQn+1	BR=1011 2's comp of BR = 0101	AC	QR	Qn+1	SC
	Initial	0000	1100	0	4
0 0	ashr(AC,QR,Qn+1)	0000	0110	0	3
0 0	ashr(AC,QR,Qn+1)	0000	0011	0	2
1 0	subtract	0101			
		0101			
	ashr(AC,QR,Qn+1)	0010	1001	1	1
1 1	ashr(AC,QR, Qn+1)	0001	0100	1	0

Restoring division algorithm flowchart

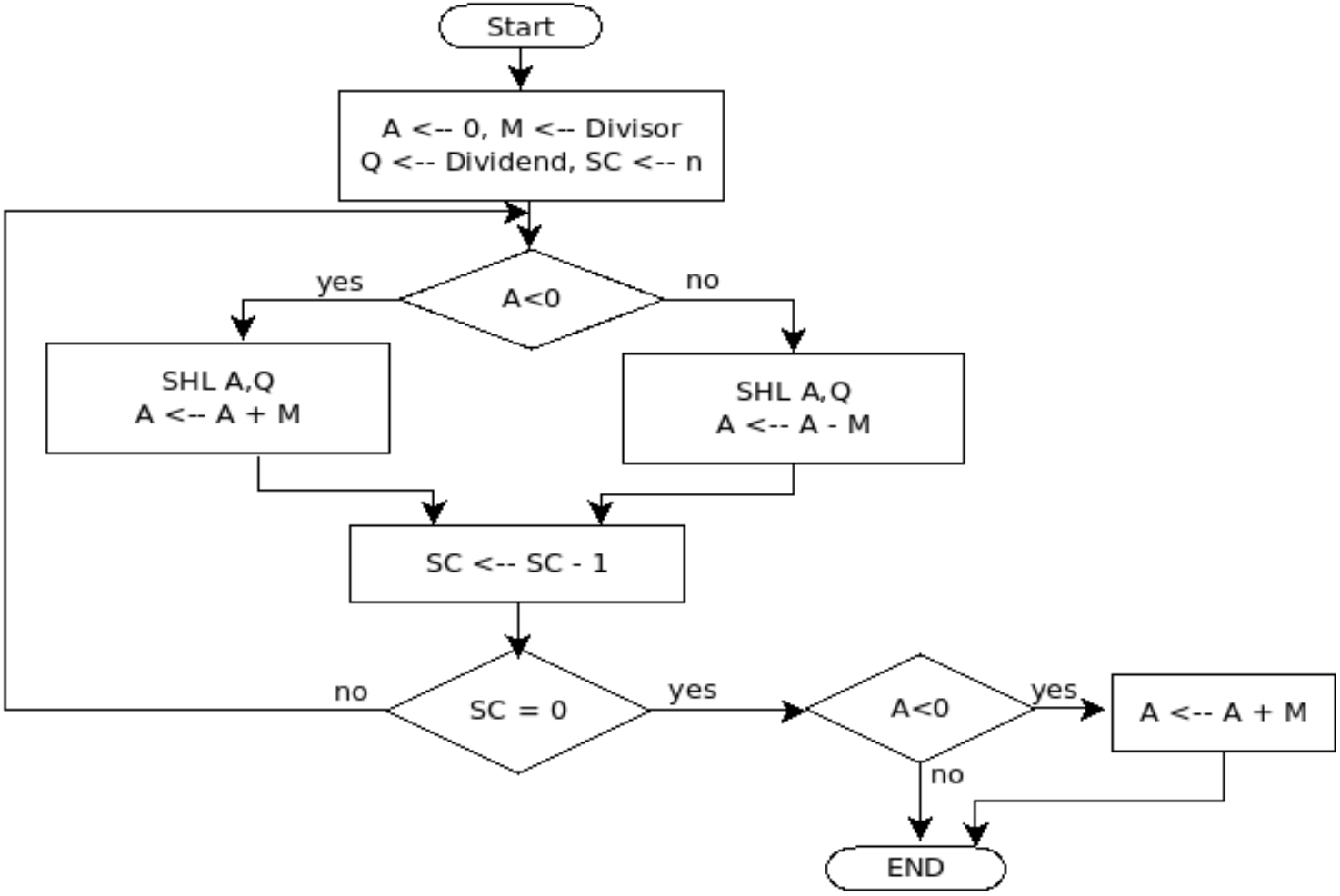


Restoring division example

7/3 M divisor binary value is 0011, 2's complement of M value is 1101

Steps	A	Q	SC
Initial	0000	0111	4
Shift left (A,Q)	0000	1110	
subtract	1101		
A<0; Q0<--0	1101	1110	
add M to A	0011		
	0000	1110	3
Shift left (A,Q)	0001	1100	
subtract(A <-- A - M)	1101		
A<0 ; restore Partial Remainder	1110		
	0011		
A=A+M	0001	1100	2
Shift left (A,Q)	0011	1000	
subtract	1101		
Carry is discarded	0000		
A is not less than zero; Q0<-- 1	0000	1001	1
Shift left (A,Q)	0001	0010	
Subtract (A=A-M)	1101		
A<0 is true then Q0<--0;	1110		
Restore partial remainder	0011		
	0001	0010	0

Non restoring division algorithm flowchart



Non restoring division algorithm(7/3)

Steps	A	Q	SC
Initial	0000	0111	4
Shift left (A,Q)	0000	1110	
Subtract (A=A-M)	1101		
	1101		
First bit 1 means A<0; q0<--0	1101	1110	3
Shift left (A,Q), A=A+M	1011	1100	
	0011		
A<0 is true then q0<--0	1110	1100	2
Shift left (A,Q)	1101	1000	
Add A=A+M	0011		
	0000		
A<0 is false then q0<--1	0000	1001	1
Shift left (A,Q)	0001	0010	
Subtract (A=A-M)	1101		
A<0 is true then q0<--0	1110	0010	0
If sc is 0 and A<0 is true then add A=A+M	0011		
	0001	0010	