Digital Systems Fundamentals

CSE 232 Digital Systems Fundamentals

Lecture 2

Binary Number Revision and new concepts

Numerical System Review

$$(56.32)_{10} = 5*10^{1} + 6*10^{0} + 3*10^{-1} + 2*10^{-2}$$

$$(34)_{8} = 3*8^{1} + 4*8^{0}$$

$$(E1)_{16} = 14*16^{1} + 1*16^{0}$$

$$(010.11)_{2} = 0*2^{2} + 1*2^{1} + 0*2^{0} + 1*2^{-1} + 1*2^{-2}$$

Numbers in the Decimal System starts from 0 to 9 NOT from 1 to 10 (which represents 2 digits)

Binary 0 to 1, Octal 0 to 7, Hexadecimal 0 to F(15)

Decimal Binary Equivalence

Decimal	Binary	
0	0	0
1	0	1
2	1	0
3	1	1

Decimal	Binary		
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Conversion

$$(34)_{10} = (?)_2$$

$$34/2 = 17 + 0 \qquad a_0 = 0$$

$$17/2 = 8 + 1/2 \qquad a_1 = 1$$

$$8/2 = 4 + 0 \qquad a_2 = 0$$

$$4/2 = 2 + 0 \qquad a_3 = 0$$

$$2/2 = 1 + 0 \qquad a_4 = 0$$

$$1/2 = 1/2 \qquad a_5 = 1$$

$$(34)_{10} = (a_5 a_4 a_3 a_2 a_1 a_0)_2$$

MSB LSB

Class objective

- What is a Digital Signal and how it is represented
- What type of signal that all IT devices has
- What are Bits, Bytes, words, other data types, strings, characters
- Why we need Binary representation
- Why we need Hexadecimal representation

Digital Signals

- The amplitude of a digital signal varies between a logical “0” and logical “1”.

All IT devices uses digital signals, why

- Easier to send signals
- Less error in transmission
- Less error
- Better in design

All memory in IT devices are represented in Bytes,

16 GB=16 Gegabyte = 16x 10^9 bytes

16 MB=16 Megabytes = 16x 10^6 bytes

A bit

- A single piece of digital information
  - Either a logical "0" or a logical "1"
    - "1101" is a 4 bit number
    - Since digital electronic circuits output voltages not bits, we assign a voltage range to be equal to a logical "0" and a different voltage range to be a logical "1"
      - Different logic families assign a different range of voltages to be equal to a logical "0" and a logical "1".
      - For example:
        - » TTL "0" $\rightarrow$ 0 – 0.7V; "1" $\rightarrow$ 2 – 5V
        - » CMOS "0" $\rightarrow$ 0 – 1.5V; "1" $\rightarrow$ 3.5 – 5V

Converting Decimal to Binary

5	2	
2	1	LSB
1	0	
0	1	MSB

101

14	2	
7	0	LSB
3	1	
1	1	
0	1	MSB

1110

23	2	
11	1	LSB
5	1	
2	1	
1	0	
0	1	MSB

10111

Grouping of Bits

- Byte: Composed of 8 bits
- String: Sequential set of bits of arbitrary length
- Nibble: Composed of 4 bits or half a byte
  - Nibbles are occasionally written as hexadecimals to make the data more readable
  - Hexadecimals are the numbers in base 16
    - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Word: The standard memory bus width in the microprocessor or computer architecture
  - 16-bit, 32-bit, or 64-bit architecture

Why we need Binary System

All signals in digital IT devices are represented in Binary, just zeros and ones

All input output signals are converted to digital
All communications/transmission are in digital form by cables

Why we need Hexadecimal

An easy way of reading a binary number

Representation is better memory wise

All data is READABLE

Can you read this
1101 0110 0111 0111 0000 or D6770

A Trick:
Converting Binary to Octal

- Mark groups of *three* (from right)
- Convert each group

10101011	10	101	011
	2	5	3

10101011 is 253 in base 8

Why does this work??

Converting Binary to Hexadecimal

- Mark groups of *four* (from right)
- Convert each group

10101011 1010 1011

A B

10101011 is AB in base 16

And this?

