CSE 100: Computer Skills

Lecture 6: Digital Systems
-Number Systems

Shahadat Hussain Parvez

TRANSFORMING DATA INTO INFORMATION

How Computers Represent Data

- Number systems
  - A manner of counting
  - Several different number systems exist
- Decimal number system
  - Used by humans to count
  - Contains ten distinct digits
  - Digits combine to make larger numbers

How Computers Represent Data

- Binary number system
  - Used by computers to count
  - Two distinct digits, 0 and 1
  - 0 and 1 combine to make numbers

Number	Binary
Zero	0
One	1
Two	10
Three	11
Four	100

How Computers Represent Data

- Octal number system
  - Eight distinct digits, 0 to 7
  - Digits combine to make numbers

Number	Octal
Zero	0
One	1
Two	2
Three	3
Four	4
Five	5
Six	6
Seven	7
Eight	10
Nine	11
Ten	12

How Computers Represent Data

- Hexadecimal number system
  - 16 distinct digits, 0 to 9, A, B, C, D, E, F
  - Digits combine to make numbers
  - A equivalent to 10 in Decimal
  - B equivalent to 11 in Decimal
  - C equivalent to 12 in Decimal
  - D equivalent to 13 in Decimal
  - E equivalent to 14 in Decimal
  - F equivalent to 15 in Decimal

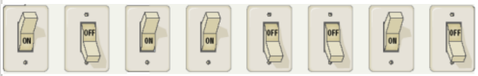
How Computers Represent Data

- Hexadecimal number system

Number	Hexadecimal
Zero	0
One	1
Two	2
Three	3
Four	4
Five	5
Six	6
Seven	7
Eight	8
Nine	9
Ten	A
Eleven	B
Twelve	C
Thirteen	D
Fourteen	E
Fifteen	F
Sixteen	10

How Computers Represent Data

- Bits and bytes
  - Binary numbers are made of bits
  - Bit represents a switch
  - A byte is 8 bits
  - Byte represents one character



CONVERSION BETWEEN NUMBER SYSTEMS

Conversion from Decimal

- To convert numbers from decimal to any other bases use the following steps.
  - **Step 1** – Divide the decimal number to be converted by the value of the new base.
  - **Step 2** – Get the remainder from Step 1 as the rightmost digit (least significant digit) of new base number.
  - **Step 3** – Divide the quotient of the previous divide by the new base.
  - **Step 4** – Record the remainder from Step 3 as the next digit (to the left) of the new base number.
- Repeat Steps 3 and 4, getting remainders from right to left, until the quotient becomes zero in Step 3.
- The last remainder thus obtained will be the Most Significant Digit (MSD) of the new base number

Conversion from Decimal

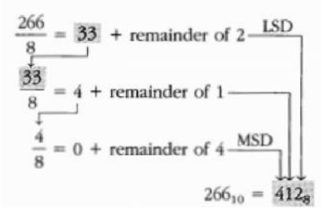
Example 1 – Decimal Number: 29_{10} Calculating Binary Equivalent

Step	Operation	Result	Remainder
Step 1	29 / 2	14	1
Step 2	14 / 2	7	0
Step 3	7 / 2	3	1
Step 4	3 / 2	1	1
Step 5	1 / 2	0	1

As mentioned in Steps 2 and 4, the remainders have to be arranged in the reverse order so that the first remainder becomes the Least Significant Digit (LSD) and the last remainder becomes the Most Significant Digit (MSD).
Decimal Number − 29_{10} = Binary Number − 11101_2.

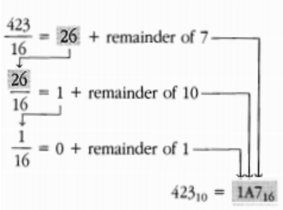
Conversion from Decimal

- Decimal Number: 266 Calculating Octal Equivalent



Conversion from Decimal

- Decimal Number: 423 Calculating Hexadecimal Equivalent



$\frac{423}{16} = \boxed{26}$ + remainder of 7

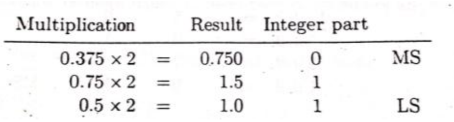
$\frac{\boxed{26}}{16} = 1$ + remainder of 10

