Digital Logic and Design (EEE-241)

Lecture 2

Dr. M. G. Abbas Malik

abbas.malik@ciitlahore.edu.pk

Picture Source: http://www.vanoast.com/old-portfolio/digital-design-logo-one%5Ba%5D.jpg

Previous lecture

- Digital Vs. Analog
- Digital System – example digital computer
- Discrete values and binary values
- Number system (positional and non-positional)
- Binary number system
  - Conversion, addition, subtraction, multiplication, division.
- Octal number system (conversion)
- Hexadecimal number system (conversion)

Complements

- Complements are used to simplify the subtraction operation and for logical manipulation
- Consider r is the base of a number system, then there exist two types of complement
  1. **r‘s complement**
  2. **(r-1)’s complement**

Example

- For binary number system, there are two complements 2’s (r) complement and 1’s (r-1) complement
- For decimal number system, these are 10’s and 9’s complements.

Complements

r's complements

r is the base of our number system.

X is a number in base r that has n digits in its integral part.

The r's complement of X is defined as $r^n - X$ for $X \neq 0$ and 0 for $X=0$.

Example 1: Decimal numbers

$\boxed{1234}.\boxed{765}$

Integral Part

Fractional Part

Decimal Number:
10's (r's) complement
$= 10^4 - 1234.765$
$= 8765.235$

Complements

r's complements

Example 1: Decimal numbers (cont…)

- The 10's complement of $(0.3456)_{10}$ is $10^0 - 0.3456 = 1 - 0.3456 = 0.6544$
- 10's complement can formed by leaving all least significant ZEROS unchanged, subtracting the first non-zero least significant digit from 10, and then subtracting all other higher significant digits from 9.

Complements

r's complements

Example 2: Binary numbers

- The 2's (r's) complement of $(101100)_2$ is $(2^6)_{10} – (101100)_2 = (1000000 - 101100)_2 = (010100)_2$
- The 2's complement of $(0.0110)_2$ is $(2^0)_{10} – (0.0110)_2 = (1 – 0.0110)_2 = 0.1010$
- The 2's complement can be computed by leaving all least significant ZEROS and first non-zero digit unchanged, and then replacing 1's by 0's and 0's by 1's in all other higher significant digits.

Complements

(r-1)'s complements

r is the base of our number system.

X is a number in base r that has n digits in its integral part and a fractional part with m digits.

The (r-1)'s complement of X is defined as $r^n - r^{-m} - X$

Example 1: Decimal numbers

$$\underbrace{\boxed{1234}}_{\text{Integral Part}}.\underbrace{\boxed{765}}_{\text{Fractional Part}}$$

Decimal Number:
9's (r-1's) complement
$= 10^4 - 10^{-3} - 1234.765$
$= 10000 - 0.001 - 1234.765$
$= 8765.234$

Complements

(r-1)'s complements

Example 1: Decimal numbers (cont…)

- The 9's complement of $(0.3456)_{10}$ is $10^0 – 10^{-4} – 0.3456 = 1 – 0.0001 - 0.3456$

  $= 0.6543$
- 9's complement of a decimal number is formed by subtracting every digit by 9.

Complements

(r-1)'s complements

Example 2: Binary numbers

- The 1's complement of $(101100)_2$ is $(2^6 - 2^0)_{10} - (101100)_2 = (1000000 - 1 - 101100)_2$ $= (111111 - 101100)_2 = (010011)_2$
- The 1's complement of $(0.0110)_2$ is $(2^0 - 2^{-4})_{10} - (0.0110)_2 = (1 - 0.0001 - 0.0110)_2$ $= (0.1111 - 0.0110)_2 = (0.1001)_2$
- The 1's complement is formed by replacing 1's by 0's and 0's by 1's.

Complements

Conversions

- r’s complement to (r-1)’s complement

r‘s complement – r^{-m}

- (r-1)’s complement to r’s complement

(r-1)‘s complement + r^{-m}

Subtraction

General Subtraction

$$
\begin{array}{rl}
246 & \leftarrow \textbf{Minuend} \\
-\ 138 & \leftarrow \textbf{Subtrahend} \\
\text{-----} & \\
108 &
\end{array}
$$

- In general, when the minuend digit is smaller than the corresponding subtrahend digit, we borrow 1 from the higher significant position.
- This is easier when people perform subtraction with paper and pencil.
- This method is less efficient in digital systems.

