COMPUTER ORGANIZATION AND ARCHITECTURE

UNIT –III

TOPIC- ADDITION & SUBTRACTION OF SIGNED 2'S COMPLEMENT DATA

• When an integer is positive, the MSB, or sign bit, is 0 and the remaining bits represent the magnitude.
• When an integer is negative, the MSB, or sign bit, is 1, but the rest of the number can be represented in one of three ways:

- Signed-magnitude representation
- Signed-1's complement representation
- Signed-2's complement representation

Addition and subtraction of Signed 2's complement data

Addition using 2's complement data

- The sign bit is represented by the leftmost bit of a binary number: 0 for positive and 1 for negative.
- The full number is displayed in 2's complement form if the sign bit is 1.
- As a result, +33 is represented as 00100001, and -33 is represented as 11011111.
- The 2's complement of 00100001 is 11011111.
- When two numbers are added in signed -2's complement form, the sign bits of the numbers are regarded the same as the other bits of the number.
- The sign-bit position's carry-out is disregarded.

Ex1: Use the 2's complement approach to add two decimal integers of +7 and +4.

Solution: The 2's complement representations of +4 and +7 with 5 bits each are shown below.

$+7_{10} = 00111_2$
$+4_{10} = 00100_2$
The addition of these two numbers is
$(+7_{10}) + (+4_{10}) = 00111_2 + 00100_2 = 01011_2$

- The resultant sum is 5 bits long.
- As a result, there is no carryover from the sign bit.
- The final total is positive, as shown by the sign bit '0'.
- In the decimal number system, the magnitude of the sum is 11.
- Therefore, the addition of two positive numbers will give another positive number.

Ex 2:use the 2's complement approach to subtract two decimal integers of +7 and +4.

Solution: The subtraction of these two numbers is

$(+4_{10}) - (+7_{10}) = (+4_{10}) + (-7_{10})$
The 2's complement representations of +4 and -7 with 5 bits each are shown below:

$+4_{10} = 00100_2$
$-7_{10} = 11001_2$
$(+4_{10}) + (-7_{10}) = 00100_2 + 11001_2 = 11101_2$

- Here, carry is not obtained from the sign bit.
- The final total is negative, as shown by the sign bit '1'.
- As a result, we may determine the magnitude of the resultant sum as 3 in the decimal number system by taking 2's complement of it.
- Therefore, the subtraction of two decimal numbers, +4 and +7, is -3.

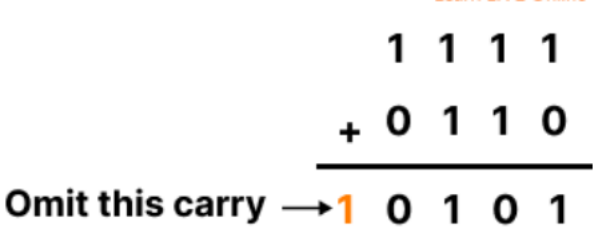
Subtraction Using 2's Complement data

The following steps are to be followed:

- **Step 1:** Determine the 2's complement of the smaller number
- **Step 2:** Add this to the larger number.
- **Step 3:** Omit the carry. Note that there is always a carry in this case.

Example: Subtract $(1010)_2$ from $(1111)_2$ using 2's complement method.

- **Step 1:** 2's complement of $(1010)_2$ is $(0110)_2$.
- **Step 2:** Add $(0110)_2$ to $(1111)_2$.