$\frac{1}{16} = 0$ + remainder of 1

$423_{10} = \boxed{1A7_{16}}$

Conversion from Decimal

- Decimal Number: 0.375 Calculating Binary Equivalent

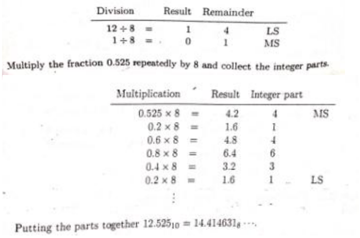


Multiplication		Result	Integer part	
0.375×2	=	0.750	0	MS
0.75×2	=	1.5	1	
0.5×2	=	1.0	1	LS

- Binary Equivalent is 0.011_2

Conversion from Decimal

- Decimal Number: 12.525 Calculating Octal Equivalent



Division		Result	Remainder	
$12 \div 8$	=	1	4	LS
$1 \div 8$	=	0	1	MS

Multiply the fraction 0.525 repeatedly by 8 and collect the integer parts.

Multiplication		Result	Integer part	
0.525×8	=	4.2	4	MS
0.2×8	=	1.6	1	
0.6×8	=	4.8	4	
0.8×8	=	6.4	6	
0.4×8	=	3.2	3	
0.2×8	=	1.6	1	LS
⋮				

Putting the parts together $12.525_{10} = 14.414631_8 \cdots$.

Conversion to Decimal

- To convert number from any other base to decimal use the following steps
  - **Step 1** – Determine the column (positional) value of each digit (this depends on the position of the digit and the base of the number system).
  - **Step 2** – Multiply the obtained column values (in Step 1) by the digits in the corresponding columns.
  - **Step 3** – Sum the products calculated in Step 2. The total is the equivalent value in decimal.

Conversion to Decimal

- Binary Number – 11101_2 Calculating Decimal Equivalent

Step	Binary Number	Decimal Number
Step 1	11101_2	$((1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0))_{10}$
Step 2	11101_2	$(16 + 8 + 4 + 0 + 1)_{10}$
Step 3	11101_2	29_{10}

Binary Number – 11101_2 = Decimal Number – 29_{10}

Conversion to Decimal

- Octal Number – 7465_8 Calculating Decimal Equivalent

Step	Octal	Decimal Number
Step 1	7465_8	$((7 \times 8^3) + (4 \times 8^2) + (6 \times 8^1) + (5 \times 8^0))_{10}$
Step 2	7465_8	$(3584 + 256 + 48 + 5)_{10}$
Step 3	7465_8	3893_{10}

Binary Number – 7465_8 = Decimal Number – 3893_{10}

Conversion to Decimal

- Hexadecimal number − $F2B_{16}$ Calculating Decimal Equivalent

$$\begin{array}{l} \quad F \qquad\quad 2 \qquad\quad B \\ = FX16^2 + 2X16^1 + BX16^0 \\ =15X256 + 2X16 + 11X1 \\ = 3840 + 32 + 11 \\ =3883_{10} \end{array}$$

Conversion to Decimal

- Binary number − 0.111_2 Calculating Decimal Equivalent

$$\begin{array}{cccc} & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ 0. & 1 & 1 & 1 \end{array}$$

$$\text{Hence, } \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = 0.5 + 0.25 + 0.125 = 0.875_{10}$$

Conversion to Decimal

- Octal number − 32.4_8 Calculating Decimal Equivalent

$$\begin{array}{cccc} 8^1 & 8^0 & & 8^{-1} \\ 3 & 2 & . & 4 \end{array}$$

$$\begin{aligned} &= 3 \times 8^1 + 2 \times 8^0 + 4 \times 8^{-1} \\ &= 3 \times 8 + 2 \times 1 + 4 \times \frac{1}{8} \\ &= 24 + 2 + 0.5 \\ &= 26.5_{10} \end{aligned}$$

Conversion within Binary and Octal

- The following table is used for inter-conversion between binary and Octal

Table 2.3: Binary and octal conversion table.

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

Conversion from Binary to Octal

- **Step 1** − Divide the binary digits into groups of three (starting from the right).
- **Step 2** − Convert each group of three binary digits to one octal digit.

