Computer Architecture

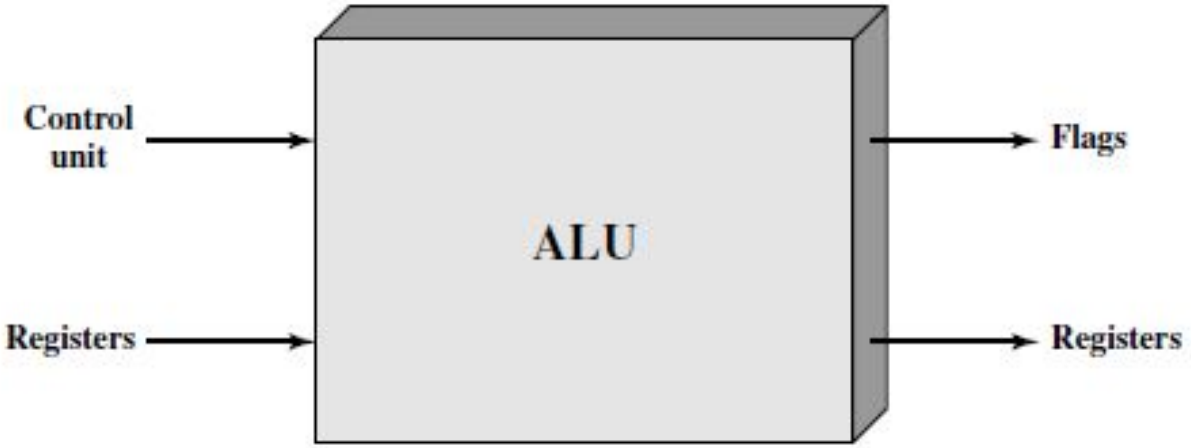
Computer Arithmetic

Instructor: Md. Biplob Hosen
Lecturer, IIT, JU

Reference Books

- Computer Organization and Architecture: Designing for Performance- William Stallings (8^{th} Edition)
  - Any later edition is fine

ALU



- Data are presented to the ALU in registers, and the results of an operation are stored in registers
- The ALU may also set flags as the result of an operation
- The flag values are also stored in registers within the processor
- The control unit provides signals that control the operation of the ALU and the movement of the data into and out of the ALU

Number Representation (Fixed Point)

- Integer Representation
  - Straight forward, can not present signed numbers
- Sign-Magnitude Representation
  - If the sign bit is 0, the number is positive; if the sign bit is 1, the number is negative
- Disadvantages:
  1) Addition and subtraction require a consideration of both the signs of the numbers and their relative magnitudes to carry out the required operation
  2) There are two representations of 0

Number Representation (Fixed Point)

- Twos Complement Representation
  - Mostly used

Range	-2^{n-1} through $2^{n-1} - 1$
Number of Representations of Zero	One
Negation	Take the Boolean complement of each bit of the corresponding positive number, then add 1 to the resulting bit pattern viewed as an unsigned integer.
Expansion of Bit Length	Add additional bit positions to the left and fill in with the value of the original sign bit.
Overflow Rule	If two numbers with the same sign (both positive or both negative) are added, then overflow occurs if and only if the result has the opposite sign.
Subtraction Rule	To subtract B from A, take the twos complement of B and add it to A.

