



Booth's Algorithm

CO-Tutorial Class

Presented By: Isha Agarwal

Introduction

- To multiply two integers
- Multiply binary equivalent's of decimal numbers
- We shall learn this algorithm in three parts
 - Positive * Positive
 - Positive * Negative
 - Negative * Negative

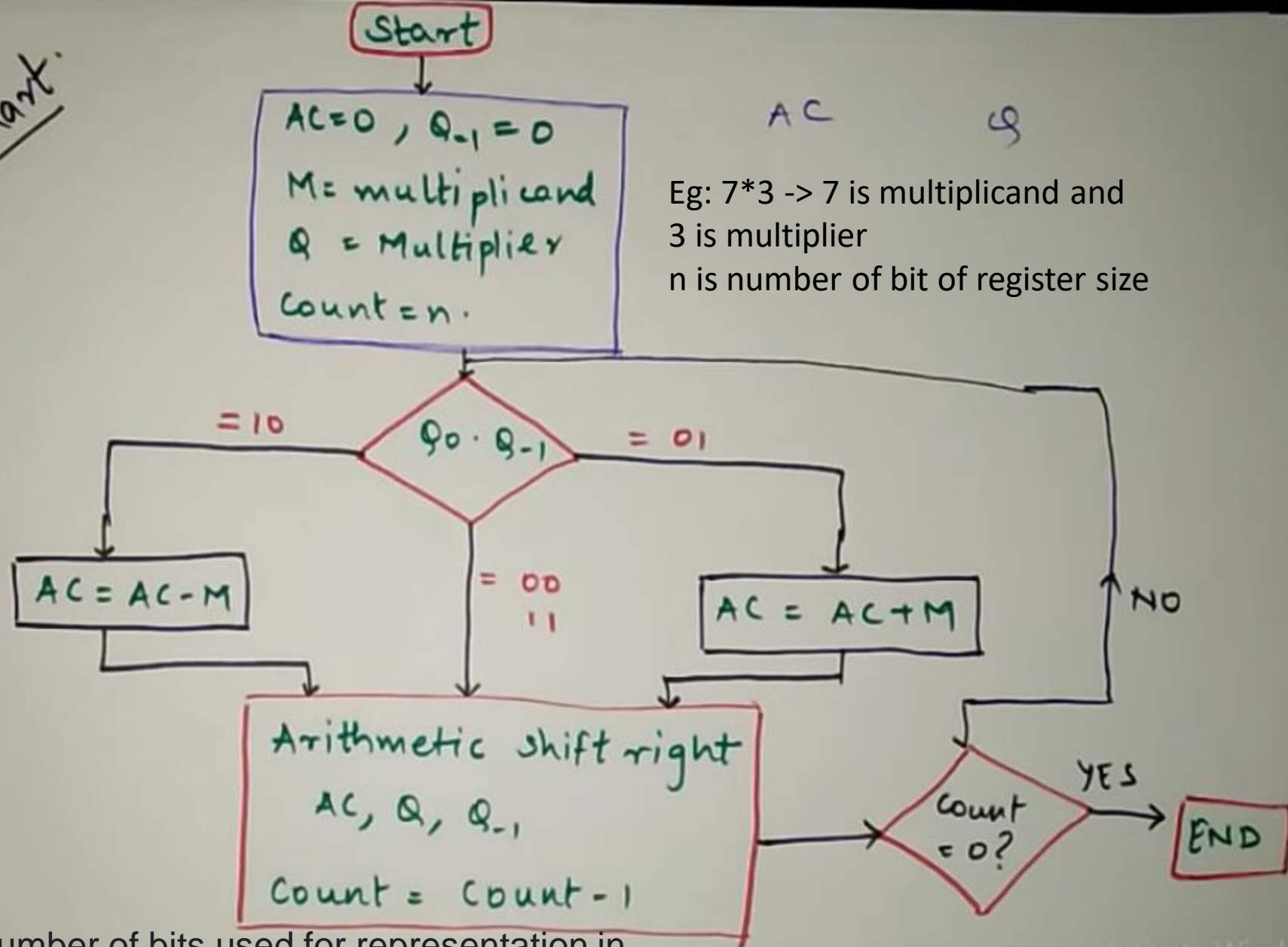
BOOTH'S ALGORITHM.

