Computer Arithmetic
Addition and Subtraction

Addition and Subtraction with Signed magnitude data

- There are 3 ways of representing negative fixed point binary numbers. They are
  1. Signed Magnitude representation.
  2. Signed one's Complement representation
  3. Signed two's Complement representation
- Most computers use the Signed two's Complement representation when performing operation on integers.
- Consider the magnitude of any two numbers A and B and the eight different operation are listed below depending on the sign of the number.

Eight Conditions for Signed-Magnitude Addition/Subtraction

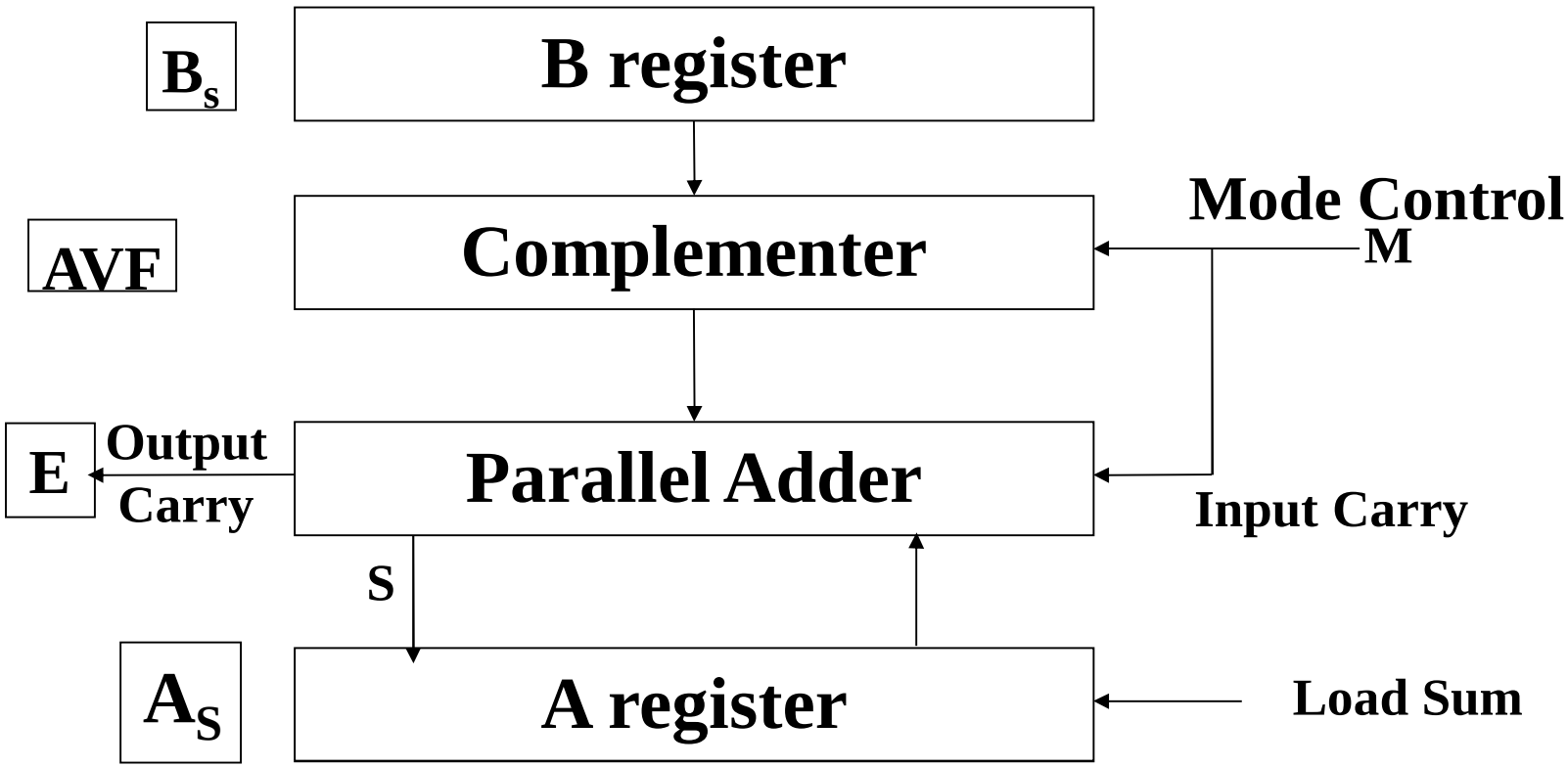
	Operation	ADD Magnitudes	SUBTRACT Magnitudes		
			A > B	A < B	A = B
1	(+A) + (+B)	+ (A + B)			
2	(+A) + (-B)		+ (A – B)	- (B – A)	+ (A – B)
3	(-A) + (+B)		- (A – B)	+ (B – A)	+ (A – B)
4	(-A) + (-B)	- (A + B)			
5	(+A) - (+B)		+ (A – B)	- (B – A)	+ (A – B)
6	(+A) - (-B)	+ (A + B)			
7	(-A) - (+B)	- (A + B)			
8	(-A) - (-B)		- (A – B)	+ (B – A)	+ (A – B)

