

# Bits and Bytes

## Aug. 28, 2003

### Topics

- Why bits?
- Representing information as bits
  - Binary/Hexadecimal
  - Byte representations
    - numbers
    - characters and strings
    - instructions
- Bit-level manipulations
  - Boolean algebra
  - Expressing in C

# Encoding Byte Values

Byte = 8 bits

- Binary  $00000000_2$  to  $11111111_2$
- Decimal:  $0_{10}$  to  $255_{10}$
- Hexadecimal  $00_{16}$  to  $FF_{16}$ 
  - Base 16 number representation
  - Use characters '0' to '9' and 'A' to 'F'
  - Write  $FA1D37E_{16}$  in C as

## Data Representations

### Sizes of C Objects (in Bytes)

■ C Data Type	Compaq Alpha	Typical 32-bit	Intel IA32
■ int	4	4	4
■ long int	8	4	4
■ char	1	1	1
■ short	2	2	2
■ float	4	4	4
■ double	8	8	8
■ long double	8	8	10/12
■ char *	8	4	4
> Or any other pointer			

## Byte Ordering

How should bytes within multi-byte word be ordered in memory?

### Conventions

- Sun's, Mac's are "Big Endian" machines
  - Least significant byte has highest address
- Alpha, PC's are "Little Endian" machines
  - Least significant byte has lowest address

## Byte Ordering Example

### Big Endian

- Least significant byte has highest address

### Little Endian

- Least significant byte has lowest address

### Example

- Variable