PART 01 - (+ve) \times (+ve.)

$$+(\text{Number 1}) \times +(\text{Number 2})$$

$$= +(\text{Product}).$$

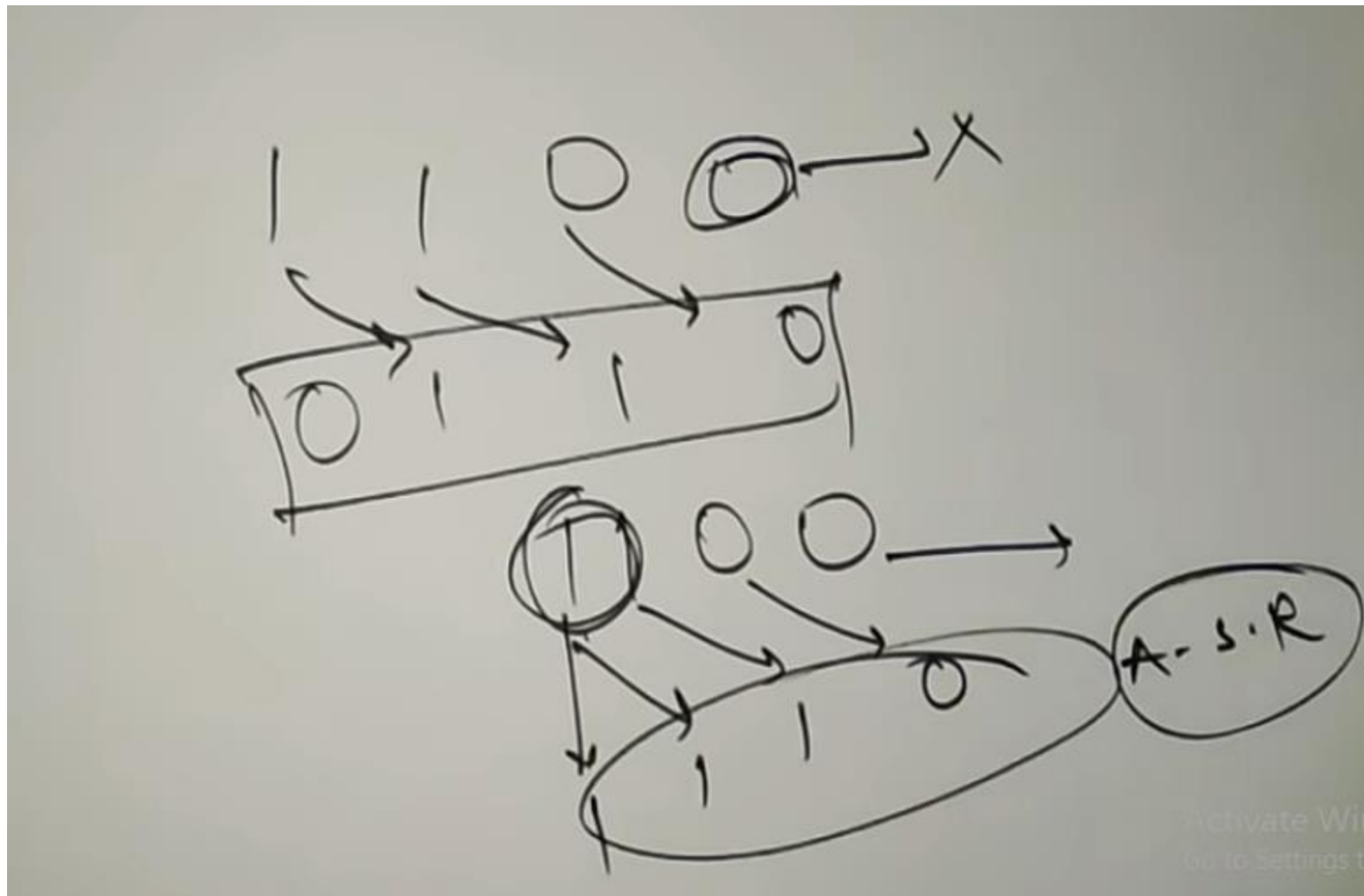
Booth's # Flowchart.