Addition and Subtraction with Signed magnitude data

ALGORITHM:

- When the sign of A and B are identical, add the two magnitudes and attach the sign of A to the result.
- When the sign of A and B are different, compare the magnitudes, subtract smaller number from the larger.
- Choose the sign of the result to be same as A if A>B or complement the sign of A if A<B.
- If the two magnitudes are equal, subtract B from A and Make the sign of the result positive.

Hardware for signed-magnitude addition and subtraction



Hardware for signed-magnitude addition and subtraction

- Let A and B be the two registers that holds the magnitudes of the numbers and As and Bs be two flipflops that holds the corresponding sign
- The result of the operation may be transferred to the third register or the result is transferred to A and As.
- First parallel adder is needed to perform microoperation A+B.
- Second comparator circuit needed to establish if A<B, A>B or A==B.
- Third subtractor circuit is needed to perform the microoperation A-B and B-A.

Hardware for signed-magnitude addition and subtraction

- The block diagram consist of register A and B and the sign flipflops As and Bs. Subtraction is done by adding A to the 2's complement of B.
- The o/p carry is transferred to E. The add overflow flipflop(AVF) holds the overflow bit when A and B are added.
- The addition A+B is done through the parallel adder and the sum is transferred to A register.
- When the Mode bit M=0 the o/p of B is transferred to the adder, the i/p carry is 0 and the o/p of the adder is equal to sum A+B
- When M=1, the 1's complement of B is applied to adder, the i/p carry is 1 and the o/p is equal to A+ B'+1.