Subtraction with Complements

With r's Complement

Suppose we have two positive number X and Y, both of base r.

Subtraction X – Y may be done as follows:

1. Add the minuend X to the r's compliment of the subtrahend Y.
2. Inspect the result obtained in step 1 for an end carry
   a) If an end carry occurs, discard the end carry.
   b) If an end carry does not occur, take the r's complement of the number obtained in step 1 and place a negative sign in front.

Subtraction with Complements

With r's Complement

Example 1 – Decimal numbers

X = 72532

Y = 3250

10' complement of 03250 =

$$
\begin{array}{rr}
 & 72532 \\
+ & 96750 \\
\hline
\text{End carry} \rightarrow 1 & 69282
\end{array}
$$

Answer of X – Y = 69282

Subtraction with Complements

With r's Complement

Example 1 – Decimal numbers

X = 3250 03250

Y = 72532

10' complement of 72532 = + 27468

No end carry 30718

Answer of X – Y = – 10's complement of 30718 = – 69282

Subtraction with Complements

Number of digits in X and Y must be same. Before taking the r's complement of Subtrahend (Y), if Y has less number of digits, then we add ZEROS to the left of the number and take r's complement considering newly added ZEROS as part of the number.

Subtraction with Complements

With r's Complement

Example 2 – Binary numbers

X = 1010100

Y = 100100

$$
\begin{array}{lrr}
 & & 1010100 \\
\text{2's complement of } 0100100 = & + & 1011100 \\
\hline
\text{End carry} \rightarrow 1 & & 0110000
\end{array}
$$

Answer of X – Y = 110000

Subtraction with Complements

With r's Complement

Example 2 – Binary numbers

X = 100100 $\qquad\qquad$ 0100100

Y = 1010100

2's complement of 1010100 = + 0101100

No end carry $\qquad$ 1010000

Answer = – 2's complement of 1010000 = – 110000

Subtraction with Complements

With (r-1)'s complement

Suppose we have two positive number X and Y, both of base r.

Subtraction X – Y may be done as follows:

1. Add the minuend X to the (r-1)'s compliment of the subtrahend Y.
2. Inspect the result obtained in step 1 for an end carry
   a) If an end carry occurs, add 1 to the least significant digit (end around carry).
   b) If an end carry does not occur, take the (r-1)'s complement of the number obtained in step 1 and place a negative sign in front.

Subtraction with Complements

With (r-1)'s Complement

Example 1 – Decimal numbers

X = 72532 72532

Y = 3250

9's complement of 03250 = + 96749

End carry → 1 69281

\+

Answer of X – Y = 69282

Subtraction with Complements

With (r-1)'s Complement

Example 1 – Decimal numbers

X = 3250			03250
Y = 72532			
9's complement of	72532 =	+	27467
	No end carry		30717

Answer of X – Y = – 9's complement of 30718 = – 69282

Subtraction with Complements

Number of digits in X and Y must be same. Before taking the r's complement of Subtrahend (Y), if Y has less number of digits, then we add ZEROS to the left of the number and take r's complement considering newly added ZEROS as part of the number.

Subtraction with Complements

With (r-1)'s Complement

Example 2 – Binary numbers

X = 1010100   1010100

Y = 100100

1' complement of   0100100 =   + 1011011

End carry → 1   0101111

+

Answer of X – Y = 110000

Subtraction with Complements

With (r-1)'s Complement

Example 2 – Binary numbers

X = 100100			0100100
Y = 1010100			
1' complement of	1010100 =		+ 0101011
	No end carry		1001111

Answer = – 1's complement of 1010000 = – 110000

Complements in Digital Systems

2's complement

1. Harder to implement
2. Subtraction take only one arithmetic addition
3. Possesses only one arithmetic 0
4. Used only in conjunction with arithmetic applications

1's complement

1. Easier to implement
2. Subtraction takes two arithmetic additions
3. Possesses two arithmetic zeros: one with all 0s and one with all 1s
4. Useful in logical manipulation: change of 1's to 0's - inversion

Binary Codes

- Electronic digital systems use **signals** that have two distinct values and **Circuit elements** that have two stable states.
- A binary number of n-digits may be represented by n binary circuit elements, each having an output signal equivalent to 1 or 0.
- Digital system not only manipulate binary numbers, but also many other discrete elements of information.
  - Colors
  - Decimal digits
  - Alphabets (English, French, Urdu, Hindi, Punjabi, etc.)