Converting between Different Bit Lengths

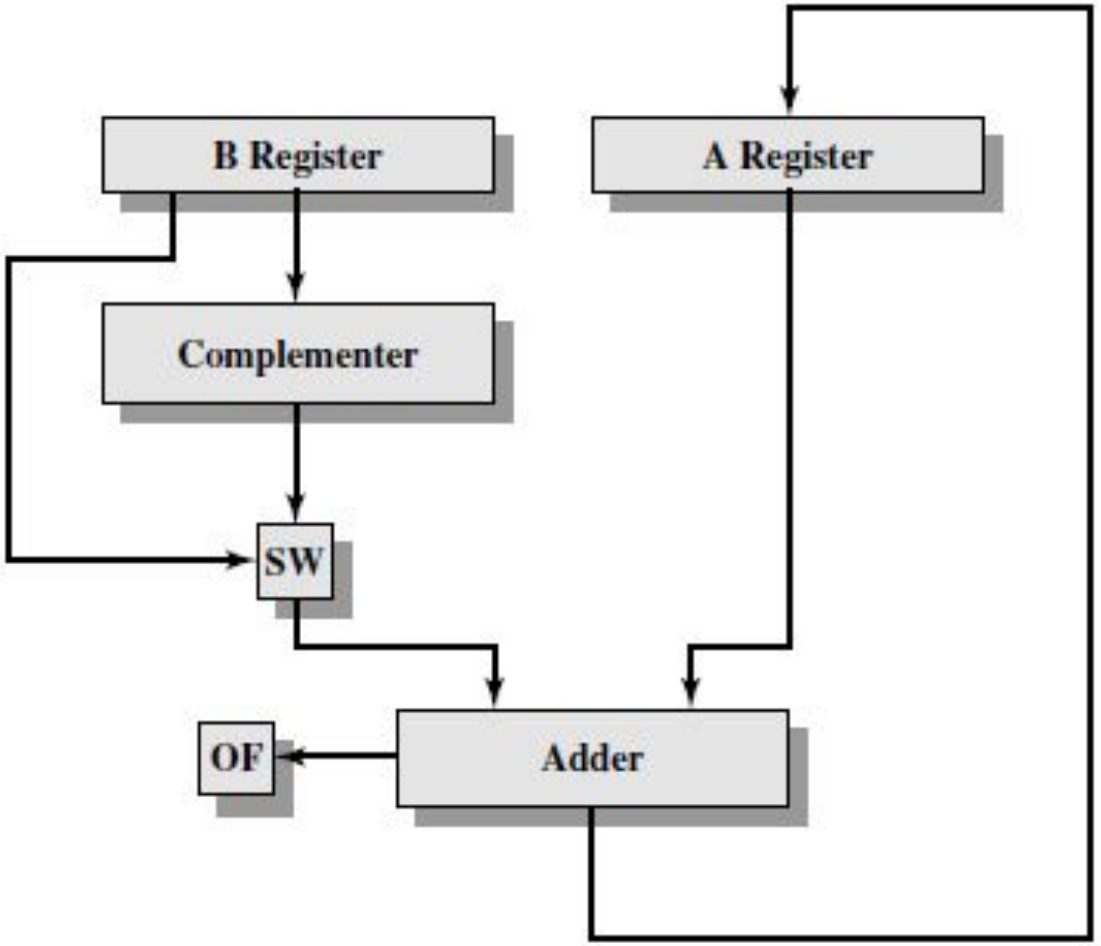
+18	=	00010010	(sign magnitude, 8 bits)
+18	=	0000000000010010	(sign magnitude, 16 bits)
−18	=	10010010	(sign magnitude, 8 bits)
−18	=	1000000000010010	(sign magnitude, 16 bits)

+18	=	00010010	(twos complement, 8 bits)
+18	=	0000000000010010	(twos complement, 16 bits)
−18	=	11101110	(twos complement, 8 bits)
−32,658	=	1000000001101110	(twos complement, 16 bits)

−18	=	11101110	(twos complement, 8 bits)
−18	=	1111111111101110	(twos complement, 16 bits)