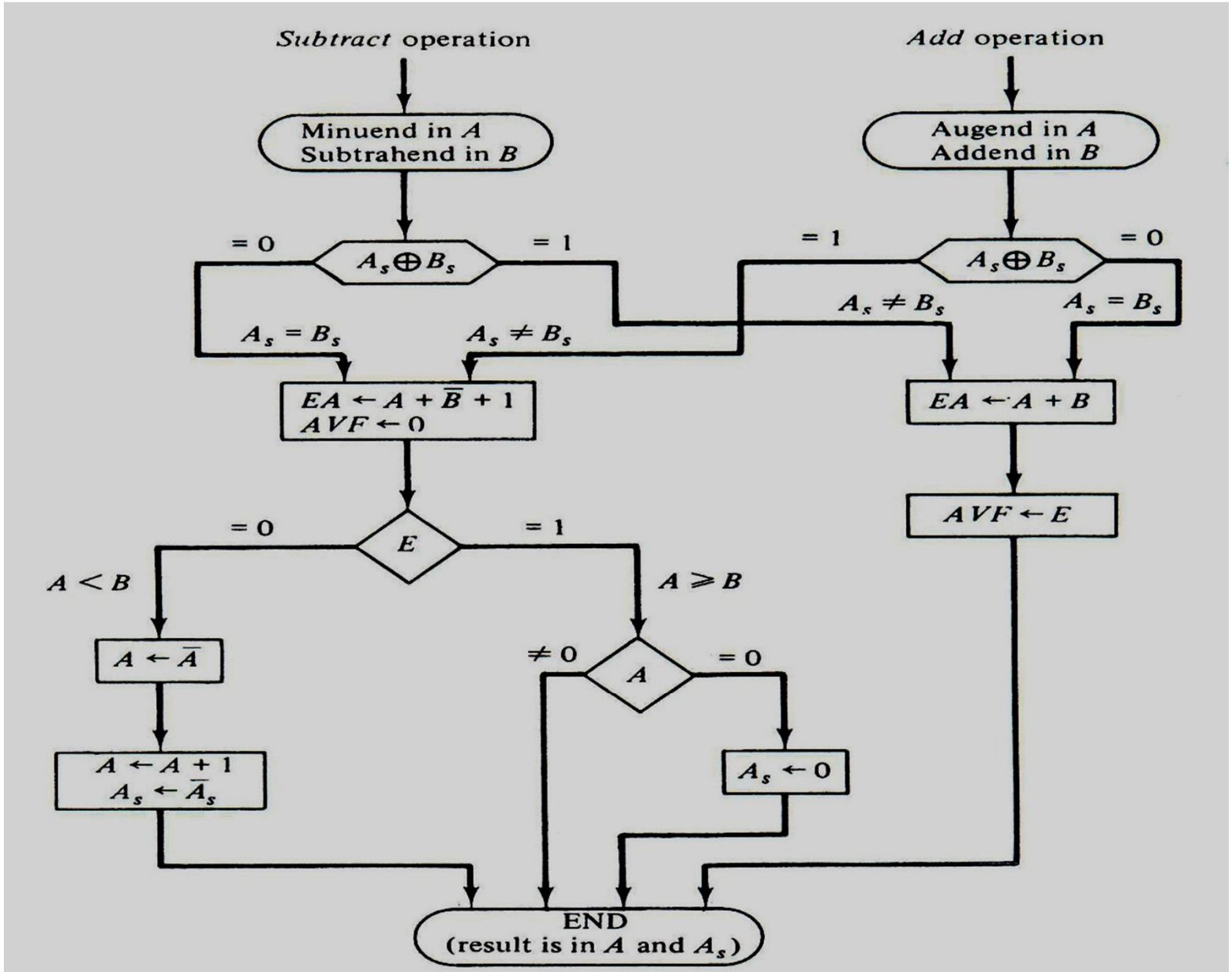


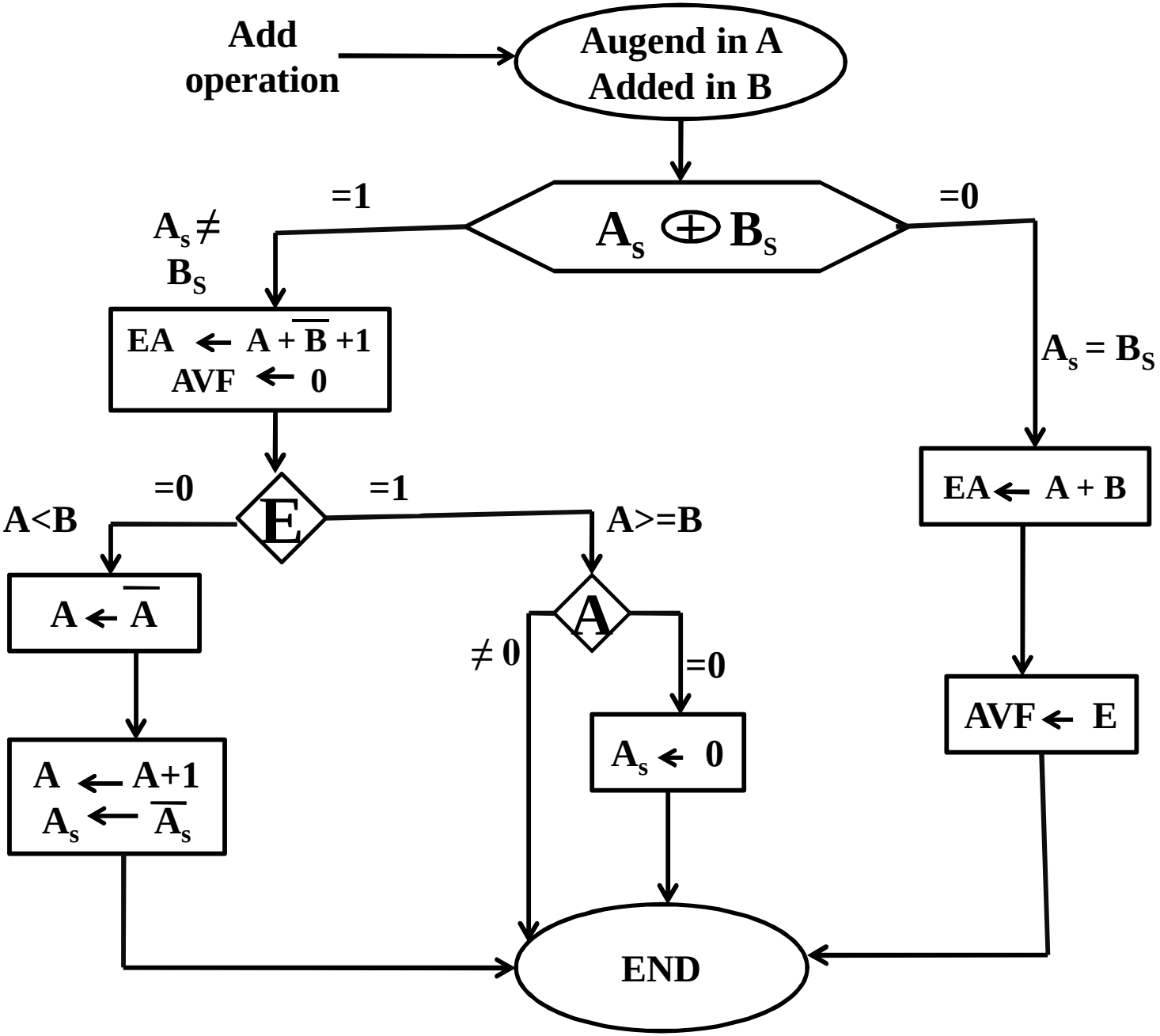
Figure 10-2 Flowchart for add and subtract operations.

