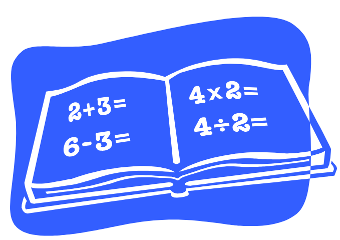
ECNG 1014

DIGITAL ELECTRONICS I

Lecture 3 – Binary Arithmetic

Some Binary Arithmetic

- Addition
- Subtraction
  - Signed magnitude numbers
  - 2's complement numbers
- Multiplication
- Division



Addition

$$0+0=0$$
$$0+1=1$$
$$1+0=1$$
$$1+1=10$$

- Which is really '0' **carry** '1'
- Like 8 + 2 = 10, which is '0' carry '1'

Example

- Try this out using the previous rules:

Decimal	Binary
50	
127	
177	

Example

Fill in the blanks

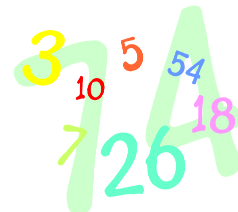
Decimal	Binary
50	00110010
127	01111111
177	

Example

Fill in the blanks

Decimal	Binary
50	00110010
127	01111111
177	10110001

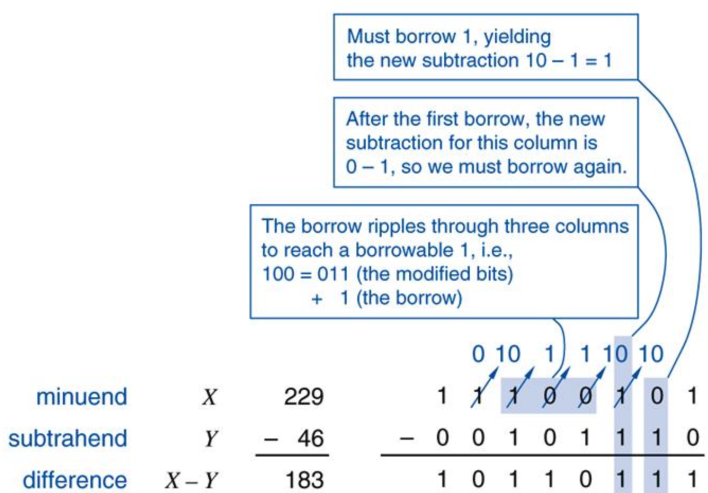
Subtraction



$$0-0=0$$
$$1-0=1$$
$$1-1=0$$

- 0 – 1 = 1 **borrow** ‘1’ which is 10 – 1 = 1

Example 2



Example 3

X	210
Y	$-$ 109
$X - Y$	

Comments

- Addition used to facilitate multiplication of numbers
  - We will see later in this lecture
- Relatively easy to create adder circuits in digital electronics

Comments

- Subtraction used to compare numbers.
  - Example if we have a set point in some engineering system
    - Say 30 Volts
  - How do we know if we have achieved this?
    - Need to compare actual value with set value by subtraction.

Subtraction

- More difficult process than addition
  - Circuitry more complex
- Leads to the **representation** of negative numbers
  - Can then use addition to perform subtraction
- Two ways of representing negative numbers
  - **Signed magnitude**
  - **Complement**

Signed Magnitude

- Use the MSB of the binary bit string to indicate the sign of the number
  - '0' is positive;
  - '1' is negative
- Easier to understand by human user

85	→	01010101
-85	→	11010101

Signed Magnitude

- Has 2 representations for zero

+0	→	00000000
-0	→	10000000

- For a given number of bits, n, lets you cover:

$$-\left(2^{n-1}-1\right)\textbf{ to }+\left(2^{n-1}-1\right)$$

Machine arithmetic with signed-magnitude representation

- Takes several steps to add a pair of numbers
  - Examine signs of the addends
  - If same, add magnitudes and give the result the same sign as the operands
  - If different, must…
    - Compare magnitude of the two operands
    - Subtract smaller number from larger
    - Give the result the sign of the larger operand
- For this reason the signed-magnitude representation is not as popular as one might think because of its "naturalness"

Complement number systems

- Negates a number by *taking its complement* instead of negating the sign
- Exact meaning of *taking its complement* is defined in various ways
- Not natural for humans, but better for machine arithmetic

- We will examine the '**Radix-complement**' system

Radix-complement number representation

- Must first decide how many bits to represent the number – say n.
- Complement of a number = r^n – number
- Example: 4-bit decimal:
  - Original number = 3524
  - 10's complement = 10000-3524 = 6476

Two's-complement representation

- Just radix-complement when radix = 2
- Used a lot in computers and other digital arithmetic circuits
- 0 and positive numbers: leftmost bit = 0
- Negative numbers: leftmost bit = 1
- To find a number's complement – just flip all the bits and add 1
- **Very easy to do in digital electronics**

Subtracting using 2's complement

- Try the following example

229
-46
183

- Discard the MSB leaving 10110111 which is 183_{10}

Binary Multiplication

	229	1 1 1 0 0 1 0 1
x	46	× 0 0 1 0 1 1 1 0
	10534	

Binary Multiplication

1	1	1	0	0	1	0	1							
						×	0	0	1	0	1	1	1	0
0	0	0	0	0	0	0	0							
	0	0	0	0	0	0	0	0						
		1	1	1	0	0	1	0	1					
			0	0	0	0	0	0	0	0				
				1	1	1	0	0	1	0	1			
					1	1	1	0	0	1	0	1		
						1	1	1	0	0	1	0	1	
							0	0	0	0	0	0	0	0
		1	1	0	0	1	0	0	1	0	0	1	1	0

- Exactly like decimal math
- Notice how it is simply a set of shifting and adding operations

Binary Division

19	10011	quotient
11)217	1011)11011001	dividend
11	1011	shifted divisor
107	0101	reduced dividend
99	0000	shifted divisor
8	1010	reduced dividend
	0000	shifted divisor
	10100	reduced dividend
	1011	shifted divisor
	10011	reduced dividend
	1011	shifted divisor
	1000	remainder

- Subtraction done using 2's complement and addition