Example Binary Number − 10101_2 Calculating Octal Equivalent

Step	Binary Number	Octal Number
Step 1	10101_2	010 101
Step 2	10101_2	$2_8\ 5_8$
Step 3	10101_2	25_8

Binary Number − 10101_2 = Octal Number − 25_8

Conversion from Octal to Binary

- **Step 1** − Convert each octal digit to a 3 digit binary number (the octal digits may be treated as decimal for this conversion).
- **Step 2** − Combine all the resulting binary groups (of 3 digits each) into a single binary number.

Example Octal Number − 25_8 Calculating Binary Equivalent

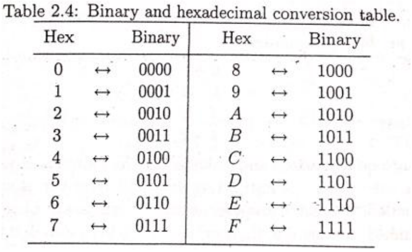
Step	Octal Number	Binary Number
Step 1	25_8	$2_{10}\ 5_{10}$
Step 2	25_8	$010_2\ 101_2$
Step 3	25_8	010101_2

Octal Number − 25_8 = Binary Number − 10101_2

Conversion within Binary and Hexadecimal

- The following table is used for inter-conversion between binary and hexadecimal

Table 2.4: Binary and hexadecimal conversion table.



Hex		Binary	Hex		Binary
0	↔	0000	8	↔	1000
1	↔	0001	9	↔	1001
2	↔	0010	A	↔	1010
3	↔	0011	B	↔	1011
4	↔	0100	C	↔	1100
5	↔	0101	D	↔	1101
6	↔	0110	E	↔	1110
7	↔	0111	F	↔	1111

Conversion From Binary To Hexadecimal

- **Step 1** − Divide the binary digits into groups of four (starting from the right).
- **Step 2** − Convert each group of four binary digits to one hexadecimal symbol.

Example Binary Number – 10101_2 Calculating hexadecimal Equivalent

Step	**Binary Number**	**Hexadecimal Number**
Step 1	10101_2	0001 0101
Step 2	10101_2	1_{10} 5_{10}
Step 3	10101_2	15_{16}

Binary Number – 10101_2 = Hexadecimal Number – 15_{16}

Conversion From Hexadecimal To Binary

- **Step 1** − Convert each hexadecimal digit to a 4 digit binary number (the hexadecimal digits may be treated as decimal for this conversion).
- **Step 2** − Combine all the resulting binary groups (of 4 digits each) into a single binary number.

Example Hexadecimal Number – 15_{16} Calculating Binary Equivalent

Step	**Hexadecimal Number**	**Binary Number**
Step 1	15_{16}	1_{10} 5_{10}
Step 2	15_{16}	0001_2 0101_2
Step 3	15_{16}	00010101_2

Hexadecimal Number – 15_{16} = Binary Number – 10101_2

BINARY ADDITION AND SUBTRACTION

Binary Addition

- Binary addition can be done according to the simple rules

0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = 10
1 + 1 + 1 = 11

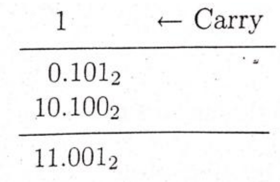
Binary Addition

- Similar to decimal addition, when in one column is a two digit number the least significant figure is written as part of the total sum and the most significant digit is carried forward.

```
   11   1 <--- Carry bits -----> 11
  1001001                      1000111
+ 0011001                    + 0010110
---------                    ---------
  1100010                      1011101
```

Binary Addition

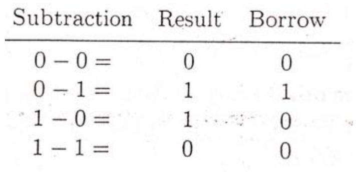
- Add 0.101_2 with 10.1_2



$$\begin{array}{ll} 1 & \leftarrow \text{Carry} \\ \hline 0.101_2 & \\ 10.100_2 & \\ \hline 11.001_2 & \end{array}$$