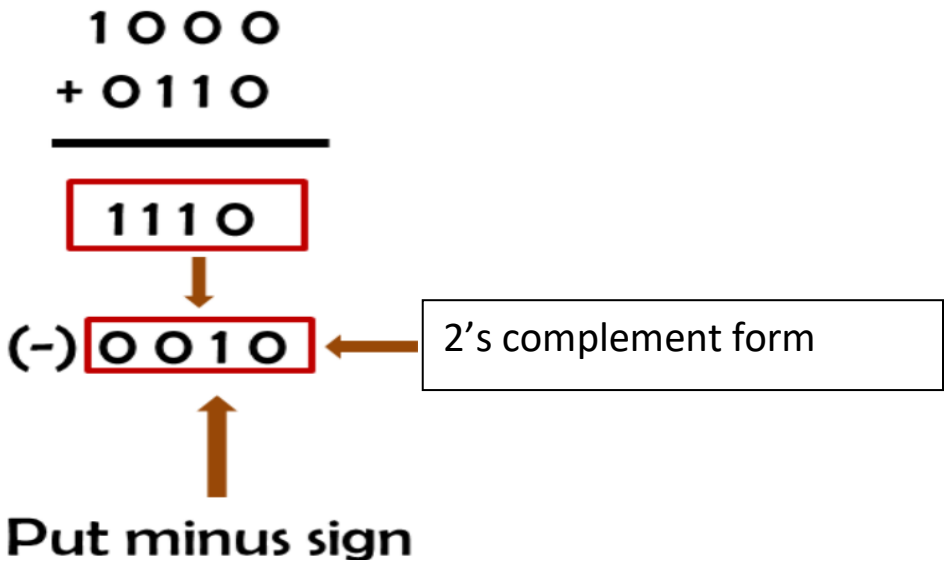


To subtract a larger number from a smaller number using 2's complement subtraction, the following steps are to be followed:

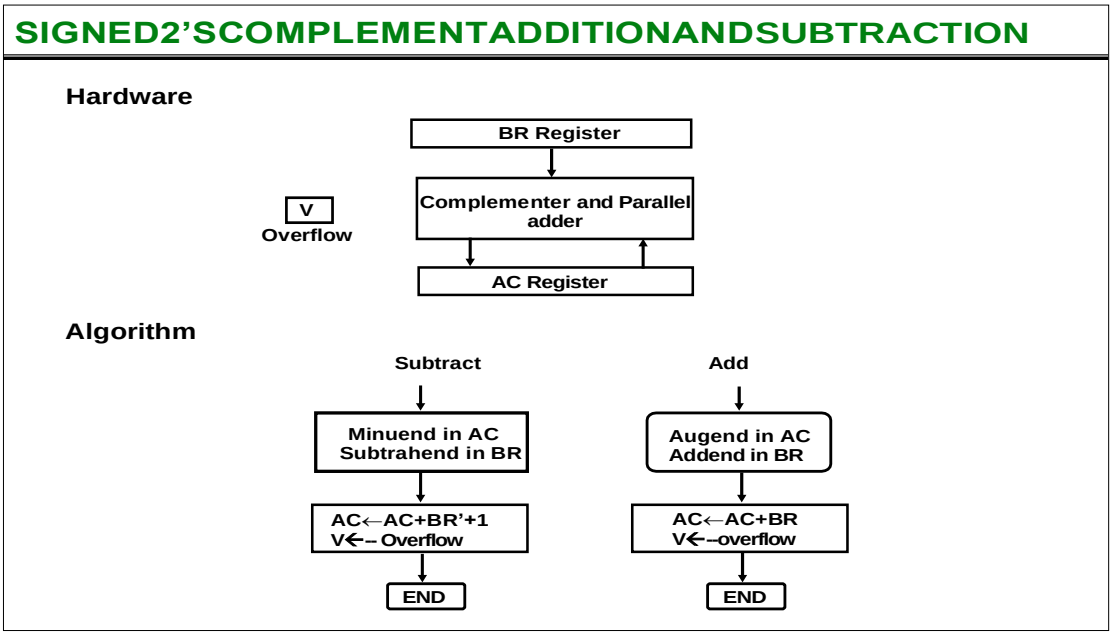
- **Step 1:** Determine the 2's complement of the smaller number.
- **Step 2:** Add this to the larger number.
- **Step 3:** There is no carry in this case. The result is in 2's complement form and is negative.
- **Step 4:** To get an answer in true form, take 2's complement and change its sign.

Example: Subtract $(1010)_2$ from $(1000)_2$ using 2's complement.

- **Step 1:** Find the 2's complement of $(1010)_2 = (0110)_2$.
- **Step 2:** Add $(0110)_2$ to $(1000)_2$.



Hardware Implementation



- This is same as signed magnitude addition & subtraction except that the sign bits are not separated from the rest of the registers.
- We name the A register AC (accumulator) and the B register BR.
- The leftmost bit in AC and BR represent the sign bits of the numbers.
- The two sign bits are added or subtracted together with the other bits in the complementer and parallel adder.
- The overflow flip-flop V is set to 1 if there is an overflow. The output carry in this case is discarded.

Algorithm

- The sum is obtained by adding the contents of AC and BR (including their sign bits).
- The overflow bit V is set to 1 if the exclusive-OR of the last two carries is 1, and it is cleared to 0 otherwise.
- The subtraction operation is accomplished by adding the content of AC to the 2's complement of BR.
- An overflow must be checked during this operation because the two numbers added could have the same sign.
- The programmer must realize that if an overflow occurs, there will be an erroneous result in the AC register.
- Comparing this algorithm with signed magnitude, it is simpler to add and subtract numbers if negative numbers are in signed 2’s complement form.
- So the most of the computers adopt this representation over signed magnitude form.