Number System

Binary Addition

- Addition
  - The two numbers in addition:
    - Augend
    - Addend
  - Result is:
    - Sum
  - Eg:

	0 0 0 0 0 1 1 1	Augend
+	0 0 0 0 0 1 0 0	Addend
	0 0 0 0 1 0 1 1	Sum

  - *Note: Generally, negative numbers are considered in 2's complement representation*

Binary Addition

- Both numbers are positive (consider 8 bit)

```
  0000  0111        (+7)
+ 0000  0100    +   (+4)
-----------     -------
  0000  1011         11
-----------     -------
```

- Positive number with magnitude larger than negative number

```
    0000  1111        (+15)
+   1111  1010    +    (-6)
-------------     --------
1   0000  1001           9
-------------     --------
```

Binary Addition

- Negative number with magnitude larger than positive number

```
  0 0 0 1   0 0 0 0        (+16)
+ 1 1 1 0   1 0 0 0    +   (-24)
-----------------------   -------
  1 1 1 1   1 0 0 0          -8
-----------------------   -------
```

- Both numbers are negative

```
    1 1 1 1   1 0 1 1        (-5)
+   1 1 1 1   0 1 1 1    +   (-9)
-------------------------   -------
1   1 1 1 1   0 0 1 0         -14
-------------------------   -------
```

Binary Addition

- Overflow condition
  - When two numbers are added and the number of bits required to represent that sum exceeds the number of bits in the two numbers, an overflow condition occurs.
  - It can occur only if both numbers are positive or both numbers are negative.
  - If the sign bit of the result is different than the sign bit of the numbers that are added, overflow is indicated.

	0 1 1 1	1 1 0 1		(+125)
+	0 0 1 1	1 0 1 0	+	(+58)
	1 0 1 1	0 1 1 1		183

Incorrect Sign bit

Binary Subtraction

- Special case of addition is subtraction
- Subtraction
  - The two numbers in subtraction:
    - Minuend
    - Subtrahend
  - Result is:
    - Difference
- Eg:

	1 0 1 1 0 1	Minuend
+	1 0 0 1 1 1	Subtrahend
	0 0 0 1 1 0	Difference

Binary Subtraction

- The sign of the number is changed by taking 2's complement
- To subtract 2 numbers
  - take the 2's complement of the subtrahend and add.
  - Discard any final carry

Binary Subtraction

- 00001000-00000011
  - 2's complement of 00000011=11111101

	0 0 0 0	1 0 0 0	(+8)
+	1 1 1 1	1 1 0 1	+ (-3)
1	0 0 0 0	0 1 0 1	+5

Binary Multiplication

- Multiplication
  - The numbers in multiplication:
    - Multiplicand
    - Multiplier
  - Result is:
    - Product
    - (Partial Product)

Binary Multiplication

- 1011 x 101

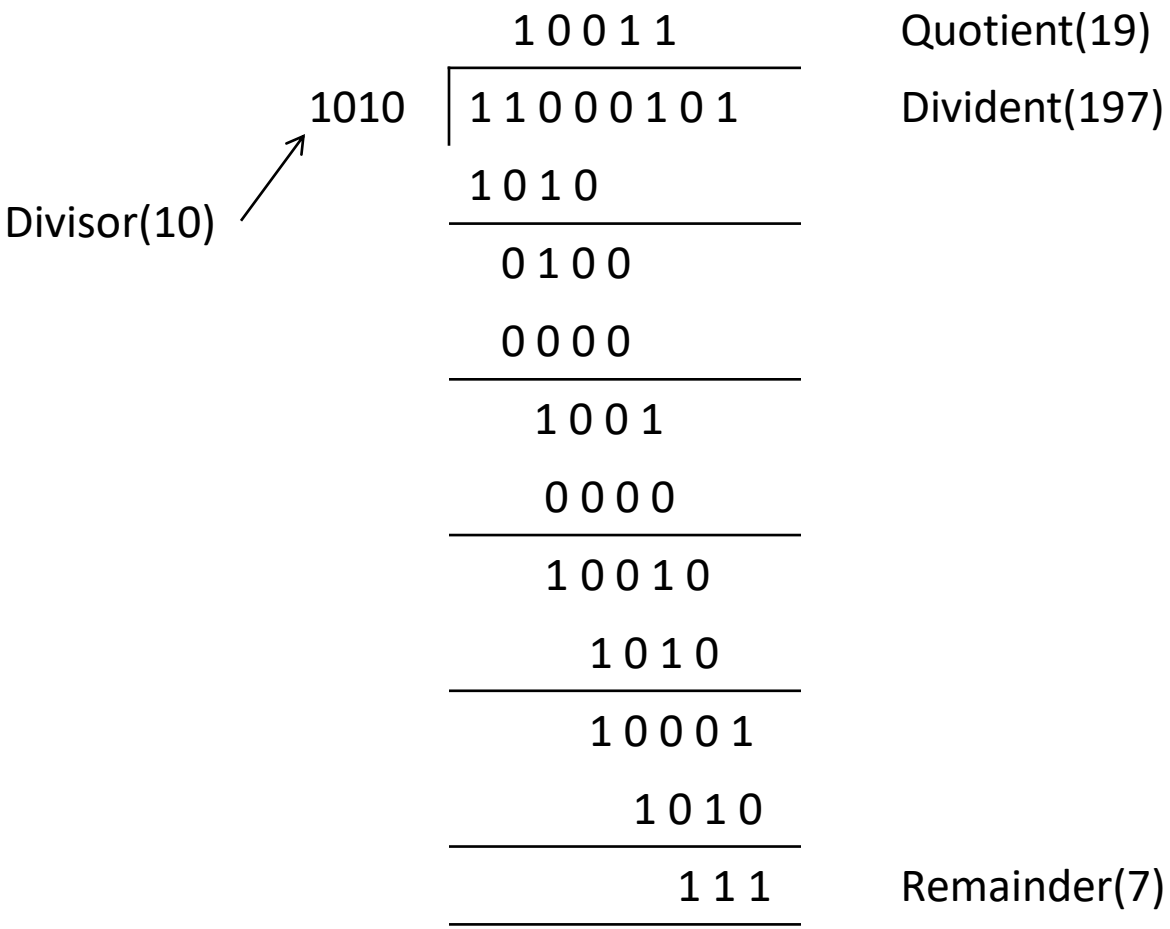
1 0 1 1	Multiplicand(11)
x 1 0 1	Multiplier(5)
1 0 1 1	Partial Product
0 0 0 0	Partial Product
1 0 1 1	Partial Product
1 1 0 1 1 1	Product(55)