Binary Codes

- A bit is a binary digit.
- To represent a group of 2^n distinct discrete elements in a binary code requires a minimum of n bits.

Example 1: Color Codes

- Consider we have four colors: RED, GREEN, BLACK and BLUE
- We want to assign a binary code to each color, so that we can represent these information in our digital system

Binary Codes

Example 1: Color Codes (cont…)

- We have total $4 = 2^2$ distinct discrete elements.
- It means we need minimum 2 bits to represent these four colors.

Color	Binary Code
Black	00
Blue	01
Green	10
Red	11

Binary Codes

Example 2: Binary Coded Decimal (BCD)

- Binary code for decimal digits require a minimum 4 bits as $2^4 = 16 > 10 > 8 = 2^3$

Bit required	Number of Discrete Elements
1	$2^0 \leq$ discrete elements $\leq 2^1$
2	$2^1 \leq$ discrete elements $\leq 2^2$
3	$2^2 \leq$ discrete elements $\leq 2^3$
4	$2^3 \leq$ discrete elements $\leq 2^4$
5	$2^4 \leq$ discrete elements $\leq 2^5$
8	$2^7 \leq$ discrete elements $\leq$ **256**$=2^8$

Binary Codes

Example 2: Binary Coded Decimal (BCD)

- Binary code for decimal digits require a minimum 4 bits as $2^4 = 16 > 10 > 8 = 2^3$

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

Home Work: Read & Understand the other decimal codes given in table 1-2 and binary codes: Error-Detection, Reflected & Alphanumeric Codes (**Section1-6**)

Binary Codes

Difference between Binary Codes and Binary Conversion

- When we convert a decimal number in binary, it is a sequence of 1's and 0's similar to binary codes, but they are different.

Binary Conversion

Conversion of decimal number 24 in binary is $(11000)_2$

Binary Codes

Binary code of decimal number 24 in BCS is $(0010\ 0100)_2$

Binary Storage and Registers

- Binary Cell is a device that possesses two stable states and is capable of storing one bit of information.
  - In Electronics, it is a flip-flop circuit, ferrite core use in memory
  - In a Punch-Card, positions punched with a hole or not punched.
- Binary cells are used to store the digital (binary) information
- **Register** is a group of binary cells

Binary Storage and Registers

- 1 bit = 1 binary cell
- 1 byte = 8 bits
- Register size depends on the computer system architecture.
  1. 8 bits architecture – register size is 8 bits
  2. 16 bits architecture – register size is 16 bits
  3. 32 bits architecture – register size is 32 bits
  4. 64 bits architecture (latest) – register size is 64 bits
- A digital computer is characterized by its registers.
- Assignment # 2 (available on course website)

Registers

- A register with n binary cells can stores any discrete quantity of information that contains n bits.
- This is a 16-cells register