Hardware Algorithm

- The two sign bits As and Bs are compared by XOR gate. If the o/p is 0, the sign are identical and if the o/p is 1, the sign are different.
- For an add operation the identical sign indicates that magnitudes are to be added.
- For the subtraction operation different sign indicate that magnitude are to added.
- The magnitudes are added with microoperation EA=A+B.

Hardware Algorithm

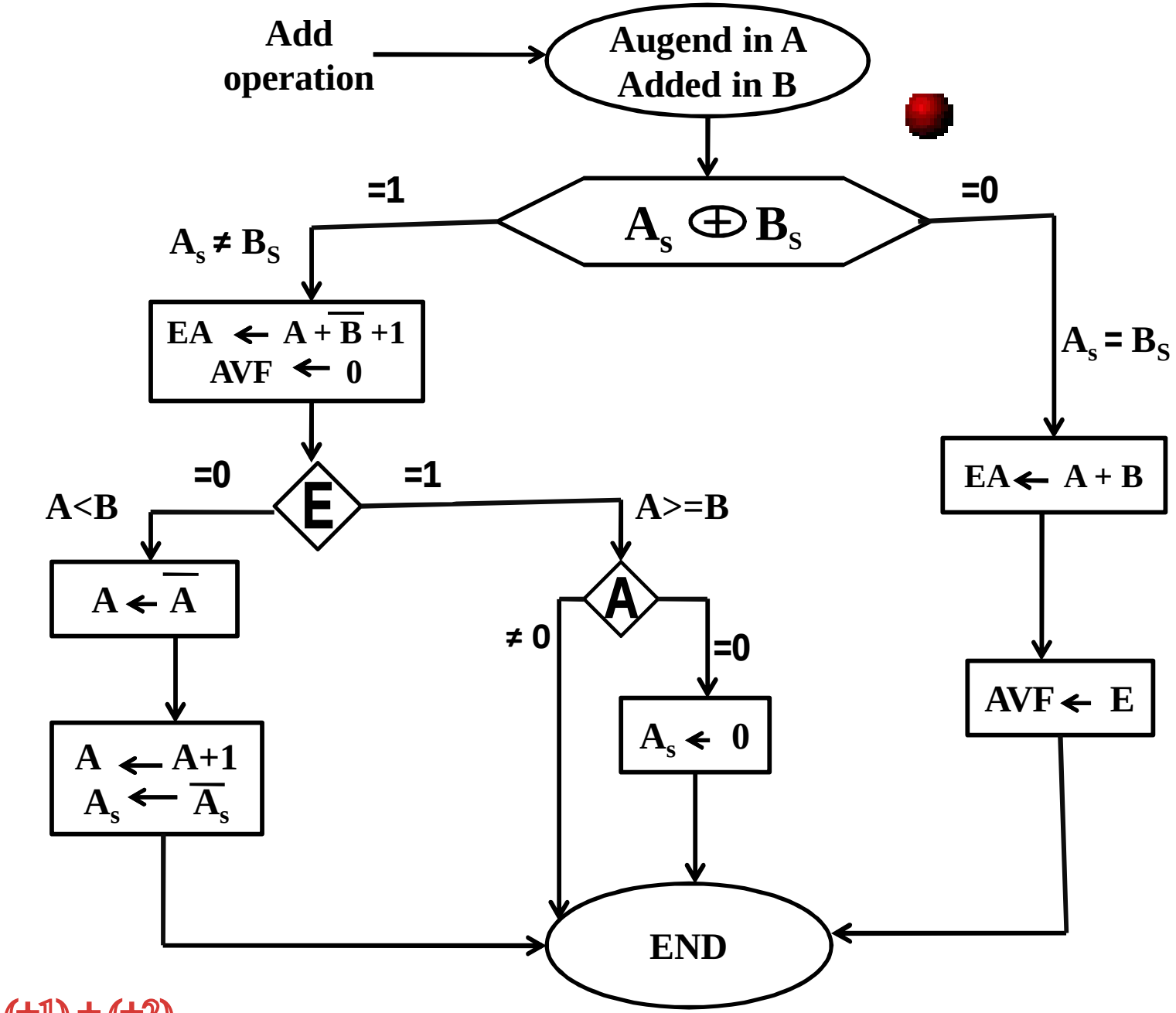
- The two magnitudes are subtracted if the sign are different for an add operation or identical for subtraction operation.
- If E=1, then the condition is A>=B and the number in A is the correct result.
- If E=0 then the condition is A<B and the number in A is taken 2's complement which is the correct result.
- If the sign of the result is same as the sign of A, So no change in As is required.
- When A<B the sign of the result is the complement of the original sign of A.
- The Final result is found in register A and its sign in As.


Add operation
Augend in A
Added in B
$A_s \oplus B_S$
=1
$A_s \neq B_S$
=0
$A_s = B_S$
EA ← A + $\overline{B}$ +1
AVF ← 0
EA ← A + B
E
=0
=1
A<B
A>=B
A ← $\overline{A}$
A
≠ 0
=0
AVF ← E
A ← A+1
A_s ← $\overline{A}_s$
A_s ← 0
END

- For Example of **Addition**

- (+1) + (+2)

  (+A) + (+B)

