Numbers in the Machine

CS1A

- Electronic Devices (binary)
- Character Representations

Image: Public Domain

A Very Simple Computer

- Computers are Electronic devices
  - Consisting of many switches
  - Naturally binary
    - Electrical on & off

Image: public Domain

- Think of a computer as a group of switches
  - Each switch is either on or off
  - On is represented with a 1
  - Off is represented with a 0

Image: public Domain

- The most basic computer can be represented as a machine with 1 switch
  - The computer could only understand two instructions (on or off)

Single Switch House

- Let's picture a house that has only one light switch.

Instructions

0 (lights off)

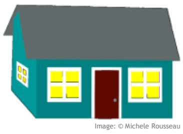
1 (lights on)

Image: © Michele Rousseau

Note: 1 switch = 2 instructions

Let's expand that a bit

2 switches



Instructions

00 (Front & Back lights off)
01 (Front on, Back off)
10 (Front off, Back on)
11 (Front & Back lights on)

Note: 2 switches = 4 or 2 x 2 or 2^2 instructions

(c) Michele Rousseau Numbers in the Machine 4

And One More

- 3 Switches

Instructions

Front Middle Back

000
001
010
011
100
101
110
111

Based on this ..
How many instructions
Would a computer with
16 switches understand?

Note: 3 switches
= 2 x 2 x 2 = 2^3 = 8 instructions

Michele Rousseau Numbers in the Machine 5

Bits, Bytes and ASCII

- A single Binary Digit is a *Bit*
- It takes 8 Bits to store a single character
  - 8 Bits is called a *Byte*
- Each character has a numerical representation on an *ASCII chart*

ASCII → American Standard Code of Information Interchange

- Each code consists of a 7-bit code that represents every number, letter, and symbol
- The 8th bit is a check bit (used for error checking)
- Extended version of ASCII → 256 Characters

This is sufficient for English → but not for foreign languages

(c) Michele Rousseau Numbers in the Machine 6

Unicode

- Unicode was developed for international use → Because ASCII did not have enough characters
- Unicode uses 2^{16} characters (that's 65,000!)
- The first 256 (2^8) characters correspond to ASCII
- Not all of the codes have been used so UNICODE can be expanded

Numbers & the Machine

- We think in decimal (Base 10)
- Computers use Binary (Base 2) - 0's & 1's
- We need to be able to convert from decimal to binary
- Since Binary numbers get long fast…
  - Eg (10000001) = 129
  - It is easier for us to convert from binary to octal or hexadecimal equivalents
    - → than from binary to decimal
  - It is easier to convert to binary from oct and hex numbers
    - → than from decimal to binary

The Decimal System

- Brought to us by the Hindus (400 A.D.)
- They Arabs picked up on it in (800 A.D.)
- And Europeans discovered it in (1200 A.D.)
- It is a great system because
  - It is easy to deal with large quantities with relatively few symbols
  - It is easy to carry and borrow
  - The decimal systems makes use of 10 digits (0-9)
  - These symbols have a "place value" or "positional concept" which allow us to represent any whole number
  - The term digit (relates to fingers and toes) → we have 10 so this system makes sense to us

Positional concept

- Position MATTERS
  - 352 in decimal
    = 300 + 50 + 2
    = $3 \times 10^2 + 5 \times 10^1 + 2 \times 10^0$
  - The position determines the power in which we raise the base
  - 10 = ten because we use base 10
    - If we used base 2, 10 would = 2
- Base (AKA Radix)

Positional Notation

Rules for Positional Notation:

1. The number of distinct symbols equals the base
   (in base 10 they are 0,1,2,3,4,5,6,7,8,9)
2. The largest value represented by 1 symbol is one less than the base
   (for base 10 this is 9)
3. Each value of a number is multiplied by the base raised to the appropriate power relative to its position
   (e.g. 352= $3 \times 10^2 + 5 \times 10^1 + 2 \times 10^0$)
4. The symbols 10 represents the base
   (for base 10 this is ten)

Decimal	Binary	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

How Number Systems Work

Think of 325_{10}

NOTE: The exponent is 1 less than the position

3	2	5
10^2	10^1	10^0
100	10	1

We multiply these 2 #s in each col & Add

Base 10

Now let's try that in Binary

0	1	0	1	0	0	0	1	0	1
2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1

Sum of the Expansion of the Products

Converting TO Decimal is easy

- Just think about the positional notation

Convert 11001_2 to base 10

- $11001_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
  $= 2^4 + 2^3 + 0 + 0 + 1$
  $= 16 + 8 + 1$
  $= 25_{10}$

This denotes Base 2

Anything raised to the 0^{th} power = 1

Sum of the Expansion of the Products

- Works for any base

Converting to base 10 (decimal)