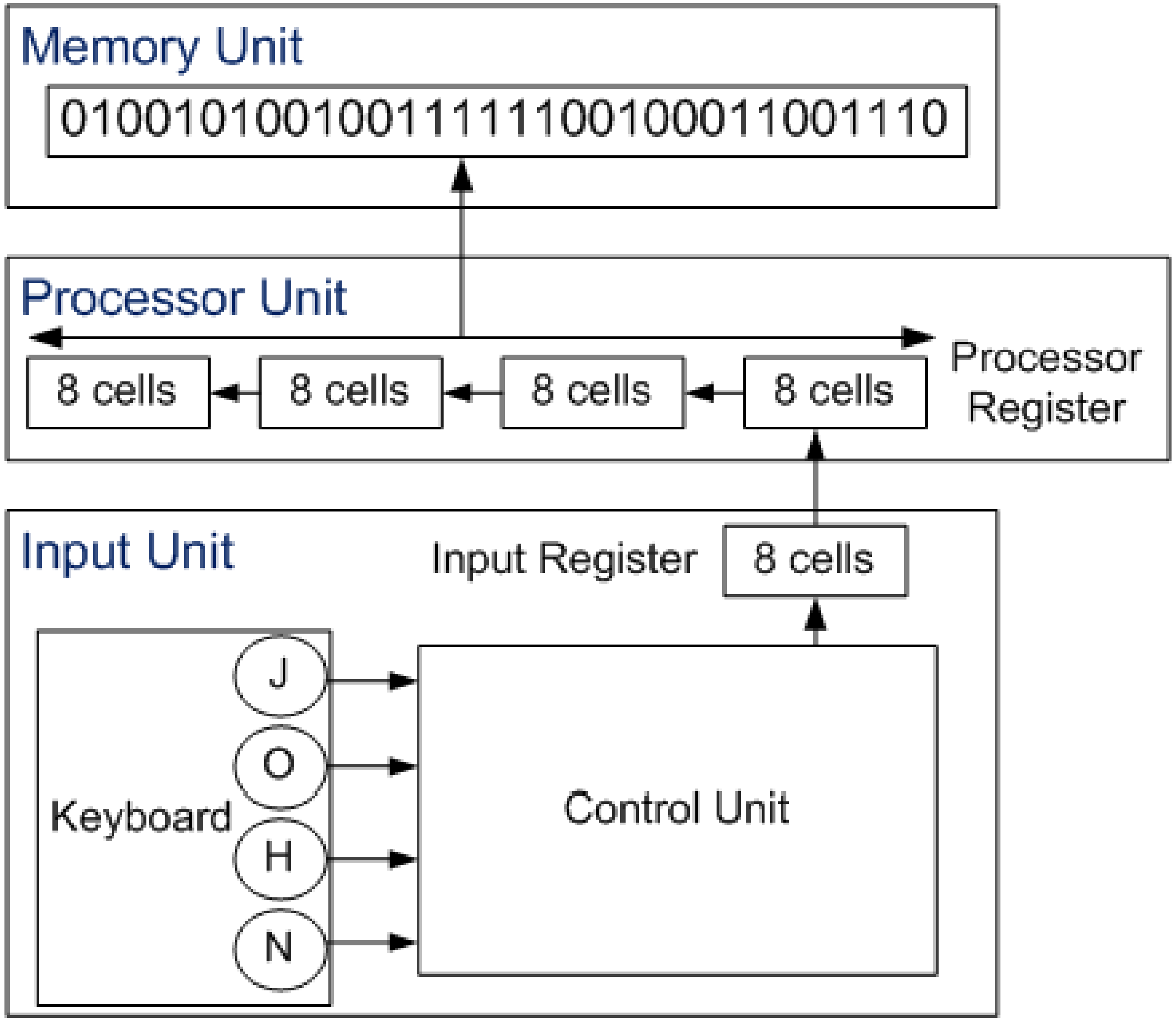
<table>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td colspan="4">8</td><td colspan="4">3</td><td colspan="4">6</td><td colspan="4">9</td></tr>
</table>