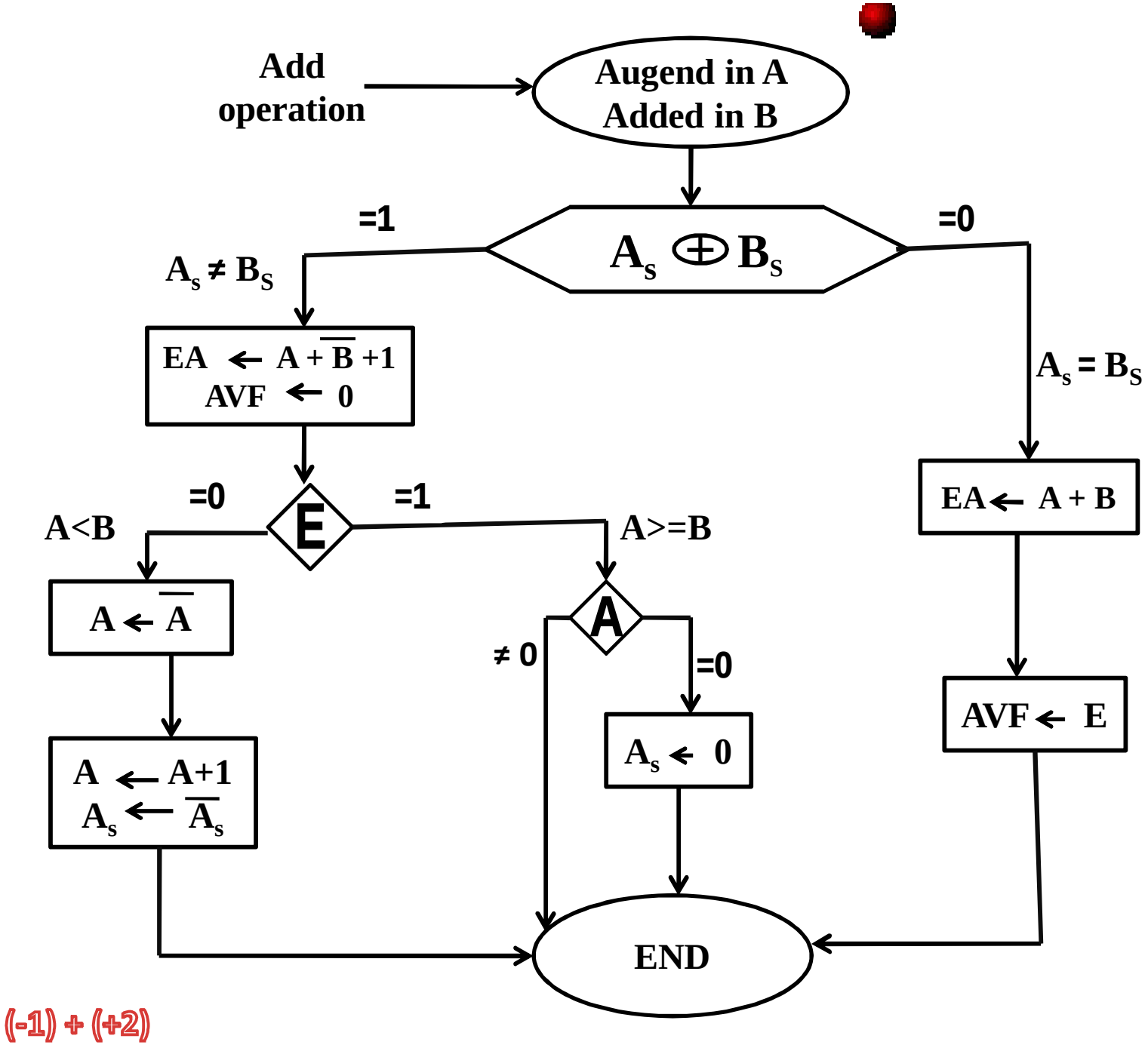

Add operation
Augend in A
Added in B
$A_s \oplus B_s$
=1
=0
$A_s \neq B_S$
$A_s = B_S$
EA ← A + $\overline{B}$ +1
AVF ← 0
EA ← A + B
E
=0
=1
A<B
A>=B
A ← $\overline{A}$
A
≠ 0
=0
A ← A+1
A_s ← $\overline{A_s}$
A_s ← 0
AVF ← E
END

(+1) + (+2)

- (-1) + (+2)

  (-A) + (+B)