Binary Division

- Division
  - The numbers in division:
    - Dividend
    - Divisor
  - Result is:
    - Quotient
    - Remainder

Binary Division

- 11000101 ÷ 1010



Binary Codes

- Any discrete elements of information that is distinct among a group of quantities can be represented with binary codes
- Sample Binary Codes:
  - Binary Coded Decimal(BCD)/8421
  - Gray Code
  - Excess-3 Code
  - 2421 Code
  - ASCII Code

  .

  .

  .

Binary Coded Decimal(BCD)/8421

- Straight binary assignment of the decimal numbers.
- It is a weighted code (codes which obey the positional weight principle.)
- 10 decimal digits requires 4 bits for representation. But 6 out of 16 4-bit possible combination remains unassigned.
  - A number with k decimal digits will require 4k bits in BCD
- Eg:
  - $(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (10111001)_2$
- Applications: Digital clocks, digital meters, Seven segment display etc…(simplify the display of decimal numbers)
- This code is not very efficient but useful if only limited processing is required.

Decimal & BCD

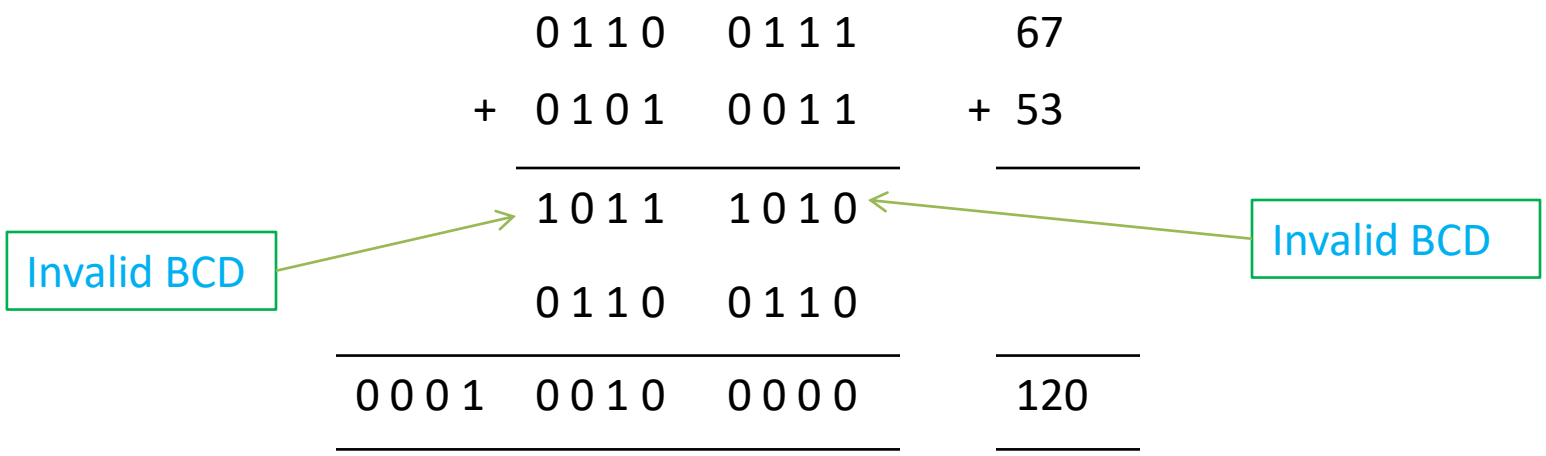
DECIMAL	BCD
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	0 0 0 1 0 0 0 0