Binary Subtraction

- Binary Subtraction has rules similar to that of addition. They are



Subtraction	Result	Borrow
$0-0=$	0	0
$0-1=$	1	1
$1-0=$	1	0
$1-1=$	0	0

Binary Complements

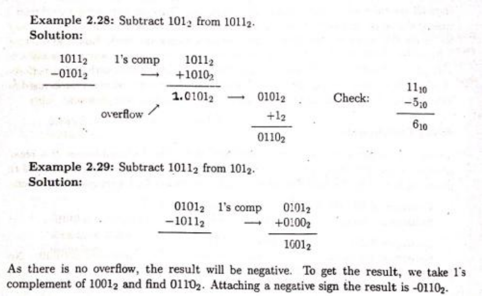
- There are two kinds of complements in binary
  - 1's complement
    - 1'S complement is found by changing the 1's to 0's and 0's to 1's
    - 10101 --- 01010
  - 2's complement
    - 2's complement of a number is 1 added to the 1's complement of that number
    - 11001 --- 00110+1 --- 00111

Subtraction using 1's complement

- The rules are as follows

1. Add extra zeroes to make the length of both numbers same
2. Take 1's complement of the number to be subtracted
3. Now add the complement to the other one
4. If the sum has an overflow (Length of the sum is more than the length of numbers) the result is positive. To get the result, add this overflow at the rightmost position of sum
5. If there is no overflow, the result is negative. To get the result, take 1's complement of the sum again, and attach a negative sign

Subtraction using 1's complement



Example 2.28: Subtract 101_2 from 1011_2.
Solution:

$$\begin{array}{r} 1011_2 \\ -0101_2 \\ \hline \end{array} \xrightarrow{\text{1's comp}} \begin{array}{r} 1011_2 \\ +1010_2 \\ \hline 1.0101_2 \end{array} \longrightarrow \begin{array}{r} 0101_2 \\ +1_2 \\ \hline 0110_2 \end{array} \qquad \text{Check:} \quad \begin{array}{r} 11_{10} \\ -5_{10} \\ \hline 6_{10} \end{array}$$

overflow ↗

Example 2.29: Subtract 1011_2 from 101_2.
Solution:

$$\begin{array}{r} 0101_2 \\ -1011_2 \\ \hline \end{array} \xrightarrow{\text{1's comp}} \begin{array}{r} 0101_2 \\ +0100_2 \\ \hline 1001_2 \end{array}$$

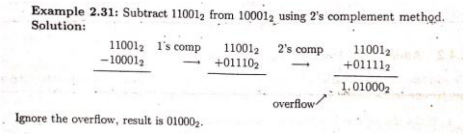
As there is no overflow, the result will be negative. To get the result, we take 1's complement of 1001_2 and find 0110_2. Attaching a negative sign the result is -0110_2.

Subtraction using 2's complement

- The rules are as follows

1. Add extra zeroes to make the length of both numbers same
2. Take 2's complement of the number to be subtracted
3. Now add the complement to the other one
4. If the sum has an overflow (Length of the sum is more than the length of numbers) the result is positive. To get the result, discard the overflow
5. If there is no overflow, the result is negative. To get the result, take 1's complement of the sum again, and attach a negative sign

Subtraction using 2’s complement



Example 2.31: Subtract 11001_2 from 10001_2 using 2's complement method.
Solution:

$$\begin{array}{r} 11001_2 \\ -10001_2 \\ \hline \end{array} \xrightarrow{\text{1's comp}} \begin{array}{r} 11001_2 \\ +01110_2 \\ \hline \end{array} \xrightarrow{\text{2's comp}} \begin{array}{r} 11001_2 \\ +01111_2 \\ \hline 1.01000_2 \end{array}$$

overflow

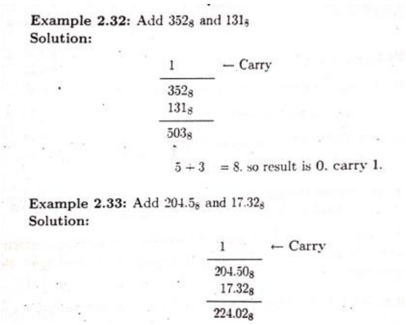
Ignore the overflow, result is 01000_2.

Addition in Octal

- The rules to add 2 octal numbers are as follow

1. If the sum is less than 8, that is the result and carry is nothing
2. If the sum is exactly 8, take 0 as a result and carry is 1
3. If the sum is larger than 8, the difference is the result and carry is 1

Addition in Octal



Example 2.32: Add 352_8 and 131_8
Solution:

$$\begin{array}{rl} 1 & \text{— Carry} \\ \hline 352_8 & \\ 131_8 & \\ \hline 503_8 & \end{array}$$

$5 + 3 = 8$. so result is 0. carry 1.

