COMPUTER ORGANIZATION AND ARCHITECTURE
Course Code: CAP403R01 Semester : I / MCA
Dr. K. Geetha/SAP/CSE/SOC/SASTRA

1.5(b) Arithmetic for Computers: Division

Figure 10.15 shows an example of the long division of unsigned binary integers.
First, the bits of the dividend are examined from left to right,
until the set of bits examined represents a number greater than or equal to the divisor; this is referred to as the divisor being able to divide the number.
Until this event occurs, 0s are placed in the quotient from left to right.
When the event occurs, a 1 is placed in the quotient and the divisor is subtracted from the partial dividend. The result is referred to as a partial remainder.

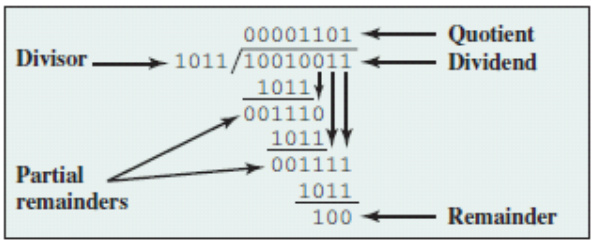

Figure 10.15 Example of Division of Unsigned Binary Integers

Figure 10.16 shows a machine algorithm that corresponds to the long division process. The divisor is placed in the M register, the dividend in the Q register. At each step, the A and Q registers together are shifted to the left 1 bit. M is subtracted from A to determine whether A divides the partial remainder.3 If it does, then Q0 gets a 1 bit. Otherwise, Q0 gets a 0 bit and M must be added back to A to restore the previous value. The count is then decremented, and the process continues for n steps. At the end, the quotient is in the Q register and the remainder is in the A register. This process can, with some difficulty, be extended to negative numbers. We give here one approach for twos complement numbers. An example of this approach is shown in Figure 10.17.

A	Q	
0000	0111	**Initial value**
0000	1110	**Shift**
1101		**Use twos complement of 0011 for subtraction**
1101		**Subtract**
0000	1110	**Restore, set $Q_0 = 0$**
0001	1100	**Shift**
1101		
1110		**Subtract**
0001	1100	**Restore, set $Q_0 = 0$**
0011	1000	**Shift**
1101		
0000	1001	**Subtract, set $Q_0 = 1$**
0001	0010	**Shift**
1101		
1110		**Subtract**
0001	0010	**Restore, set $Q_0 = 0$**

Figure 10.17 Example of Restoring Twos Complement Division (7/3)

The algorithm assumes that the divisor V and the dividend D are positive and that $|V| < |D|$. If $|V| = |D|$, then the quotient $Q = 1$ and the remainder $R = 0$. If $|V| > |D|$, then $Q = 0$ and $R = D$. The algorithm can be summarized as follows:

1. Load the twos complement of the divisor into the M register; that is, the M register contains the negative of the divisor. Load the dividend into the A, Q registers. The dividend must be expressed as a $2n$-bit positive number. Thus, for example, the 4-bit 0111 becomes 00000111.
2. Shift A, Q left 1 bit position.
3. Perform $A \leftarrow A - M$. This operation subtracts the divisor from the contents of A.
4. a. If the result is nonnegative (most significant bit of $A = 0$), then set $Q_0 \leftarrow 1$.
   b. If the result is negative (most significant bit of $A = 1$), then set $Q_0 \leftarrow 0$. and restore the previous value of A.
5. Repeat steps 2 through 4 as many times as there are bit positions in Q.
6. The remainder is in A and the quotient is in Q.

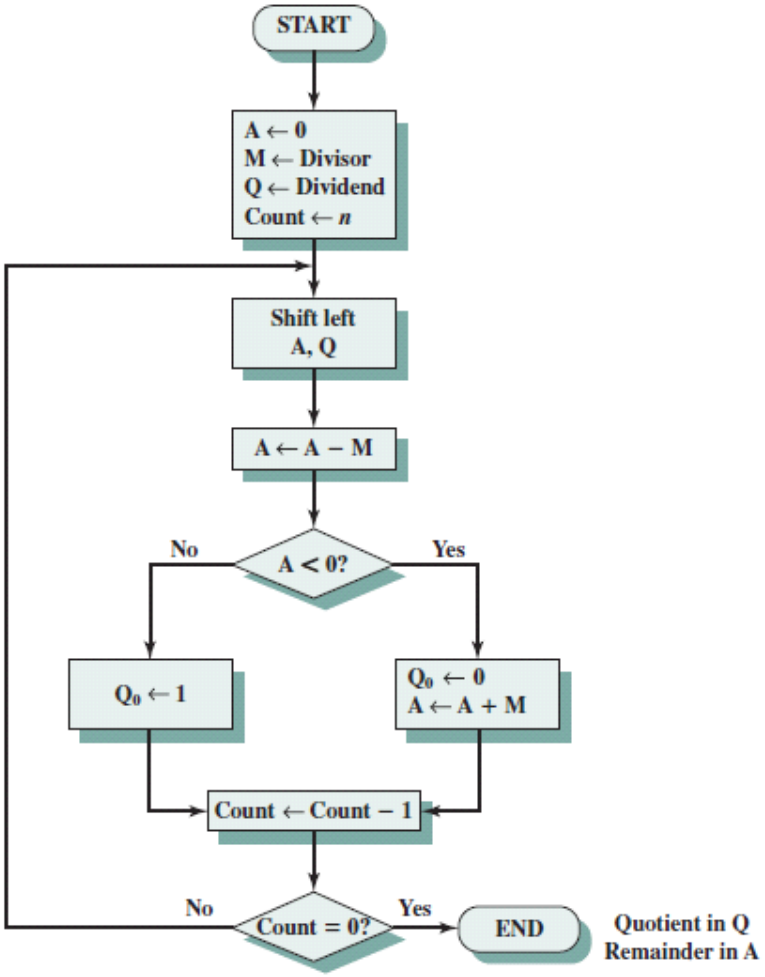

Figure 10.16 Flowchart for Unsigned Binary Division

To deal with negative numbers, we recognize that the remainder is defined by

$$D = Q \times V + R$$

That is, the remainder is the value of R needed for the preceding equation to be valid. Consider the following examples of integer division with all possible combinations of signs of D and V:

$$\begin{array}{llllll} D = 7 & V = 3 & \Rightarrow & Q = 2 & R = 1 \\ D = 7 & V = -3 & \Rightarrow & Q = -2 & R = 1 \\ D = -7 & V = 3 & \Rightarrow & Q = -2 & R = -1 \\ D = -7 & V = -3 & \Rightarrow & Q = 2 & R = -1 \end{array}$$

The reader will note from Figure 10.17 that $(-7)/(3)$ and $(7)/(-3)$ produce different remainders. We see that the magnitudes of Q and R are unaffected by the input signs and that the signs of Q and R are easily derivable from the signs of D and V. Specifically, $\text{sign}(R) = \text{sign}(D)$ and $\text{sign}(Q) = \text{sign}(D) \times \text{sign}(V)$. Hence, one way to do twos complement division is to convert the operands into unsigned values and, at the end, to account for the signs by complementation where needed. This is the method of choice for the restoring division algorithm [PARH10].