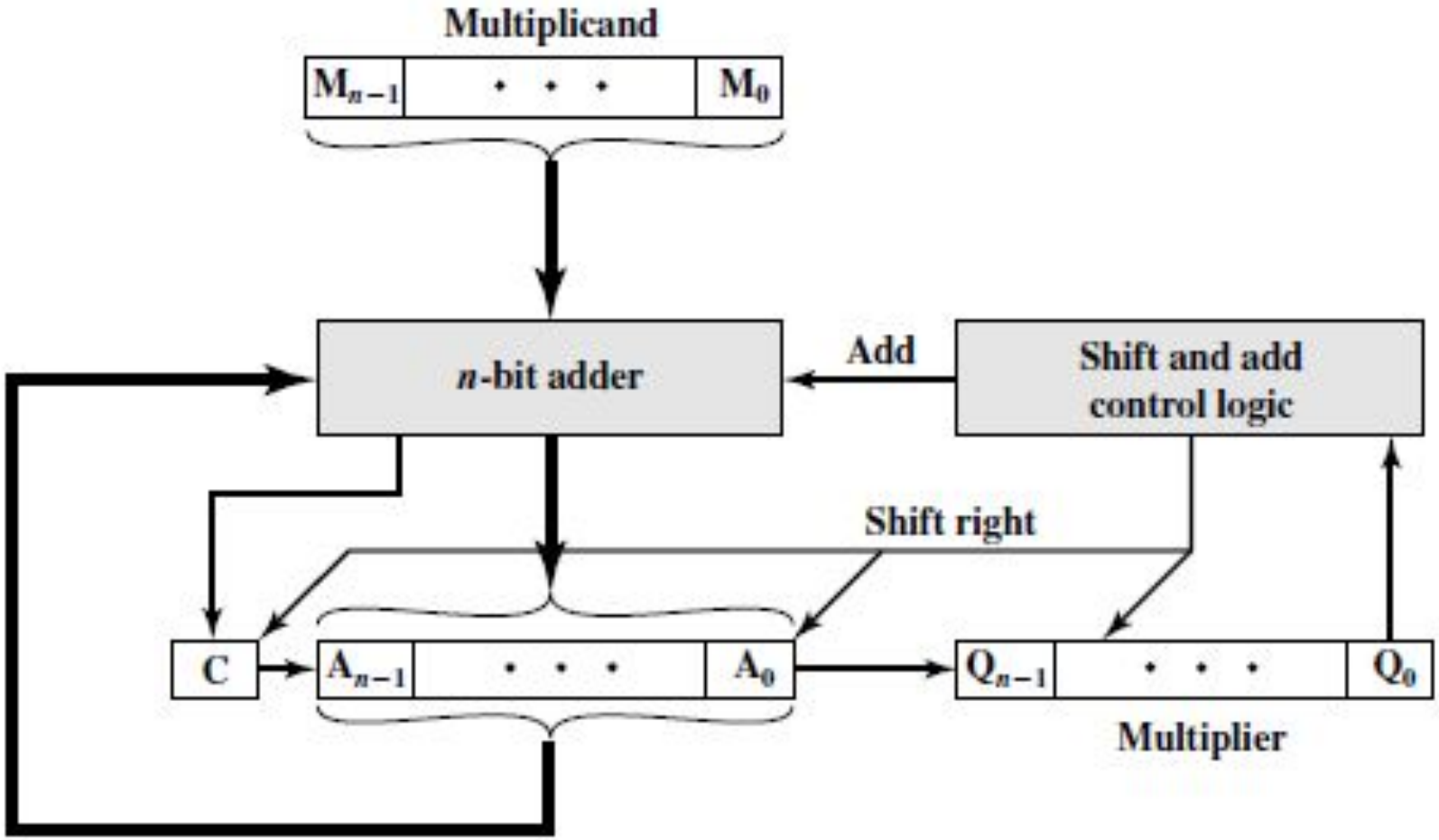
- This is called sign extension

Block Diagram of Hardware for Addition and Subtraction



OF = Overflow bit
SW = Switch (select addition or subtraction)

