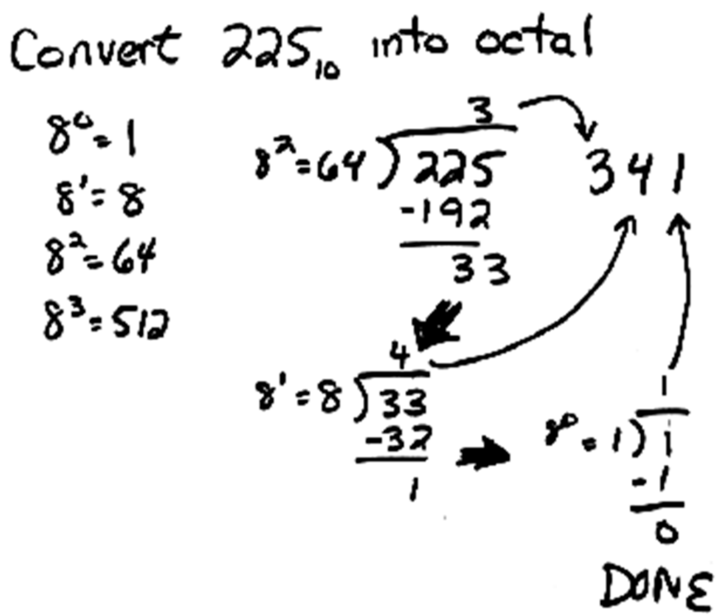
Numbers!!....is that right?
CSCI 255



BASE10

- What do we know and use all the time?

0, 1, 2, 3, 4, 5, 6, 7, 8, 9 → zero through nine….Decimals!

- Position matters too….

5 8 7 4 2 6 3

3×10^0 Ones

6×10^1 Tens

2×10^2 Hundreds

4×10^3 Thousands

7×10^4 Ten Thousands

8×10^5 Hundred Thousands

5×10^6 Million

Decimal value

Base10 factor

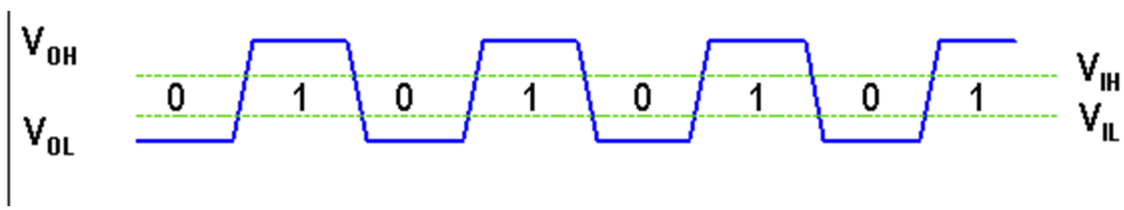
Exponent: significant position

- We understand decimal/base10 systems well…
- Computer systems understand numbers differently
- Various numerical systems are used for computational purposes
- Digital Languages:
  - Binary/BASE2
  - Octal/BASE8
  - Hexadecimal/BASE16

BASE2

- Binary / BASE2 / Digital numbers
- Since it is base2 → Only two values exist in binary → 0 & 1
- Why use only two values?

Physical logic on hardware – Voltage values



...and the logic is like a switch: on/off
true/false
yes/no

BASE2

- How to read binary numbers

1100101_2

1×2^0 — $2^0 = 1$

0×2^1 — $2^1 = 2$

1×2^2 — $2^2 = 4$

0×2^3 — $2^3 = 8$

0×2^4 — $2^4 = 16$

1×2^5 — $2^5 = 32$

1×2^6 — $2^6 = 64$

Binary value

Base2 factor

Exponent: significant position

Adding all values:

$1 + 0 + 4 + 0 + 0 + 32 + 64 = 101_{10}$

Therefore:
Binary 1100101_2 = Decimal 101_{10}

(binary-to-decimal conversion)

- Some Binary Terminology:

Bit =>	a single digit binary value (0 or 1)
Nibble =>	a group of 4 bits
Byte =>	a group of 8 bits or 2 nibbles
Word =>	a group of 16 bits, 8 nibbles or 2 bytes
Double Word => (long word)	a group of 32 bits, 16 nibbles, 4 bytes or 2 words
Quad-Word => (very long word)	64bits, 32 nibbles, 8 bytes, 4 words or 2 long words