Converting Decimal to Octal

27	8	
3	3	LSB
0	3	MSB

33

55	8	
6	7	LSB
0	6	MSB

67

19	8	
2	3	LSB
0	2	MSB

23

Converting Decimal to Hex

29	16	LSB
1	13=D	
0	1	MSB

1D

40	16	LSB
2	8	
0	2	MSB

28

1399	16	LSB
87	7	
5	7	
0	5	MSB

577

Strings, Characters

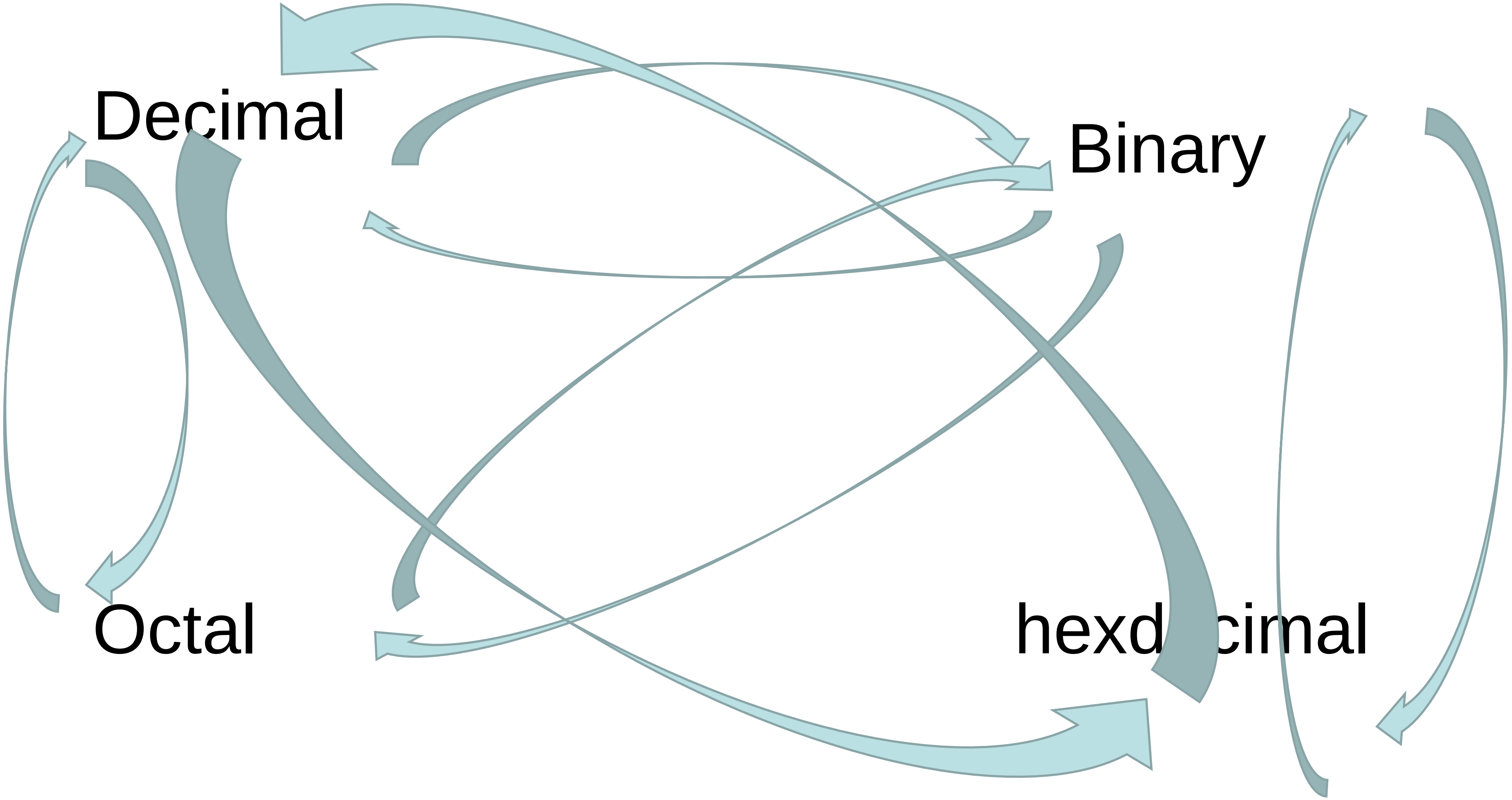
Each character has its own code in any computer or IT device, known as ASCII code
For example
‘A’=32
‘X’=55

Strings are concatenated characters, all a few digitals in HEX according to size of string
“This is the first string”=ADBD3F46789…..
and so on

How to convert Hex or Octal to Binary

CONVERSIONS			
Decimal	Binary	Octal	Hex
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	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
17	10001	21	12
18	10010	22	13

Remember



Decimal Binary Equivalence

Decimal	Binary	
0	0	0
1	0	1
2	1	0
3	1	1

This is BCD
Binary Coded
Decimal

Decimal	Binary		
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

1’s Complement and 2’s Complement

An easier implementation for negative numbers in the subtraction process

1’s complement of C(N)=2^n - N -1

1’s complement of C(N)=(1000-1)-110=001

2’s complement of C(N)=1’s complement + 1

=010

Subtraction with 2's complement

$K + C(N) = K - N + 2^n$

If K > N, then you get a carry that you can discard

If K < N, then you get the 2's complement of the result

Example

X=1001 Y=0110

X-Y=1001+1010

=1 0011

X=0110 Y=1010

X-Y=0110+0110

= 1100 take its 2's

= 0100

Codes

Binary Coded Decimals (BCD):

Each decimal digits 4 binary digits

ASCII
Each character get 7 binary digits

Gray Coded Numbers:
1 bit change per one increment

Home work

Problems:

Number 8, 16, 18, 21 due next week in tutorial