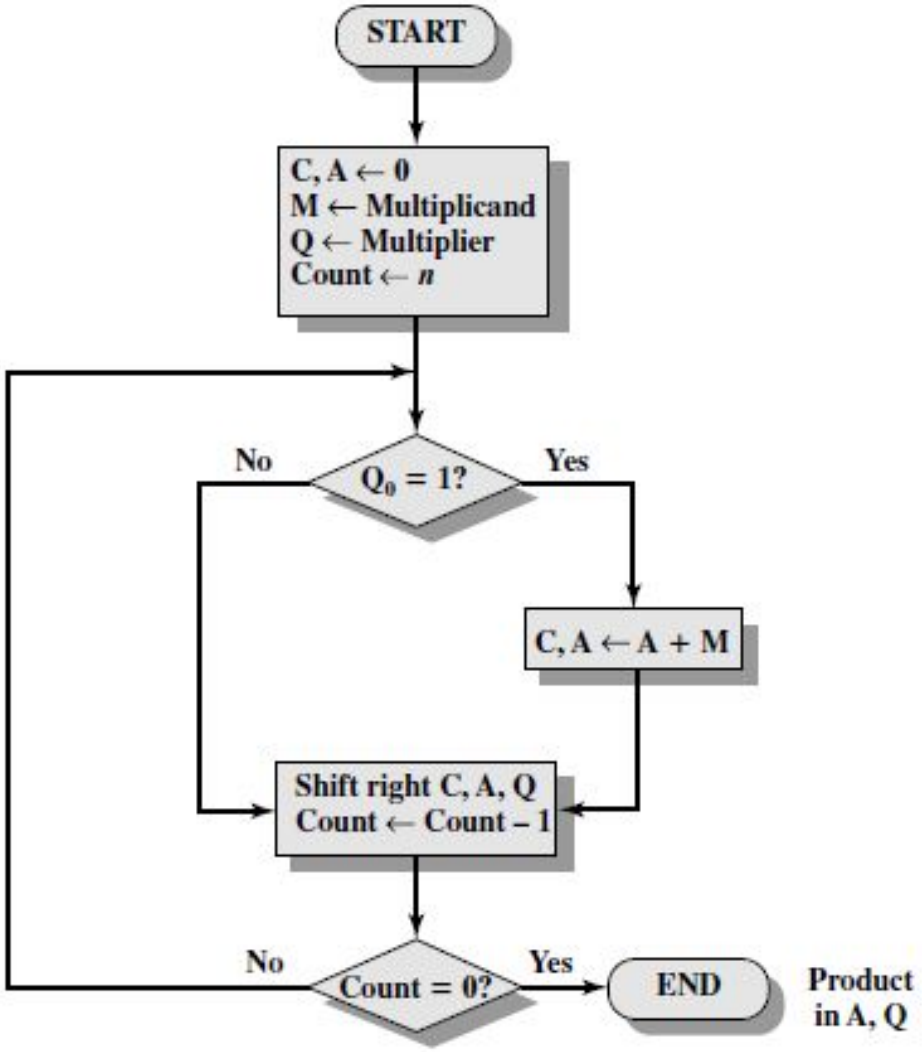
Multiplication Implementation



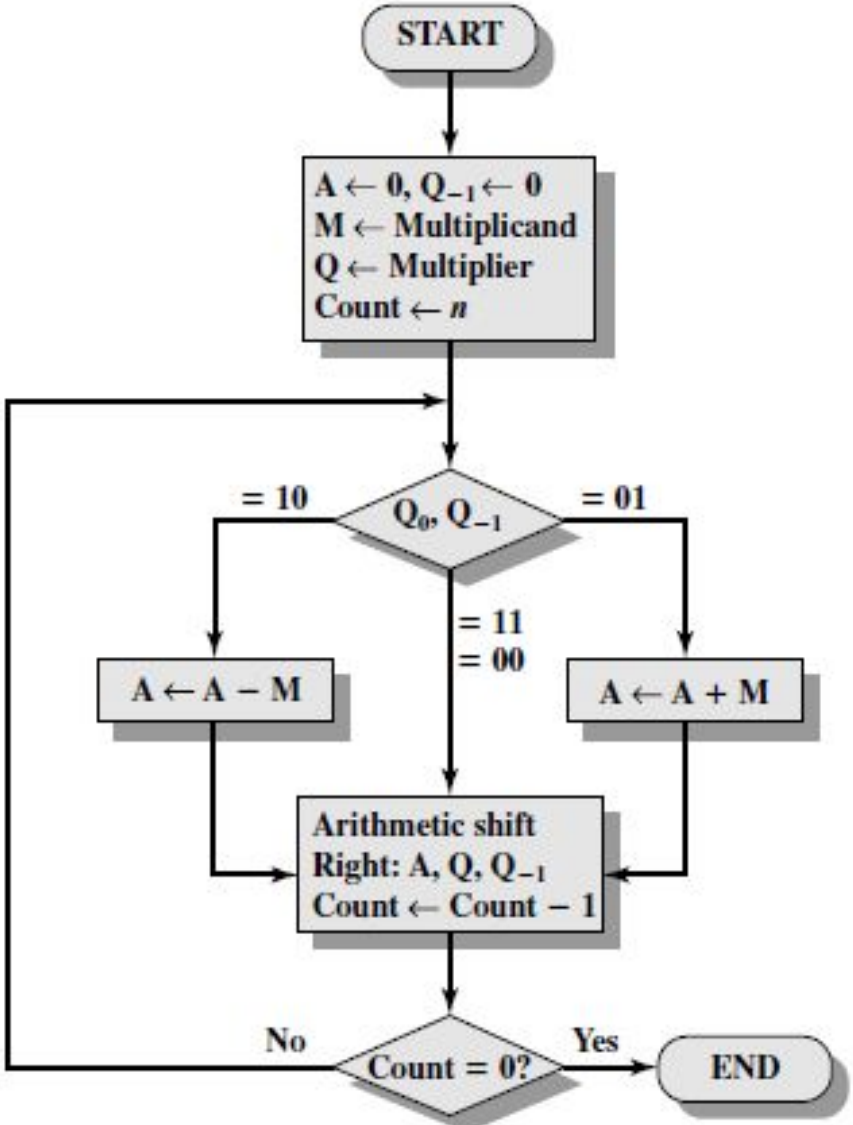
Multiplication Implementation

C	A	Q	M		
0	0000	1101	1011	Initial values	
0	1011	1101	1011	Add	First cycle
0	0101	1110	1011	Shift	
0	0010	1111	1011	Shift	Second cycle
0	1101	1111	1011	Add	Third cycle
0	0110	1111	1011	Shift	
1	0001	1111	1011	Add	Fourth cycle
0	1000	1111	1011	Shift	

