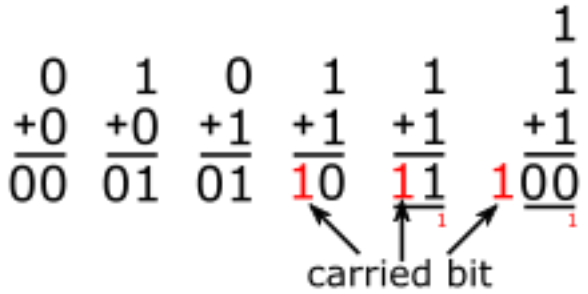


14. Integer Arithmetic

Chapter 10, section 10.3

Arithmetic Logic Unit

- The ALU performs arithmetic and logical operations on data

- All other elements … control unit, registers, memory, I/O mainly bring data to ALU for processing and then take the results back

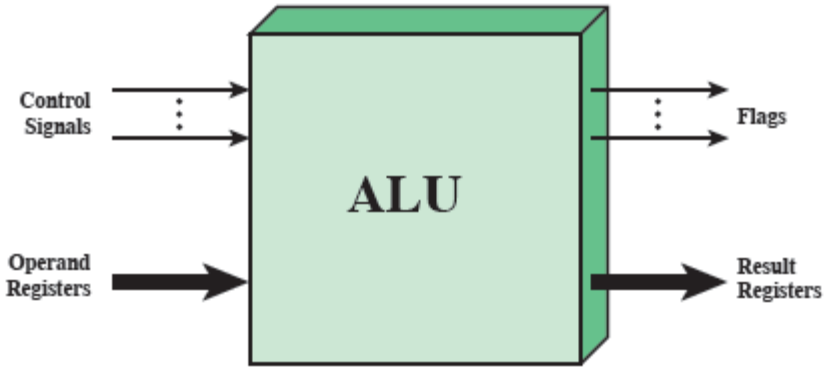


Figure 10.1 ALU Inputs and Outputs

```
  1 0 0 1 0
+   1 0 0 1
-----------
  1 1 0 1 1
-----------
```

ADDITION

Binary Addition

- In general, we know the following is true:

```
0 + 0 = 0 0
0 + 1 = 0 1
1 + 0 = 0 1
1 + 1 = 1 0
```

Integer Addition

- Perform the following addition and interpret the result in:
  a. Unsigned numbers
  b. Two's complement notation

```
   1100
+  0001
   ----
```

Carry-in & Carry-out

- Consider the following:

```
 00001111
+01010101
 --------
```

1. What is the carry-in and carry-out of bit 3 (bit numbering starts at bit 0)
2. The carry-out of the MSB during an addition is the value of the external carry in the flags register for an addition
3. What is the external carry for the above example?

```
  1  0  1  1  0  1  1
-       1  0  0  1  0
---------------------
  1  0  0  1  0  0  1
---------------------
```

SUBTRACTION

Binary Subtraction

- In general, we know the following is true:

```
0 - 0 = 0
0 - 1 = borrow!
1 - 0 = 1
1 - 1 = 0
10 - 1 = 1
```

Subtraction of Unsigned Numbers

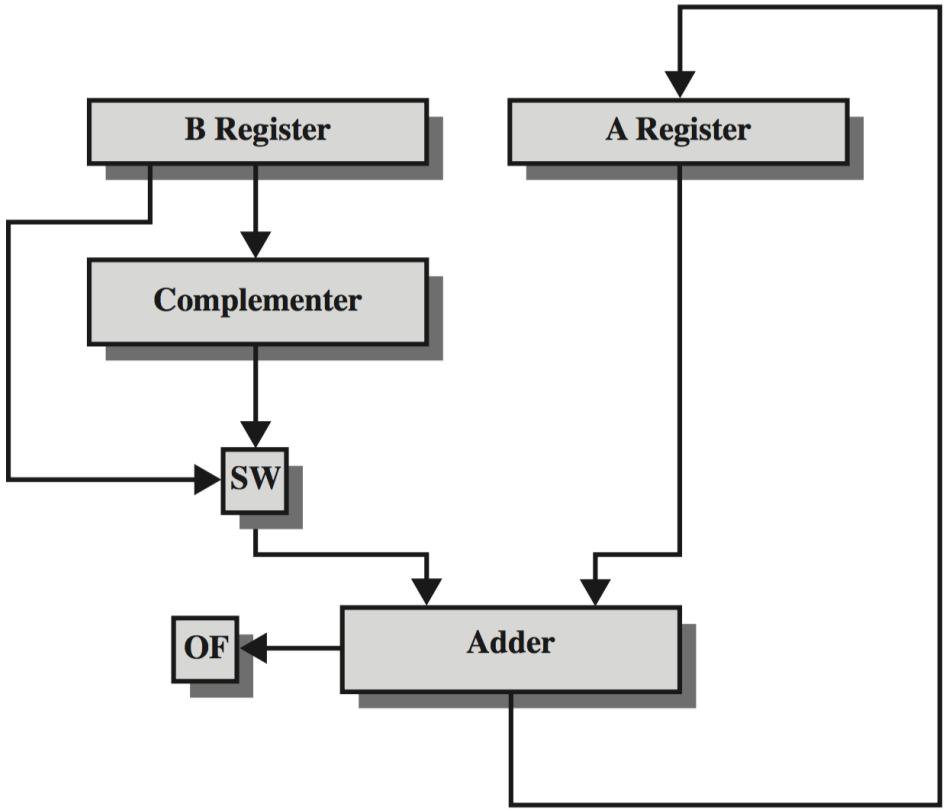
- Perform the following subtraction:

```
 00110011     (Minuend)
-00001111     (Subtrahend)
 --------
```

1. Before performing the subtraction, identify the two numbers being subtracted. Assume the numbers are represented as unsigned numbers.
2. Perform the subtraction.
3. Interpret the result. Is it what you would expect it to be?

