Number Systems

- A modern computer is just a vast collection of electronic switches.
- A switch has just two states
  - on
  - off
- We can represent these two states by two symbols
  - on = 1
  - off = 0

Number Systems (2)

- So how do we count with just two symbols?
  - The same way we count with ten symbols!
- In the decimal system we use everyday we have ten symbols.
  - 0 1 2 3 4 5 6 7 8 9 (ordered)
  - The following combination of these symbols represent the number one thousand four hundred and sixty two
    - 1462
    - $(1 \times 10^3) + (4 \times 10^2) + (6 \times 10^1) + (2 \times 10^0)$

Number Systems (3)

- A number system with ten symbols is a decimal system.
- A number system with two symbols is a binary system.
  - 0 1 (ordered)
- A ***base x*** number system has ***x*** symbols.
- What number does 11 represent?
  - That depends on the base of the number system!
  - To avoid confusion we write the base as a subscript after the symbols, i.e. 11_{10}

Number System (4)

- If there is no base attached to a number we will assume it is a base 10 number.
- The number systems we use all follow the same basic rules.
  - The rightmost symbol represents the number of $base^0$ units.
  - If the symbol to the right represent $base^i$ units then the current symbol represents $base^{i+1}$ units.
  - The number is the sum of all the units represented by each symbol.

Converting Binary to Decimal

- What decimal number does 1011_2 represent?
  - $(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$
  - $(1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1)$
  - $11_{(10)}$
- Convert 11001010_2 to decimal.
  - $(1 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$
  - $(1 \times 128) + (1 \times 64) + (0 \times 32) + (0 \times 16) + (1 \times 8) + (0 \times 4) + (1 \times 2) + (0 \times 1)$
  - 202

Converting Decimal to Binary

- You need to know the powers of 2.
  - 2^0=1 2^1=2 2^2=4 2^3=8 2^4=16 2^5=32 2^6=64 2^7=128
- General algorithm for converting a number to a base *n* number.
- First, find the smallest power of *n* that is larger than the number you are converting.
- Divide the next smallest power of *n* into the number to be converted. Record result and keep remainder.
- Repeat previous step while power of n is positive.
- Result is the recorded numbers, left to right.

Converting Decimal to Binary (2)

- Convert 175_{10} to binary.
- $2^7(=128) < 175 < 2^8(=256)$
- $175 / 2^7(=128) = 1$ remainder 47
- $47 / 2^6(=64) = 0$ remainder 47
- $47 / 2^5(=32) = 1$ remainder 15
- $15 / 2^4(=16) = 0$ remainder 15
- $15 / 2^3(=8) = 1$ remainder 7
- $7 / 2^2(=4) = 1$ remainder 3
- $3 / 2^1(=2) = 1$ remainder 1
- $1 / 2^0(=1) = 1$ remainder 0
- Result: 10101111

Converting Decimal to Binary (3)

- Convert 83_{10} to binary.
- $2^7(=64) < 83 < 2^7(=128)$
- $83 / 2^6(=64) = 1$ remainder 19
- $19 / 2^5(=32) = 0$ remainder 19
- $19 / 2^4(=16) = 1$ remainder 3
- $3 / 2^3(=8) = 0$ remainder 3
- $3 / 2^2(=4) = 0$ remainder 3
- $3 / 2^1(=2) = 1$ remainder 1
- $1 / 2^0(=1) = 1$ remainder 0
- Result: 1010011

Octal

- Octal is a base 8 number system.
  - Symbols are: 0 1 2 3 4 5 6 7
- Converting octal to decimal.
  - 672_8
  - $(6 \times 8^2) + (7 \times 8^1) + (2 \times 8^0)$
  - $(6 \times 64) + (7 \times 8) + (2 \times 1)$
  - $384 + 56 + 2$
  - 442

Octal (2)

- Converting decimal to octal
  - 788
  - $8^3(=512) < 788 < 8^4(=4096)$
  - $788 / 8^3 = 1$ remainder 276
  - $276 / 8^2 = 4$ remainder 20
  - $20 / 8^1 = 2$ remainder 6
  - $6 / 8^0 = 6$ remainder 0
  - Result: 142_8

Octal (3)

- Converting between binary and octal.
  - Since 8 is a power of 2, converting between binary and octal is straight forward.
  - binary to octal
    - from right to left group the binary digits in groups of 3.
    - convert each 3 digit binary grouping into an octal number.

110101011110_2

110	101	011	110

$_2$

6536_8

Octal (4)

- octal to binary
  - convert each octal digit into a binary number

5271_8

101	010	111	001

$_2$

101010111001_2

Hexidecimal

- Hexidecimal is a base 16 number system.
  - Symbols: 0 1 2 3 4 5 6 7 8 9 A B C D E F
    - The symbol A represent 10_{10}.
    - The symbol B represent 11_{10}.
    - The symbol C represent 12_{10}.
    - The symbol D represent 13_{10}.
    - The symbol E represent 14_{10}.
    - The symbol F represent 15_{10}.

Hexidecimal (2)

- Converting hexidecimal to decimal.
  - $6CA_{16}$
    - $(6 \times 16^2) + (12 \times 16^1) + (10 \times 16^0)$
    - (6 x 256) + (12 x 16) + (10 x 1)
    - 1536 + 192 + 10
    - 1738
  - $7FFF_{16}$
    - $(7 \times 16^3) + (15 \times 16^2) + (15 \times 16^1) + (15 \times 16^0)$
    - (7 x 4096) + (15 x 256) + (15 x 16) + (15 x 1)
    - 28672 + 3840 + 240 + 15
    - 32767

Hexidecimal (3)

- Converting decimal to hexidecimal
  - 49728
  - 16^3(=4096) < 49728 < 16^4(=65536)
  - 49726 / 16^3(=4096) = 12 remainder 576
  - 576 / 16^2(=256) = 2 remainder 64
  - 64 / 16^1 = 4 remainder 0
  - 0 / 16^0 = 0 remainder 0
  - Result: $C240_{16}$

remember to convert decimal numbers into corresponding hexidecimal symbols.

Hexidecimal (4)

- Converting between hexidecimal and binary.
  - Again straight forward since 16 is a power of 2.
  - binary to hexidecimal
    - from right to left group the binary digits in groups of 4.
    - convert each 4 digit binary grouping into a hexidecimal number.

1101010111101001_2

1101	0101	1110	1001

$D5E9_{16}$

Hexidecimal (5)

– hexidecimal to binary
  - convert each hexidecimal symbol into a binary number.

$D3F9_{16}$

1101	0011	1111	1001

$_2$

1101001111111001_2

– hexidecimal to octal
  - convert the hexidecimal number into a binary number; and then convert to octal.

– octal to hexidecimal
  - convert the octal number into a binary number; and then convert to hexidecimal.

Bits, Bytes and Words

- So <u>why</u> are we interested in these number systems?
  – Binary
    - Because a computer is a massive collection of electronic switches, with each switch having two states; on (1) and off (0).
  – Octal and Hexidecimal
    - Useful ways to encode binary numbers.
    - 1,000,000 is 11110100001001000000_2 and $F4240_{16}$.
- A ***bit*** is a single binary digit.

Bits, Bytes and Words (2)

- A ***byte*** is an 8 digit binary number.
  - A byte is usually written in hexidecimal.
    - The byte 11001010_2 is also written as CA_{16}.
- A ***word*** is a fixed size collection of bits, usually 2 bytes, 4 bytes or 8 bytes, depending on the internal structure of the computer.
  - A word is usually written in hexidecimal.
  - The available word sizes are a defining characteristic of the architecture (internal structure) of a computer.