$n \geq$ the number of bits used for representation in binary (both positive numbers)

$n + 1 \geq$ the number of bits used for representation in binary (any one number negative)

Difference between SR and ASR



Example

- Multiply 7 and 3 using Booth's algorithm.

Register size = 4

$$(M) \rightarrow (7)_{10} \rightarrow (0111)_2$$

$$(Q) \rightarrow (3)_{10} \rightarrow (0011)_2$$

$$(-M) \rightarrow 2's \text{ comp } (0111)_2 \rightarrow (1001)_2$$

$$\begin{aligned} AC + (-M) \\ AC = AC \oplus (-M) \\ AC = AC + M \end{aligned}$$

$$\begin{array}{r} 0111 \\ 1's \rightarrow 1000 \\ 2's \rightarrow + 1 \\ \hline (1001)_2 \end{array}$$

AC (4)
0000

Q
0011

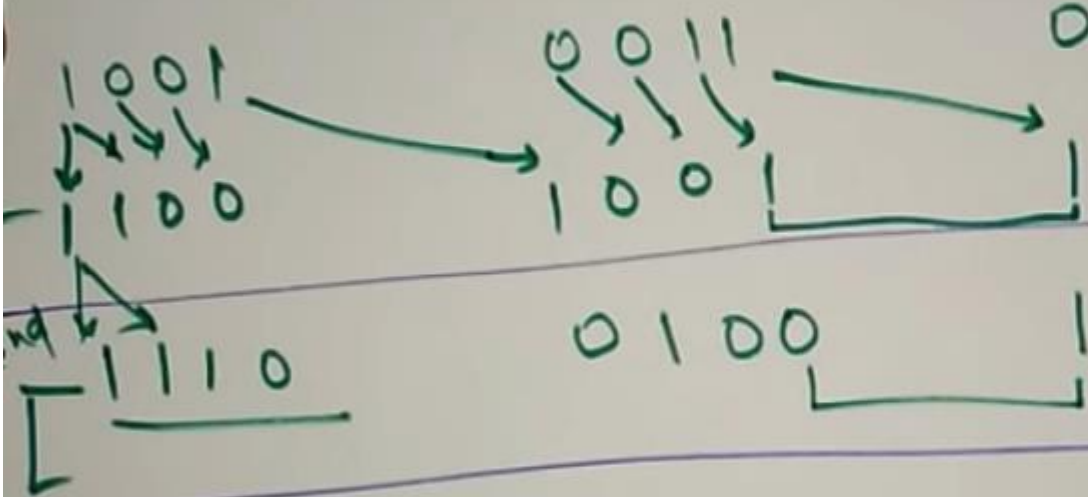
Q-1
0

Operation.

i) $AC = AC - M$

$$\begin{array}{r} 0000 \\ + 1001 \\ \hline (1001) \end{array}$$

ii) A.S.R.



i) A.S.R.

$$Q_0 \cdot Q_{-1} = 11$$

3rd
0101
0010

0100
1010

1
0

i) $Q_0 \cdot Q_{-1} = 01$

$AC = AC + M$

$$\begin{array}{r} 1110 \\ + 0111 \\ \hline 0101 \end{array}$$

ii) A.S.R.

(D) Second operation

3rd
0101
0010

0100
1010

$$i) Q_0 \cdot Q_{-1} = 01$$

$$AC = AC + M$$

1110

+ 0111

0101

Operation.

$$Q_0 \cdot Q_{-1} = 00$$

AC

4th

0001

0101

$$(00010101)_2 = (21)_{10}$$

$$7 \times 3 = (21)$$

$$(7)_{10} \times (3)_{10} = (21)_{10}$$

BOOTH'S ALGORITHM

PART 02 - (+ve) \times (-ve) / (-ve) \times (+ve)

+ (Number 1) \times - (Number 2)

= - (Product).