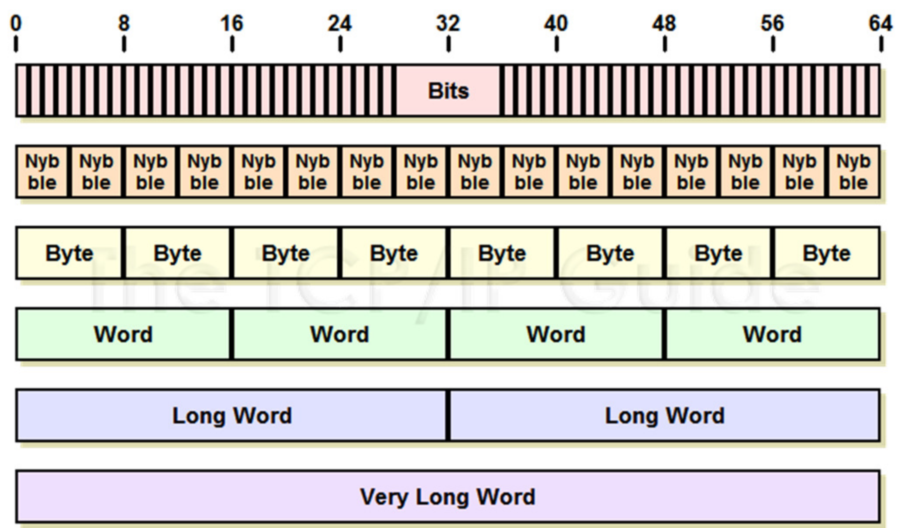


Diagram courtesy of Tcpipguide.com

- If we can read binary to decimal, we can do the opposite

741_{10} = Binary?

Approach 1) Figure out where to place the 1's

MSB –
Most Significant Bit

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	1	1	0	0	1	0	1
1024	512	256	128	64	32	16	8	4	2	1

LSB –
Least Significant Bit

Recheck:

$512 + 128 + 64 + 32 + 4 + 1 = 741_{10}$

Therefore:
Binary 1011100101_2 = Decimal 741_{10}

- Take the same decimal value: 7 4 1 $_{10}$ = Binary?

Approach 2) Through Division

2 / 741		
2 / 370	$R = 1$	LSB
2 / 185	$R = 0$	
2 / 92	$R = 1$	
2 / 46	$R = 0$	
2 / 23	$R = 0$	
2 / 11	$R = 1$	
2 / 5	$R = 1$	
2 / 2	$R = 1$	
2 / 1	$R = 0$	
0	$R = 1$	MSB

Therefore:
Binary 1011100101_2 = Decimal 741_{10}

BASE8

- Octal numbers

0, 1, 2, 3, 4, 5, 6, 7 → zero through seven

74263_8

3×8^0 $8^0 = 1$

6×8^1 $8^1 = 8$

2×8^2 $8^2 = 64$

4×8^3 $8^3 = 512$

7×8^4 $8^4 = 4096$

Octal value

Octal factor

Exponent: significant position

To Decimal:

$(3x1)+(6x8)+(2x64)+(4x512)+(7x4096) = 30899_{10}$

Therefore:
Octal 74263_8 = Decimal 30899_{10}

BASE8

- Conversion Base10 to Base8

741_{10} = Octal?

Approach: Division by octal

$8^0 = 1$ --smaller than 741

$8^1 = 8$ --smaller than 741

$8^2 = 64$ --smaller than 741

$8^3 = 512$ --smaller than 741

$8^4 = 4096$ --larger than 741

..then use 8^3 to start division

$512 \,/\, 741 \rightarrow 1 \quad R = 229$

$64 \,/\, 229 \rightarrow 3 \quad R = 37$

$8 \,/\, 37 \rightarrow 4 \quad R = 5$

$1 \,/\, 5 \rightarrow 5 \quad R = 0$

Therefore:
Decimal 741_{10} = Octal 1345_8

BASE8

- Conversion Base8 to Base2 & vice versa
- Review: 8 (octal:0-7) in binary = 2^3
- Meaning: it takes a group of 3 bits to represent 8 digits

  i.e. $7_{10} = 111_2$
- Ergo:

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

- Conversion: Binary -> Octal

100101011110001001_2 -> Base8?