- If contains the BCD information then it represent the decimal number 8369

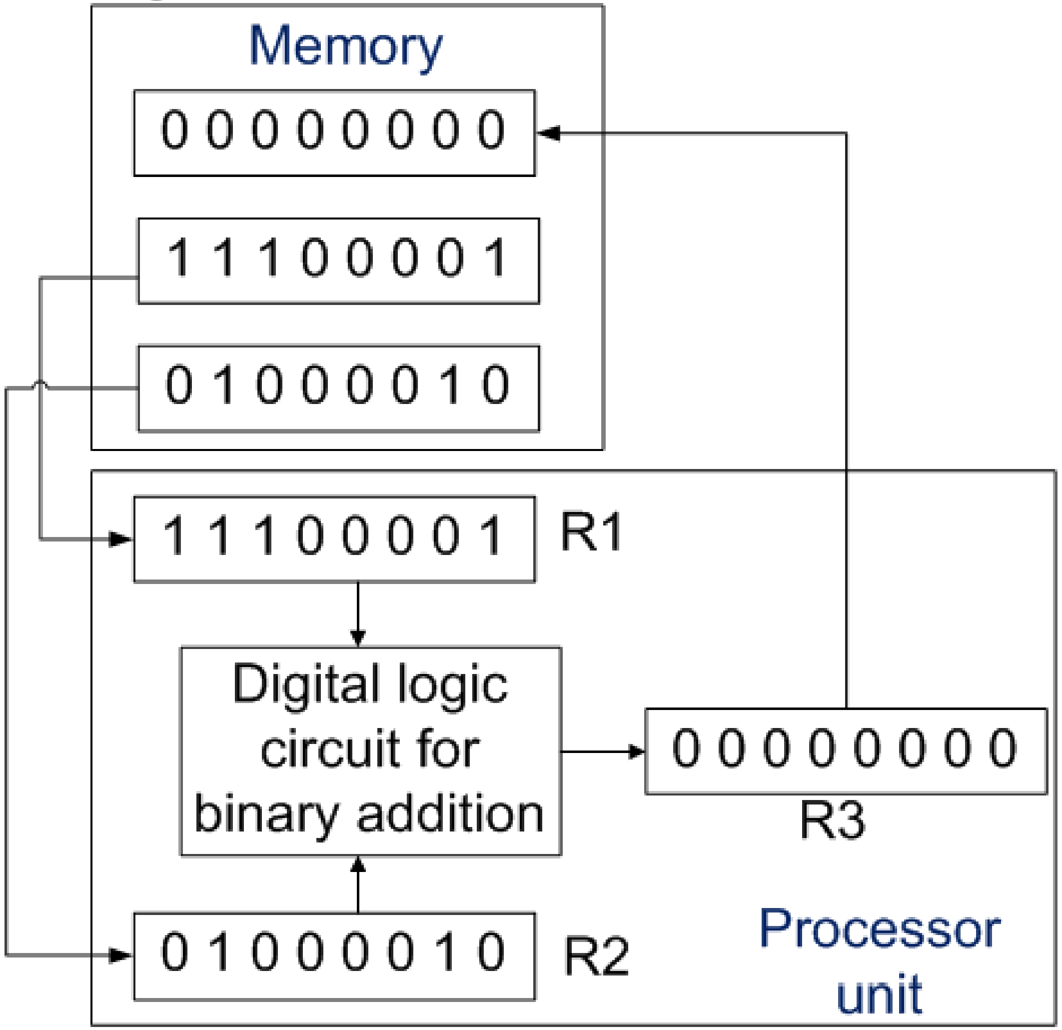
Register

- Memory (RAM) is merely a collection of thousands of registers.
- The processor unit composed of various registers that stores operands on which operations are performed.
  - For example for the addition of two numbers, first number (operand) is stored in the register A and the second number (operand) is stored in the register B and their sum after calculations is stored in the register C
- The control unit uses registers to keep track of various computer sequences
- Every input and output device must have at least one register to store the information transferred to or from the device

Register Transfer



Register Transfer



Logic

- NOT
- AND
- OR

Logic Gates

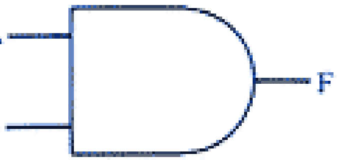
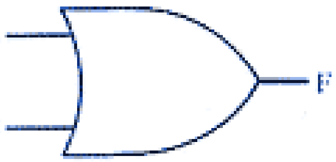
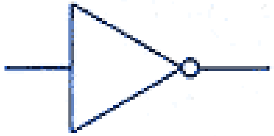
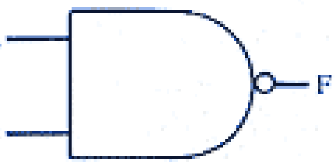
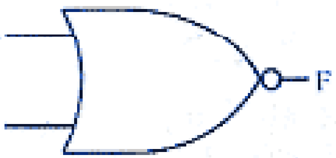
<table>
<tr><th>Name</th><th>Graphic Symbol</th><th>Algebraic Function</th><th>Truth Table</th></tr>
<tr><td>AND</td><td>
</td><td>$F = A \cdot B$
or
$F = AB$</td><td>A B | F
0 0 | 0
0 1 | 0
1 0 | 0
1 1 | 1</td></tr>
<tr><td>OR</td><td>
</td><td>$F = A + B$</td><td>A B | F
0 0 | 0
0 1 | 1
1 0 | 1
1 1 | 1</td></tr>
<tr><td>NOT</td><td>
</td><td>$F = \overline{A}$
or
$F = A'$</td><td>A | F
0 | 1
1 | 0</td></tr>
<tr><td>NAND</td><td>
</td><td>$F = (\overline{AB})$</td><td>A B | F
0 0 | 1
0 1 | 1
1 0 | 1
1 1 | 0</td></tr>
<tr><td>NOR</td><td>
</td><td>$F = (\overline{A + B})$</td><td>A B | F
0 0 | 1
0 1 | 0
1 0 | 0
1 1 | 0</td></tr>
</table>

Image Source: http://images.wikia.com/bmet/images/a/ab/Logic_gate.gif