Multiplication (Unsigned) Implementation



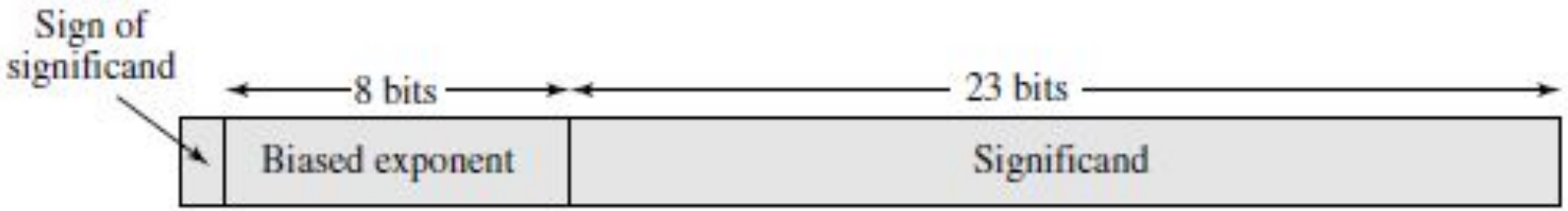
Multiplication (2's Complement) Implementation



Multiplication (2's Complement) Implementation

A	Q	Q_{-1}	M		
0000	0011	0	0111	Initial values	
1001	0011	0	0111	$A \leftarrow A - M$	First cycle
1100	1001	1	0111	Shift	
1110	0100	1	0111	Shift	Second cycle
0101	0100	1	0111	$A \leftarrow A + M$	Third cycle
0010	1010	0	0111	Shift	
0001	0101	0	0111	Shift	Fourth cycle

Floating-Point Representation Format



(a) Format

$1.1010001 \times 2^{10100} = 0\ 10010011\ 10100010000000000000000 = 1.6328125 \times 2^{20}$

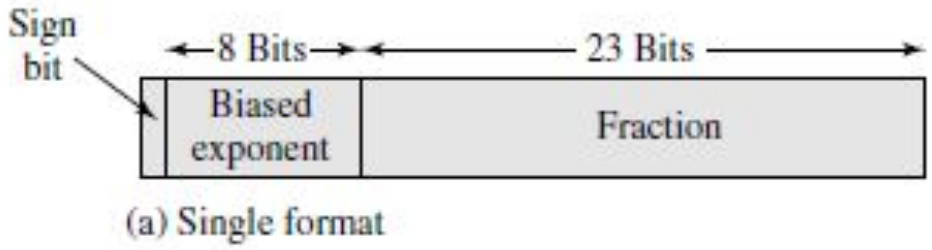
$-1.1010001 \times 2^{10100} = 1\ 10010011\ 10100010000000000000000 = -1.6328125 \times 2^{20}$

$1.1010001 \times 2^{-10100} = 0\ 01101011\ 10100010000000000000000 = 1.6328125 \times 2^{-20}$

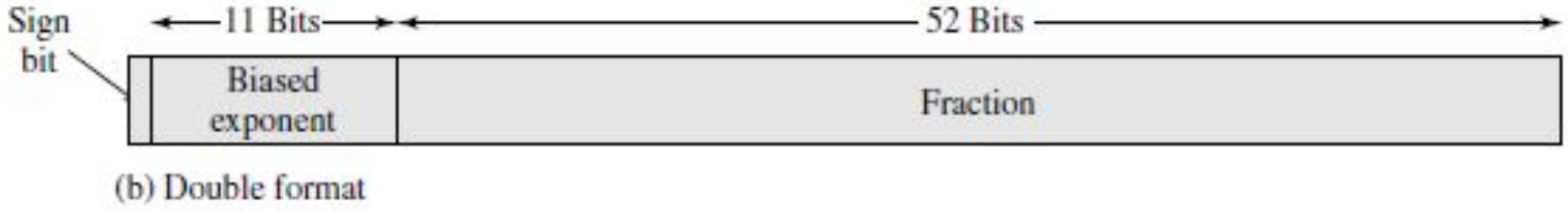
$-1.1010001 \times 2^{-10100} = 1\ 01101011\ 10100010000000000000000 = -1.6328125 \times 2^{-20}$

(b) Examples

IEEE Standard for Binary Floating-Point Representation



(a) Single format



(b) Double format