Since octal digits are represented in groups of 3-bits…

1 0 0 1 0 1 0 1 1 1 1 0 0 0 1 0 0 1

4 5 3 6 1 1

Group the binary string into groups of 3, starting with the LSB

Each 3-bit group represents its octal value

Therefore:
Binary 100101011110001001_2 = Octal 453611_8

Works on both directions

BASE16

- Hexadecimal (16) numbers

0, 1, 2, 3, 4, 5, 6, 7, 8, 9 → zero through nine &

A, B, C, D, E, F → A through F… (10 – 15)

$7\ E\ 9\ 6\ A_H$

10×16^0 $16^0 = 1$

6×16^1 $16^1 = 16$

9×16^2 $16^2 = 256$

14×16^3 $16^3 = 4096$

7×16^4 $16^4 = 65536$

Hex value

Hex factor

Exponent: significant position

To Decimal:

$(10x1)+(6x16)+(9x256)+(14x4096)+(7x65536) = 518506_{10}$

Therefore:
Hex $7E96A_H$ = Decimal 518506_{10}

BASE16

- Conversion Base16 to Base2 & vice versa
- Review: 16 (hex:0-F) in binary = 2^4
- Meaning: it takes a group of 4 bits to represent 16 digits

  i.e. $15_{10} = 1111_2$
- Ergo:

Hex	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Hex	Binary
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

- Conversion Base10 to Base16

5 0 2 8 7 $_{10}$ = Hex?

Approach: Division by hex

$16^0 = 1$ --smaller than 50287

$16^1 = 16$ --smaller than 50287

$16^2 = 256$ --smaller than 50287

$16^3 = 4096$ --smaller than 50287

$16^4 = 65536$ --larger than 50287

..then use 16^3 to start division

4096 / 50287

$C <- 12$ $R = 1135$

256 / 1135

4 $R = 111$

16 / 111

6 $R = 15$

1 / 15

$F < -15$ $R = 0$

Therefore:
Decimal 50287$_{10}$ = Hex C46F$_H$

BASE16

- Conversion: Binary -> Hex

100101011110001001_2 -> Base16?

Since hex digits are represented in groups of 4-bits…

1 0 | 0 1 0 1 | 0 1 1 1 | 1 0 0 0 | 1 0 0 1

2 | 5 | 7 | 8 | 9

Group the binary string into groups of 4, starting with the LSB

Each 4-bit group represents its hex value

Therefore:
Binary 100101011110001001_2 = Hex 25789_H

Works on both directions

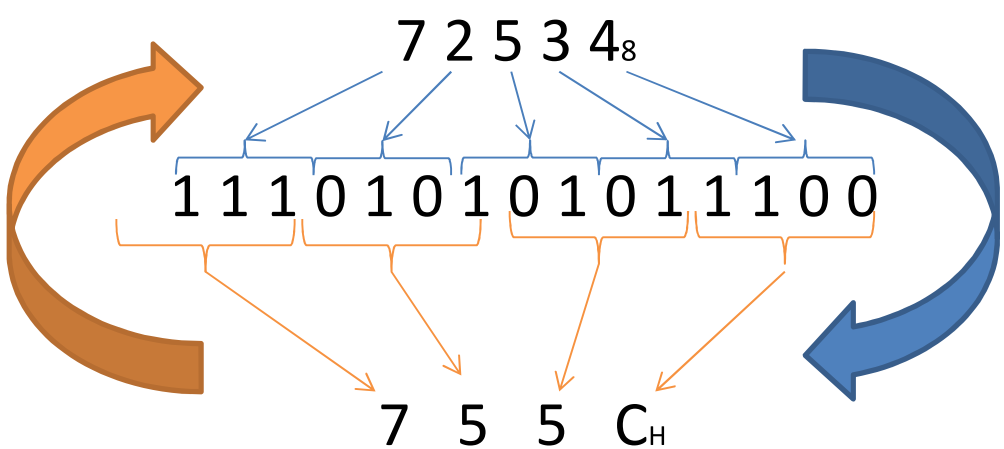
BASE16

- Conversion: Octal -> Hex

72534₈ -> Base16?

Since hex digits are represented in groups of 4-bits & octal in 3-bits, we can expect a "smaller" number for hex representation

1. Conversation from hex to octal works too going the opposite way



1. Octal is converted to binary
2. Binary value is converted to hex number