CS 220 – COMPUTER ARCHITECTURE

WEEK #2: BOOLEAN ARITHMETIC

COUNTING SYSTEMS

quantity	decimal	binary	3-bit register
	0	0	000
*	1	1	001
**	2	10	010
***	3	11	011
****	4	100	100
*****	5	101	101
******	6	110	110
*******	7	111	111
********	8	1000	overflow
*********	9	1001	overflow
**********	10	1010	overflow

POSITIONAL CONCEPT

- Position Matters

  352 in decimal: $(352)_{10}$ or 352_{10}

  $= 300 + 50 + 2$

  $= 3 * 100 + 5 * 10 + 2 * 1$

  $= 3 * 10^2 + 5 * 10^1 + 2 * 10^0$

- Position determines power we raise the base
- Decimal Notation
  - Base (Radix) 10
- Binary Notation
  - Base (Radix) 2

HOW NUMBER SYSTEMS WORK

Let's use 325_{10} again:

325^{10}			
[position$_{\text{numbered from 0}}$]	[2]	[1]	[0]
Digit	3	2	5
Base$^{\text{position}}$	10^2	10^1	10^0
Digit * Base$^{\text{position}}$	300	20	5

Instructions:

1) Fill in top 3 rows

2) Last row = Digit * Base$^{\text{position}}$

3) Add up bottom row for total

TOTAL = 325_{10}

Now let's try it with 0101000101_2:

0101000101_2										
[position]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Digit	0	1	0	1	0	0	0	1	0	1
Base$^{\text{position}}$	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Simplified	*512*	*256*	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*
Digit * Base$^{\text{position}}$	0	256	0	64	0	0	0	4	0	1

TOTAL = 325

What just happened?!

CONVERTING TO DECIMAL

$= 1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0$

- Convert $\mathbf{11001_2}$ to Base 10 (decimal):

$= 2^4 + 2^3 + 0 + 0 + 2^0$

$= 16 + 8 + 1$

$\mathbf{= 25_{10}}$

- **Sum of Expansion of Products**
  - Multiply by base you're converting FROM
  - Anything to the power of 0 is 1
  - Don't skip steps!

EXERCISE #1

- Your Turn!
- Convert $\mathbf{11011_2}$ to decimal
  - Using sum of expansion of products. ***Don't skip steps!***

CONVERTING FROM DECIMAL

- **Dibble Dabble**
  - Successive divisions by the base, track the remainder
- Convert **35_{10}** to Binary:

  35/2 = 17 -- remainder 1

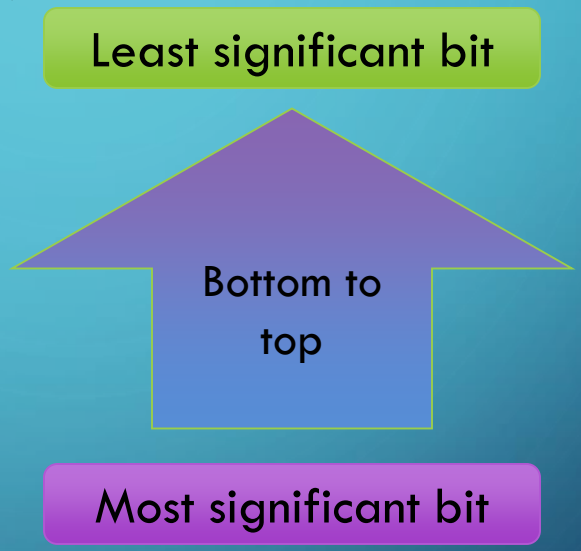
  17/2 = 8 -- remainder 1

  8/2 = 4 -- remainder 0

  4/2 = 2 -- remainder 0

  2/2 = 1 -- remainder 0

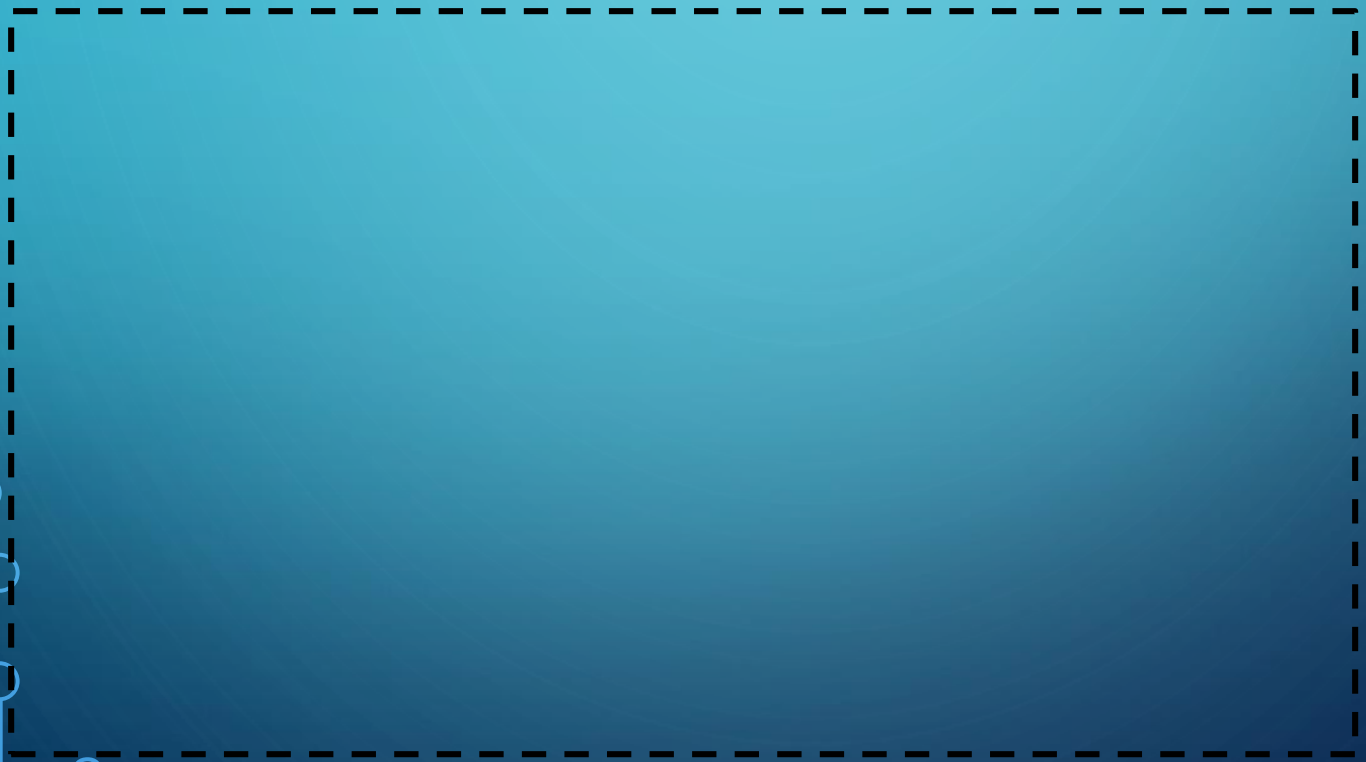
  1/2 = 0 -- remainder 1



- Stop when result is 0
- Thus, 35_{10} = **100011_2**

EXERCISE #2

- Your Turn!
- Convert $\mathbf{172_{10}}$ to binary
  - ***Don't skip steps!***



RATIONALE

$$(9038)_{ten} = 9 \cdot 10^3 + 0 \cdot 10^2 + 3 \cdot 10^1 + 8 \cdot 10^0 = 9038$$

$$(10011)_{two} = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 19$$

$$(x_n x_{n-1} \ldots x_0)_b = \sum_{i=0}^{n} x_i \cdot b^i$$

BINARY ADDITION

Start with decimal addition…

Assuming a 4-bit system:

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

no overflow

overflow

- Algorithm: exactly the same as in decimal addition
- Overflow in the HACK computer is generally ignored to facilitate negative numbers (coming soon).

EXERCISE #3

- Complete the following addition problem:

$$\begin{array}{r} 1011_2 \\ +\ 1001_2 \\ \hline \\ \quad\ \ 2 \end{array}$$

REPRESENTING NEGATIVE NUMBERS (4-BIT SYSTEM)

0	0000		
1	0001	1111	-1
2	0010	1110	-2
3	0011	1101	-3
4	0100	1100	-4
5	0101	1011	-5
6	0110	1010	-6
7	0111	1001	-7
		1000	-8

- Sign Bit
  - The codes of all positive numbers begin with a "0"
  - The codes of all negative numbers begin with a "1"
- To convert a number:
  leave all trailing 0's and first 1 intact, and flip all the remaining bits
  - ***Easier way: flip all bits and add 1***
- 2's Complement

Example: 2 - 5 = 2 + (-5) =

```
  0 0 1 0
+ 1 0 1 1
---------
  1 1 0 1   =  -3
```