BCD Addition

- Add 2 BCD numbers using the rules for binary addition
- If a 4-bit sum is equal or less than 9, it's a valid BCD number
- If a 4-bit sum is greater than 9 or if a carry out of the 4-bit group is generated, it is an invalid result.
  - Add 6(0110) to the 4-bit sum in order to skip the invalid states.
  - If a carry results when 6 is added, simply add the carry to the next 4-bit group

BCD Addition

- 0110 0111 + 0101 0011

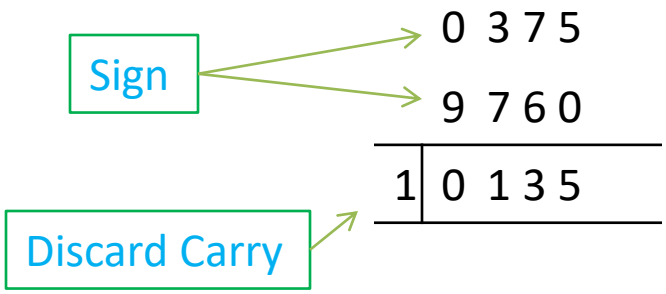


BCD Addition

- The sign of the **decimal number** is represented using 4 bits
  - 0000 represents positive
  - 1001 represents negative
- Sign-magnitude is seldom used in computers
- Sign-complement uses 9's or 10's complement

BCD Addition (10's complement)

- (+ 0011 0111 0101) + (- 0010 0100 0000)
  - 0011 0111 0101 –> 375
  - 0010 0100 0000 –> 240
  - 10's complement of 0240 = 9999-0240+1= 9760



  - Answer:

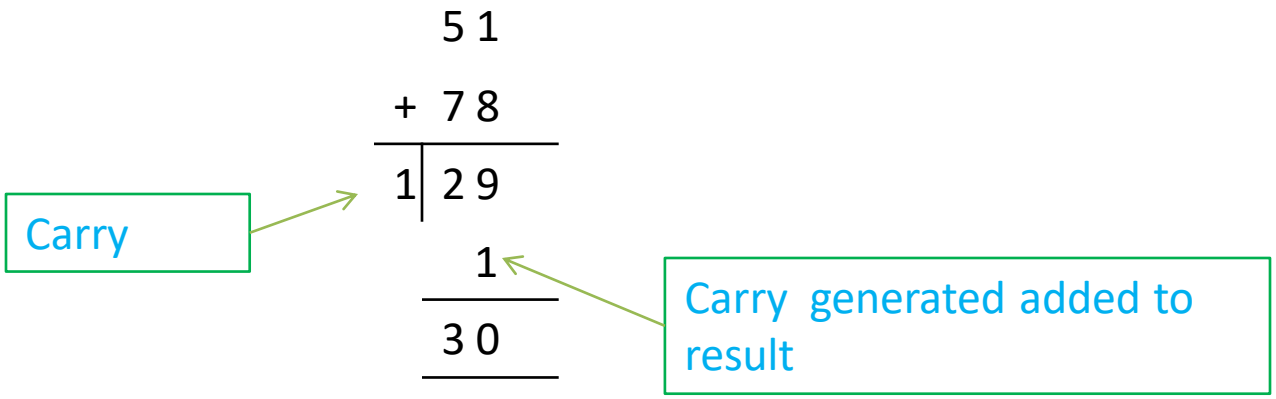
(+0011 0111 0101) + (-0010 0100 0000) = (+0001 0011 0101)

BCD Subtraction

- At first the decimal equivalent of the given Binary Coded Decimal (BCD) codes are found out.
- Then the 10’s compliment of the subtrahend is done and then that result is added to the number from which the subtraction is to be done.
- Discard Carry if generated
- *Note: If 9’s complement is used, carry is added to the result of subtraction!*

BCD Subtraction (9's complement)

- 0101 0001 – 0010 0001
  - 0101 0001 -> 51
  - 0010 0001 -> 21
  - 9's complement of 21 = 99-21 = 78



  - 30 -> 0011 0000
  - Answer: 0101 0001 – 0010 0001 = 0011 0000