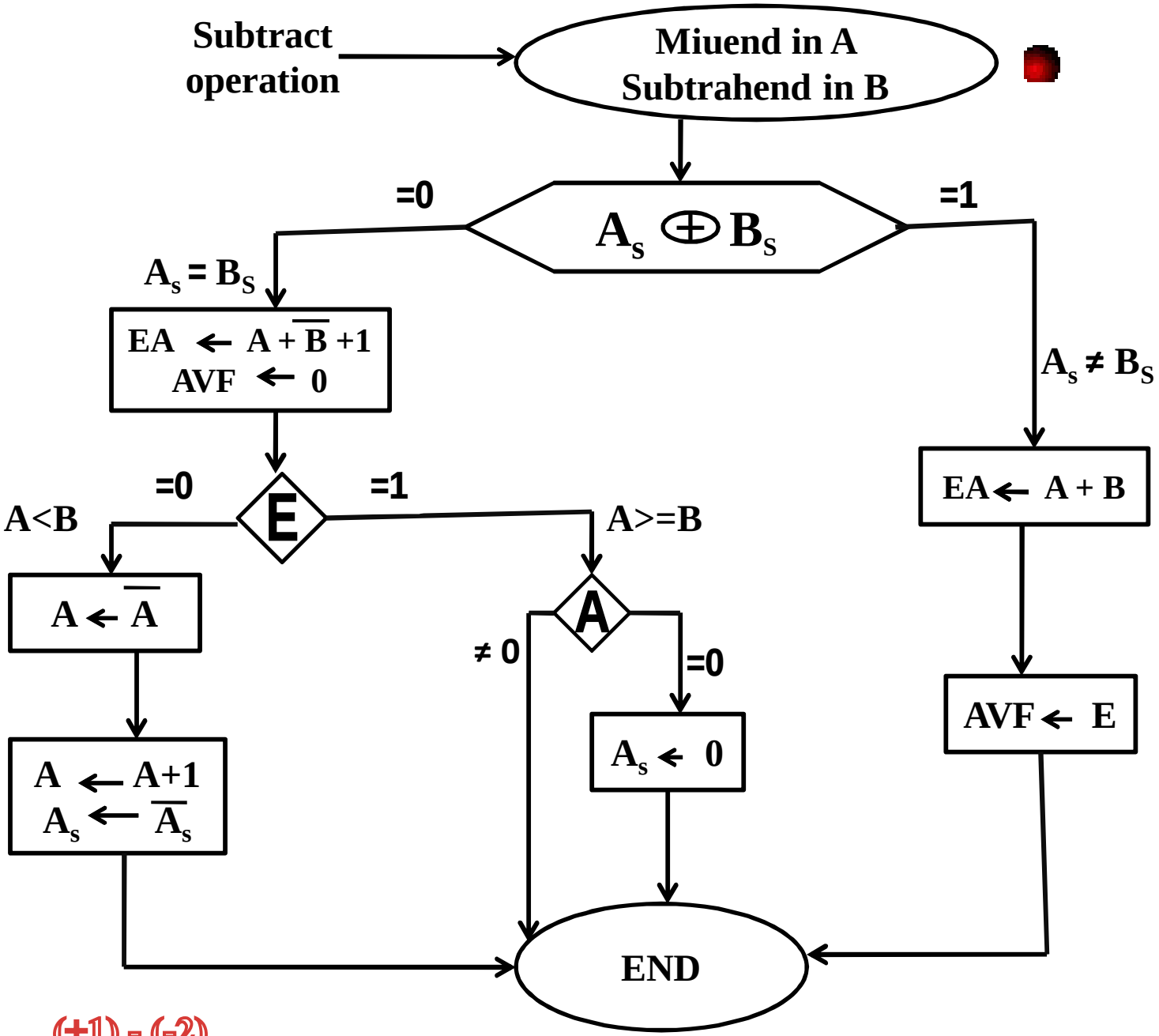
  Take A=-1, B=+2 and perform the calculation


Add operation
Augend in A
Added in B
=1
=0
$A_s \oplus B_s$
$A_s \neq B_S$
$A_s = B_S$
EA ← A + $\overline{B}$ +1
AVF ← 0
EA ← A + B
=0
E
=1
A<B
A>=B
A ← $\overline{A}$
A
≠ 0
=0
AVF ← E
A ← A+1
A_s ← $\overline{A}_s$
A_s ← 0
END
(-1) + (+2)