Example 2.33: Add 204.5_8 and 17.32_8
Solution:

$$\begin{array}{rl} 1 & \leftarrow \text{Carry} \\ \hline 204.50_8 & \\ 17.32_8 & \\ \hline 224.02_8 & \end{array}$$

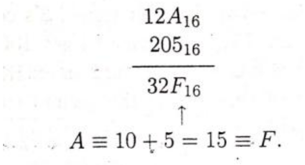
Addition in Hexadecimal

- The rules to add 2 hexadecimal numbers are as follow

1. If the sum is less than 16, that is the result and carry is nothing
2. If the sum is exactly 16, take 0 as a result and carry is 1
3. If the sum is larger than 16, the difference is the result and carry is 1

Addition in Hexadecimal

- Add $12A_{16}$ and 205_{16}



$$\begin{array}{r} 12A_{16} \\ 205_{16} \\ \hline 32F_{16} \\ \uparrow \end{array}$$

$$A \equiv 10 + 5 = 15 \equiv F.$$

Signed Numbers

- Negative numbers denoted by placing a minus sign
- Difficult in computer system
- We use a physical bit to represent the sign
- Depending on the sign bit the number is either positive or negative
- The leftmost bit is usually the sign bit
- When sign bit is used, the total length of a binary number including the sign bit is significant
- Signed numbers are usually stored in 8, 16, or 32 bits or so on

Signed Numbers

- Using sign bit the positive numbers are stored as it is
- The negative number is stored as complement
- Depending on complement method, signed numbers can be signed 1's complement or signed 2's complement
- 2's complement is most common in computer

Signed Numbers

- In signed 2's complement
  - Sign bit is 0 for positive numbers
  - Sign bit is 1 for negative numbers
    - The whole number represents the value in 2's complement
    - To find the value we calculate the 2's complement

Signed Numbers (examples)

- If 00101101_2 is in signed 2's complement form what is the decimal equivalent

		32		8	4		1
0	0	1	0	1	1	0	1

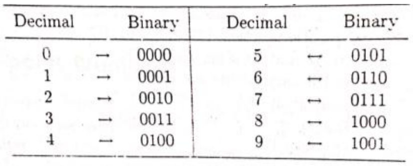
  - The represented value is $32+8+4+1=45_{10}$
- If 10101101_2 is in signed 2's complement form what is the decimal equivalent
  - First find 2's complement and find the value

	64		16			2	1
0	1	0	1	0	0	1	1

  - The represented value is $64+16+2+1=-83_{10}$

How Computers Represent Data

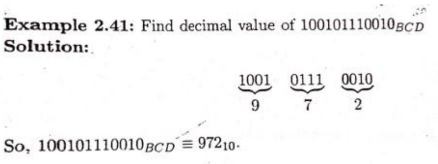
- BCD (Binary Coded Decimal)
  - Represents decimal numbers
  - Uses 4-bit code
  - Follows the following table



Decimal		Binary	Decimal		Binary
0	→	0000	5	→	0101
1	→	0001	6	→	0110
2	→	0010	7	→	0111
3	→	0011	8	→	1000
4	→	0100	9	→	1001

How Computers Represent Data

- BCD (Binary Coded Decimal)



Example 2.41: Find decimal value of 100101110010_{BCD}

Solution:

$$\underbrace{1001}_{9}\quad\underbrace{0111}_{7}\quad\underbrace{0010}_{2}$$

So, $100101110010_{BCD} \equiv 972_{10}$.

How Computers Represent Data

- Text codes
  - Converts letters into binary
  - Standard codes necessary for data transfer
  - ASCII (American Standard Code for Information Interchange)
    - American English symbols
    - 7 bit code
  - Extended ASCII
    - Graphics and other symbols
  - Unicode
    - 16 bit code
    - All languages on the planet

Homework Based on Lecture 6

- Questions from Chapter 2, Introduction to computers by Mohammed Alamgir
- Example 2.1-2.36
- Exercise 2, 4, 5, 6, 7, 8, 9, 10, 14, 15, 19, 20, 23, 24, 28, 29, 36

END