Announcements

- Quiz 1
  - Monday, September 8
  - Lecture 2, Homework 1 & 2
- TA sessions: 7:00 p.m. – 9:00 p.m. (MECC 124)

Date	TA
Sunday	Anupam Khargharia
Monday	Kamilla Volkova
Tuesday	Chi Le
Wednesday	Shivanshu Dwivedi

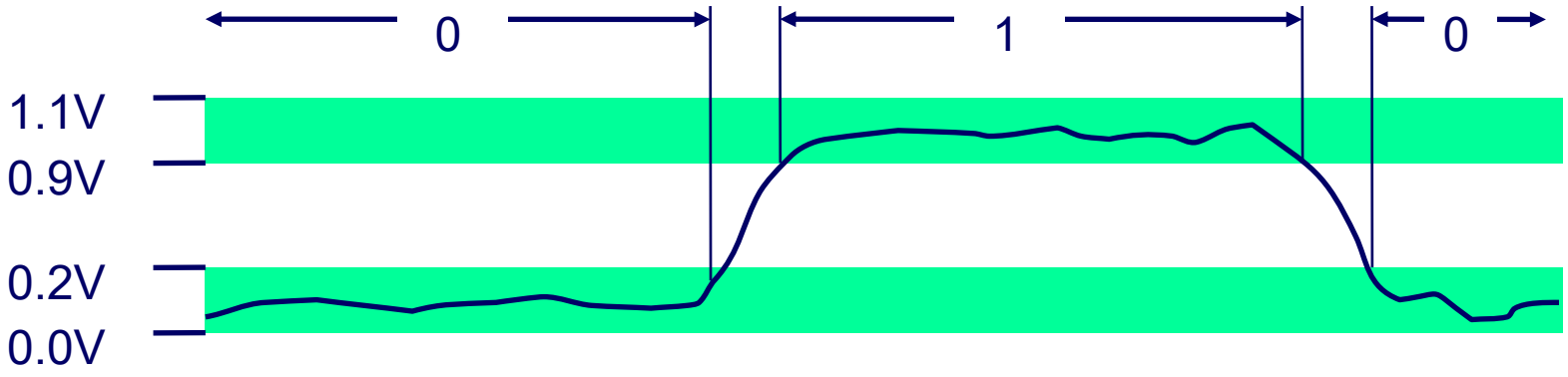
Lecture 2

Bit Operations

CPSC 275
Introduction to Computer Systems

Everything is bit

- Each *bit* is 0 or 1
- By encoding/interpreting sets of bits in various ways
  - Computers determine what to do (instructions) and represent and manipulate numbers, sets, strings, etc.
- Why bits? Electronic Implementation
  - Easy to store with bi-stable elements
  - Reliably transmitted on noisy and inaccurate wires



Encoding Byte Values

- *Byte* = 8 bits
  - Binary 00000000_2 to 11111111_2
  - Decimal: 0_{10} to 255_{10}
  - *Hexadecimal*: Base 16
    - Use characters '0' to '9' and 'A' to 'F'
    - 00_{16} to FF_{16}
    - e.g. write $FA1D37B0_{16}$ in C as

      0xFA1D37B0 or **0x**fa1d37b0

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Pop Quiz

- Convert the decimal 199 to hexadecimal.

$$199_{10} = 11000111_2$$
$$= \underline{C7}_{16}$$

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Pop Quiz

- Convert the binary 1011101_2 to hexadecimal.

  $1011101_2 = \underline{5D}_{16}$

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Pop Quiz

- Convert the binary 1011101_2 to hexadecimal.

  $1011101_2 = \underline{5D}_{16}$

- Convert the hexadecimal $FACE_{16}$ to binary.

  $FACE_{16} = \underline{1111101011001110}_2$

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

HTML color codes

- HTML color codes are a triplet of the form:

  where each color channel is a hexadecimal.

- Examples

#FF0000 Red	#00FF00 Green	#0000FF Blue
#FFFFFF White	#000000 Black	#808080 Gray
#00FFFF Aqua	#800080 Purple	#FFFF00 Yellow

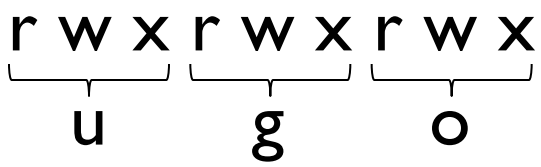
Octal

- *Octal*: Base 8
  - Binary 000_2 to 111_2
  - Decimal: 0_{10} to 7_{10}
  - Octal constants in C:
    - e.g., write 123_8 as **0**123

Octal	Decimal	Binary
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111

File permissions in Linux

- Every file and directory in Linux is associated with a triplet of the form:



where each defines file permission for the user, group, and others, respectively.

- Examples

0755 = 111101101 The user has full access; group and others have read and execute permissions

Boolean Algebra

- Developed by George Boole in 19th Century
  - Algebraic representation of logic
    - Encode "True" as 1 and "False" as 0

AND

A&B = 1 when both A=1 and B=1

&	0	1
0	0	0
1	0	1

OR

A|B = 1 when either A=1 or B=1

|	0	1
0	0	1
1	1	1

NOT

~A = 1 when A=0

~	
0	1
1	0

XOR (Exclusive-Or)

A^B = 1 when either A=1 or B=1, but not both

^	0	1
0	0	1
1	1	0

General Boolean Algebras

- Operate on Bit Vectors
  - Operations applied bitwise

```
  01101001     01101001     01101001
& 01010101   | 01010101   ^ 01010101   ~ 01010101
----------   ----------   ----------   ----------
  01000001     01111101     00111100     10101010
```

Bit-Level Operations in C

- Operations `&`, `|`, `~`, `^` available in C
  - Apply to any "integral" data type
    - `int`, `short`, `char`, `unsigned`, `long`
  - View arguments as bit vectors
  - Arguments applied bit-wise

- Examples (`char` data type)

~0x41 & 0xBE ~01000001_2 & 10111110_2

0x55 ^ 0x7D 01010101_2 ^ 01111101_2

Contrast: Logic Operations in C

- Contrast to logical operators: &&, ||, !
  - View 0 as "False"
  - Anything nonzero as "True"
  - Always return 0 or 1

- Examples (`char` data type)

  0x00 && !0x41 = 0 (short-circuit evaluation)

  !0x00 || 0x01 = 1

  0x69 && 0x55 && 0x01 = 1

  0x69 || 0x55 && 0x01 = 1