- For Example of **Subtraction**

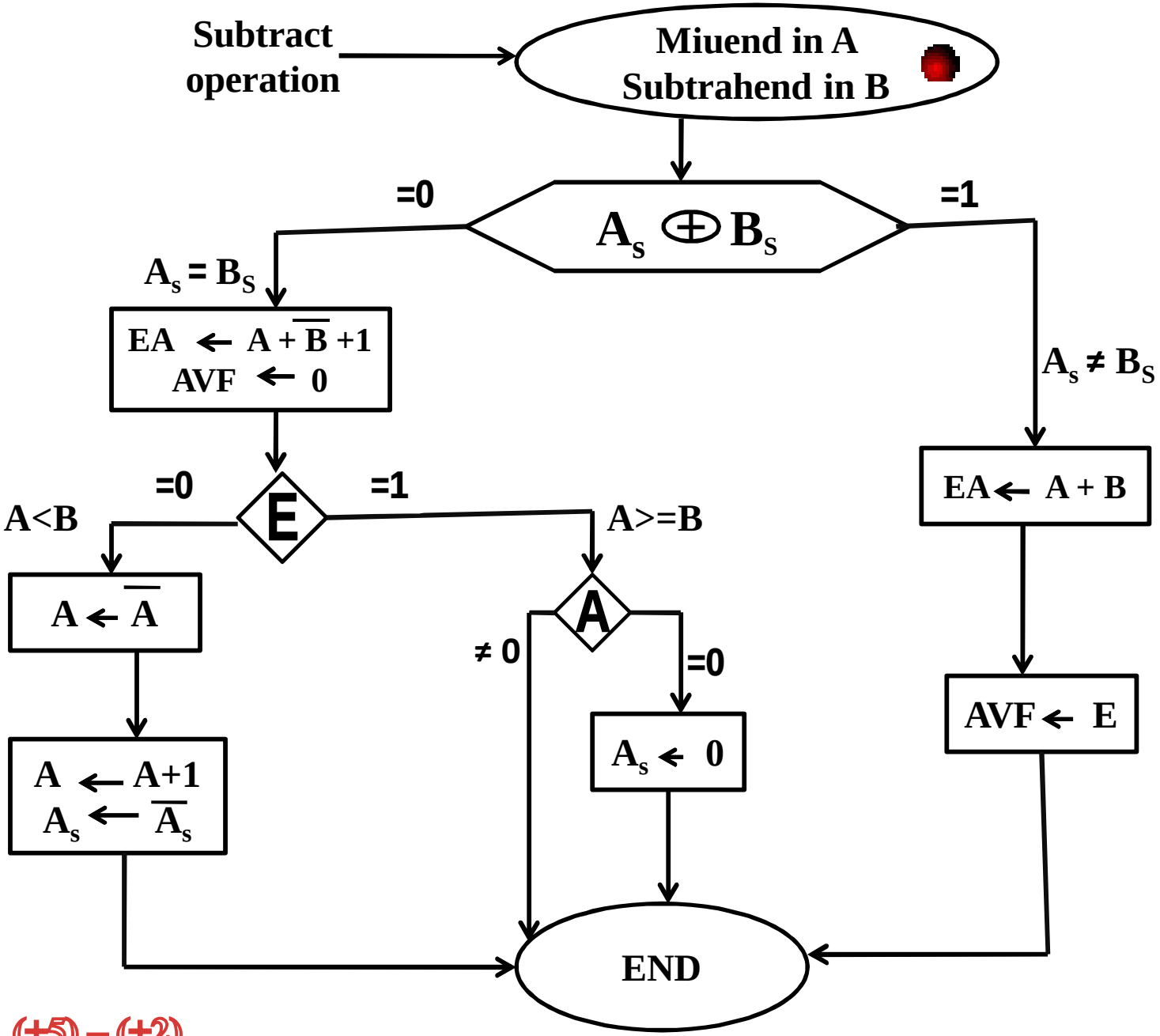
- (+1) - (-2)

  (+A) - (-B)


Subtract operation
Miuend in A
Subtrahend in B
=0
$A_s \oplus B_S$
=1
$A_s = B_S$
EA ← A + $\overline{B}$ +1
AVF ← 0
$A_s \neq B_S$
=0
E
=1
A<B
A>=B
EA ← A + B
A ← $\overline{A}$
A
≠ 0
=0
AVF ← E
A ← A+1
A_s ← $\overline{A_s}$
A_s ← 0
END
(+1) - (-2)

- $(+5) - (+2)$

  $(+A) - (+B)$


Subtract operation
Miuend in A
Subtrahend in B
$A_s \oplus B_s$
=0
=1
$A_s = B_S$
$A_s \neq B_S$
EA ← A + $\overline{B}$ +1
AVF ← 0
E
=0
=1
A<B
A>=B
EA ← A + B
A ← $\overline{A}$
A
≠ 0
=0
AVF ← E
A ← A+1
A_s ← $\overline{A}_s$
A_s ← 0
END

(+5) – (+2)