- $125_8 = 1 \times 8^2 + 2 \times 8^1 + 5 \times 8^0$
  $= 1 \times 64 + 2 \times 8 + 5 \times 1$
  $= 64 + 16 + 5$
  $= 85_{10}$

Anything raised to the 0^{th} power = 1

- $AB_{16} = A \times 16^1 + B \times 16^0$
  $= 10 \times 16 + 11 \times 1$
  $= 160 + 11$
  $= 171_{10}$

Exercises

Convert to decimal using the sum of the expansion of products

- 110111_2
- 163_8
- $5C_{16}$

Dibble Dabble

To convert FROM decimal to any base we use the "dibble-dabble" method to

- DIBBLE-DABBLE
  - ➔Use successive divisions by the base
  - ➔ Keep track of the remainder

Convert 35_{10} to binary

35/2 = 17 -- remainder 1 (least significant bit)
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 (most significant bit)
Thus, $35_{10} = 100011_2$

Read from the bottom Up

You can use the same method for Octal or Hexidecimal

Dibble Dabble Oct & Hex

- 435_{10} Decimal to Octal
  - Dibble - Dabble Method

435 / 8 = 54 - REMAINDER 3
54 / 8 = 6 - REMAINDER 6
6 / 8 = 0 - REMAINDER 6

Reverse the numbers $435_{10} = 663_8$

Read from the bottom Up

- 435_{10} Decimal to Hex
  - Dibble - Dabble Method

435 / 16 = 27 - REMAINDER 3
27 / 16 = 1 - REMAINDER 11 = B
1 / 16 = 0 - REMAINDER 1

Reverse the numbers $435_{10} = 1B3_{16}$

Read from the bottom Up

Alternate Method

- 435_{10} FROM Decimal to Binary - Alternative method
- Think of the positional notation
  - Step 1: Start with 1, multiply by each successive position by 2 do this until you get to a # greater than or equal to the # you are converting.
  - Step 2: Put a 1 under the largest value that you can subtract without having a negative result.
  - Step 3: Subtract that value from the # you are converting
  - Step 4: Repeat until you get to 0

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1
0	1	1	0	1	1	0	0	1	1

To go back to Decimal use the opposite approach
-Just write these #s above your binary the locations
- Add the positions w/ 1s

512 is too big so 256
435 - 256 = 179
179 - 128 = 51
64 is too big so go to 32

51 - 32 = 19
19 - 16 = 3
3 - 2 = 1
1 - 1 = 0
We're Done!

110110011

(c) Michele Rousseau Numbers in the Machine 19

Exercise #1

Convert 172_{10} from decimal to hex, oct and binary

HEX

BIN

OCT

(c) Michele Rousseau Numbers in the Machine 20

Exercise #2

Convert 201_{10} from decimal to hex, oct and binary

BIN

HEX

OCT

(c) Michele Rousseau Numbers in the Machine 21

Addition Base 10

- Process of Adding

Step 1: Add the two digits in the first position.
Check if the solution is greater than or equal to the base.
If not then just write down the solution. You're done.
Otherwise go to step 2.
Step 2: Carry 1.
Subtract the base from the solution found in step one.
Write this down – this is your solution for that position.
Step 3: Add 1 to the position to the left.
Step 4: Repeat this for all positions.

Base 10

$$\begin{array}{r} {\scriptstyle 1\,1} \\ 199_{10} \\ +22_{10} \\ \hline 221_{10} \end{array}$$

Add the carry too

9+2 = 11 – this is > 9 (base -1) , so we have to carry
11-10=1 write down 1 & add 1 to the position to the left.
9+2 +1 = 12 - this is > 9 so we have to carry.
12-10=2 so write down 1 & add 1 to the position on the left
1+1 = 2 Not > 9 so no carry

Addition Bases 8 & 2

- Same as in Base 10 except you carry when you reach the base #

Base 8

$$\begin{array}{r} {\scriptstyle 1\,1} \\ 162_{8} \\ +\ 47_{8} \\ \hline 231_{8} \end{array}$$

Add the carry too

7 + 2 = 9 — 9 is greater than 7(base-1) → CARRY
9-8 =1 (solution –base) – write down 1
Add 1 to the next position
6 + 4 + 1 = 11 — 11 > 7, so 11 – 8 = 3, We write down 3. carry the 1
1 + 1 = 2

Base 2

$$\begin{array}{r} {\scriptstyle 1\,1} \\ 1001_{2} \\ +1011_{2} \\ \hline 10100_{2} \end{array}$$

1+1 = 2 — 2>1 → carry, so 2-2= 0 – write down 0, carry1
0+1+1 = 2 — Same as above
1+0 = 1 — No Carry
1+1 = 1 — 2>1→ again, 2-2=0, write 0 & Carry the 1