- Multiply -7 and $+3$ using Booth's algorithm.
register bits = 5



$$\begin{array}{r}
 \overline{)00111} \\
 11000 \\
 \overline{+} \quad \quad 1 \\
 \hline
 11001
 \end{array}$$

$$M \rightarrow (-7)_{10} \rightarrow (11001)_2$$

$$Q \rightarrow (3)_{10} \rightarrow (00011)_2$$

$$(-M) \rightarrow (7)_{10} \rightarrow (00111)_2$$

1st AC

00000

00111
00111

2nd 00001

3rd

11010
11101

5

Q₉ Q₀
00011

00011

10001

11000

11000

01100

Q-1

0

0

1

1

1

0

Operation.

i) $AC = AC - M$

00000
+ 00111

00111

ii) A.S.R

i) A.S.R.


i) $AC = AC + M$

00001
+ 11001

11010

ii) A.S.R.

①



i) A.S.R.

1

$$i) AC = AC + M$$

00001
11001

Operation

i) A.S.R

~~A~~ C

9

9-1

10110

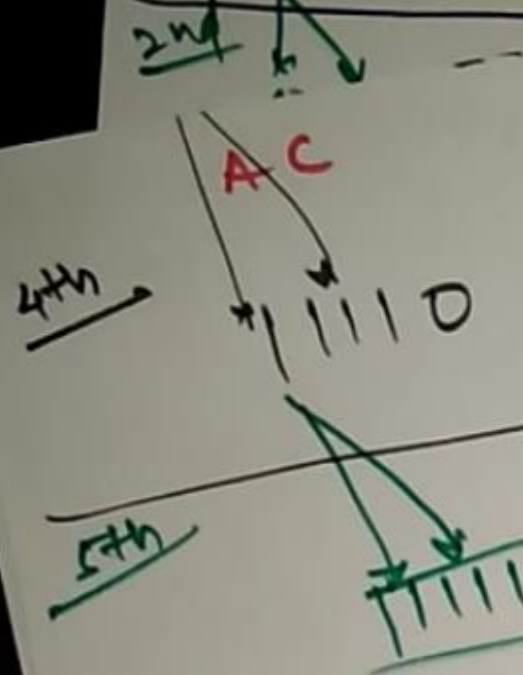
①

i) A.S.R.

01011

(100%)

$$(111101011)_2$$



10110

0

i) A.S.R.

01011

$$-7 \times 3 = (-)$$

$$(1001)_2$$

$(1111101011)_2$

1's \rightarrow 0000010100

0000010101

2's

$$-7 \times 3 = (-21)_{10}$$

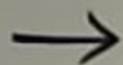
BOOTH'S ALGORITHM

PART 03 - (-ve) \times (-ve)

$$-(\text{Number 1}) \times -(\text{Number 2})$$

$$= + (\text{Product}).$$

- Multiply -7 and -3 using Booth's Algo.
Register size = 4 bits.



- Multiply -7 and -3 using Booth's Algo.
Register size = 4 bits.

→

$$M \rightarrow (-7)_{10} \rightarrow (1001)_2$$

$$-M \rightarrow (7)_{10} \rightarrow (0111)_2$$

$$Q \rightarrow (-3)_{10} \rightarrow (1101)_2$$

$$\begin{array}{r} \boxed{0111} \\ 1's \text{ } 1000 \\ + \\ 2's \text{ } (1001)_2 \end{array}$$

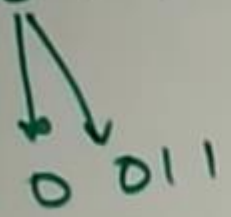
$$\begin{array}{r} 0011 \\ 1's \text{ } 1100 \\ + \\ 2's \text{ } (1101)_2 \end{array}$$

1st

A C

0 0 0 0

0 1 1 1



1 1 0 1

0

1 1 0 1

0

1 1 1 0

1

i) $AC = AC - M$

$$\begin{array}{r} 0000 \\ + 0111 \\ \hline 0111 \end{array}$$

ii) A.S.R.

2nd

1 1 0 0



1 1 1 0

1

0 1 1 1

0

i) $AC = AC + M$

$$\begin{array}{r} 0011 \\ + 1001 \\ \hline 1100 \end{array}$$

ii) A.S.R.

AC	Q	Q-1	Operation.
<u>3rd</u> 0101	0111	0	i) $AC \leftarrow AC - M$ $\begin{array}{r} 1110 \\ + 0111 \\ \hline 0101 \end{array}$ (Discard carry)
0010	1011	1	ii) A.S.R

<u>4th</u> 0001	0101	1	i) A.S.R
--------------------	------	---	----------

$(00010101)_2$