Subtraction

- Subtraction is performed by taking the two's complement of the subtrahend and adding this value to the minuend.



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

Subtraction of 2's Complement Numbers

- Perform the following subtraction:

```
 00110011     (Minuend)
-00001111     (Subtrahend)
 --------
```

1. Before performing the subtraction, identify the two numbers being subtracted. Assume the numbers are represented in 2's complement.
2. Perform the subtraction.
3. Interpret the result. Is it what you would expect it to be?

OVERFLOW

Arithmetic Overflow

- Remember that the range of values that can be represented using 8-bits for:
  - Unsigned numbers is 0 to 255
  - Two's complement is -128 to 127.
- The microprocessor will perform the addition or subtraction of two numbers, but the question is how do we know if the result is correct?
- The answer lies with two flags: (a) the external carry flag and (b) the overflow flag.
- First we will define overflow as a condition such that an arithmetic operation produces a result outside the range of the number system being used.

Overflow Detection

- Unsigned Numbers:
  - When there is a carry out of the MSB (external carry)
- 2's Complement:
  - If two numbers are added, and they are both positive or both negative, then overflow occurs if and only if the result has the opposite sign
  - There will never be an overflow when adding two numbers of opposite signs
  - The carry out of the MSB does not necessarily indicate an overflow

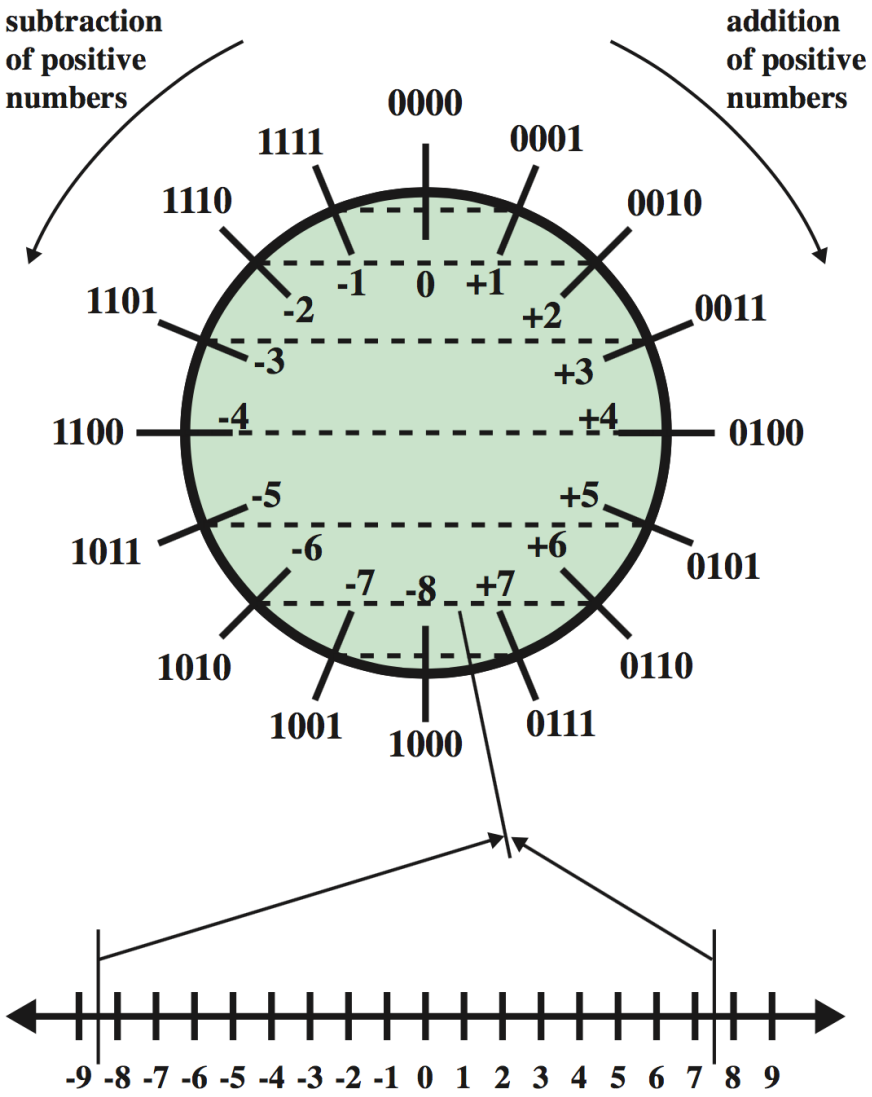
Arithmetic Overflow

- Perform the operations below and interpret the result in:
  1. Unsigned numbers
  2. Two's complement notation

```
 11111111      01111111
+00000001     +00000001
 --------      --------
```

- Were there any examples of overflow? Identify each case and briefly explain why.

Geometric Depiction of 2's Complement Numbers



Practice

- Let's perform the following operations and determine where any overflows occurred for both unsigned and 2's complement representations.

```
 1001          1100
+0101         +0100
-----         -----

 0011          1100
-0100         -1111
-----         -----

 1000          1000
+0001         -0001
-----         -----
```