Addition Base 16

- Same as in Base 10 except you carry when you reach the base # (in this case if the # > 15)
- Note in Base 16 we count 0-9 just like in decimal then we represent 10 with an A, 11=B, 12=C, 13=D, 14=E, 15=F

$$\begin{array}{r} {\scriptstyle 1\,1} \\ AE9_{16} \\ +\ 67_{16} \\ \hline B50_{16} \end{array}$$

The Base

E=14

9 + 7 = 16 — 16 is greater than 16(base-1) → CARRY
16-16 = 0 (solution–base)→ write 0
Add 1 to the next position
E + 6 + 1 = 21 — 21 > 15, so 21 – 16 = 5, We write 5 carry the 1
A + 1 = 11 — 11=B, so we write B

$$\begin{array}{r} {\scriptstyle 1\,1} \\ FFA_{16} \\ +A9_{16} \\ \hline 10A3_{16} \end{array}$$

A+9 = 19 — 19>15 → carry, so 19-16= 3 – write down 3, carry 1
F+A+1 = 15+10+1=26 — 26>15 → carry, so 26-16=10 - write down A, carry 1
F+1 = 16 — 16>15, 16-16=0, so 0, carry 1

Exercises

- Same as in Base 10 except you carry when you reach the base #

Base 8

$$\begin{array}{r} 125_8 \\ +\ 72_8 \\ \hline \end{array}$$

Base 16

$$\begin{array}{r} F9E_{16} \\ +\ 71_{16} \\ \hline \end{array}$$

Base 2

$$\begin{array}{r} 11011_2 \\ +1001_2 \\ \hline \end{array}$$

Subtraction Base 10

- Basic Process of Subtraction

Step 1: Check the two digits in the first position to see if you have to borrow. If not then just subtract. If so, go to step 2.
Step 2: Borrowing 1 – if the position on the left > 0.
Subtract 1 from the position to the left.
Add the base to the current position and subtract.
Step 3: If the position on the left = 0.
Then borrow from the next position.
Step 4: Repeat this for all positions.

$$\begin{array}{r} \scriptstyle 1\ \cancel{10}^{\,9}\ 10 \\ \cancel{2}\ 0\ 1_{10} \\ -\ \ 2\ 3_{10} \\ \hline 1\ 7\ 8_{10} \end{array}$$

1-3 = -2
-2<0 , so we have to borrow. The position on the left (the 10s) = 0 so we have to borrow the base from the position to the left (the hundreds).
* Remember to borrow we subtract 1 and borrow the base
In this case, we borrow from the 100s, so 2-1=1, give 10 to the 10s position. Borrow again so 10-1=9, and add the base (10 to the ones position).

Borrowed

So now we have 10+1-3 =8
9-2 = 7
1-0 = 1

Subtraction Base 16

Same as in Base 10 except you borrow the base (in this case 16)
Again, note in Base 16 we count 0-9 just like in decimal then we represent 10 with an A, 11=B, 12=C, 13=D, 14=E, 15=F

$$\begin{array}{r} \scriptstyle 13\ 16 \\ A\ \cancel{E}\ 9_{16} \\ -1\ 6\ A_{16} \\ \hline 9\ 7\ F\ _{16} \end{array}$$

9 - A =
9 – 10= -1 Need to Borrow.
Subtract 1 from the position to the left (E-1=D)

Borrowed

Add the base, 16 + 9 – A = 16 + 9 - 10 = 15 = F
D -6 = 13-6 = 7 -- Don't need to borrow
A-1 = 10-1 = 9

$$\begin{array}{r} \scriptstyle 16 \\ \scriptstyle 14\ 14\ 16 \\ \cancel{F}\ \cancel{F}\ A_{16} \\ -\ 2\ F\ C_{16} \\ \hline C\ F\ E_{16} \end{array}$$

A-C= 10-12=-2 Need to borrow,
so F-1=E,

Borrowed

16 + A – C = 16 + 10 – 12 = 14 = E
E-F =15-16=-1 Need to borrow,
so F – 1 = E
16 + E - F = 16 + 14 - 15 = 15 = F
E - 2 = 14- 2 = 12 = C

Exercises

- Same as in Base 10 except you borrow when you reach the base

Base 8

$$\begin{array}{r} 551_8 \\ \underline{-\ 1\,2_8} \end{array}$$

Base 16

$$\begin{array}{r} F\,0\,3_{16} \\ \underline{-\ \ 1\,A_{16}} \end{array}$$

Base 2

$$\begin{array}{r} 1001_2 \\ \underline{-\ \ 101_2} \end{array}$$

For More Help

http://www.saddleback.edu/faculty/lperez/algebra2go/compsci/index.html

+ Additional links online

Announcement

- Quiz next class on the Numbers in the Machine
- Lab on Simple computer -> Due next class