Digital Electronics and Microprocessors

Class 2

CHHAYADEVI BHAMARE

Last class Coverage

- Introduction to the course
- Number System(Weighted (Positional)Number)
  - Decimal and conversion to binary, Hexadecimal and Octal.
  - Binary and conversion to decimal, Hexadecimal and octal
  - Hexadecimal and conversion to Decimal, Binary
  - Octal and conversion to Decimal and Binary

Today's class content

- BCD with example
- Gray code
- Meaning of Byte, Nibble and Word
- Alphanumeric codes
- Parity Method for error detection
- Logic Gates and Boolean Algebra(Chapter 3 T1)

BCD

- Binary Coded Decimal (BCD) is another way to present decimal numbers in binary form.
- BCD is widely used and combines features of both decimal and binary systems.
- Each digit is converted to a binary equivalent.

BCD (also called 8421 code)

- To convert the number 874_{10} to BCD:

$$8 \quad 7 \quad 4 = (1101101010)_2 = 2^9 + 2^8 + 2^6 + 2^5 + 2^3 + 2^1$$

$$0100 \quad 0111 \quad 0100 = (010001110100)_{BCD}$$

- Each decimal digit is represented using 4 bits.
- Each 4-bit group can never be greater than 9.
- Reverse the process to convert BCD to decimal.

Example

- BCD to decimal
- 0111 1100 0001 → 7 12 1

Invalid BCD, as you cannot have 12 (only 0 to 9) so there is an error in this code

BCD summary

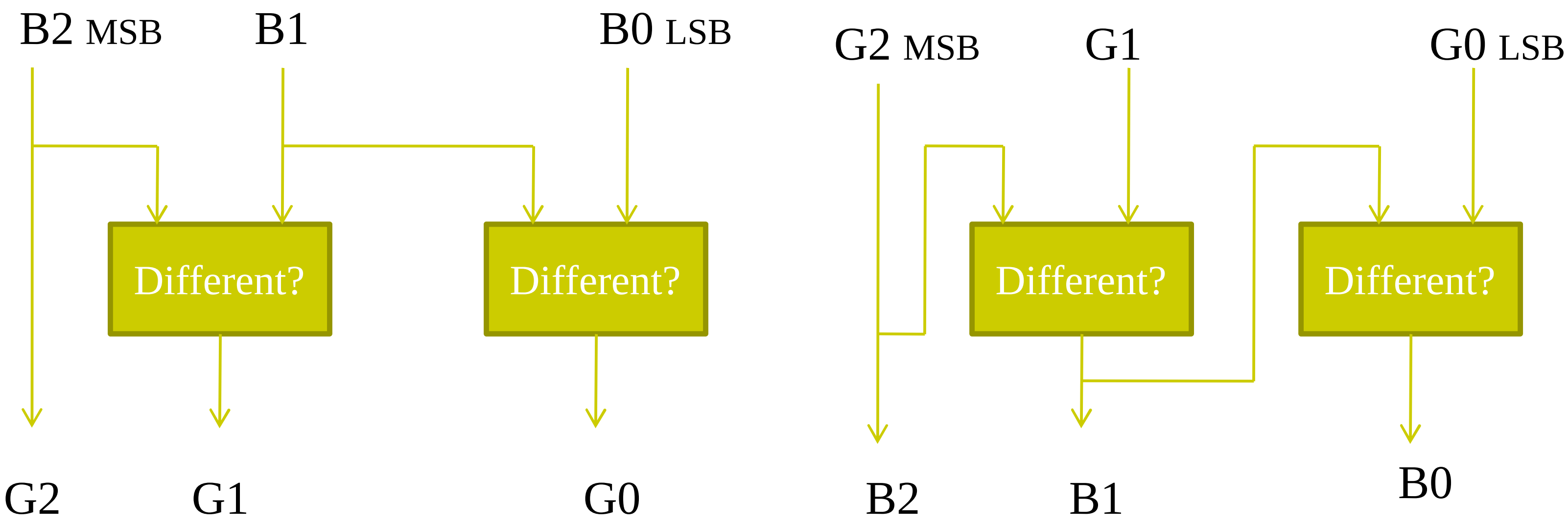
- BCD is not a number system.
- BCD is a decimal number with each digit encoded to its binary equivalent.
- A BCD number is not the same as a straight binary number.

Gray Code

- The gray code is used in applications where numbers change rapidly.
- In the gray code, only one bit changes from each value to the next.

Binary	Gray Code
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

Binary to gray and Gray to binary



If different = 1
Not Different = 0

Putting It All Together

Decimal	Binary	Hexadecimal	BCD	Gray
0	0	0	0	0
1	1	1	0001	0001
2	10	2	0010	0011
3	11	3	0011	0010
4	100	4	0100	0110
5	101	5	0101	0111
6	110	6	0110	0101
7	111	7	0111	0100
8	1000	8	1000	1100
9	1001	9	1001	1101
10	1010	A	0001 0000	1111
11	1011	B	0001 0001	1110
12	1100	C	0001 0010	1010
13	1101	D	0001 0011	1011
14	1110	E	0001 0100	1001
15	1111	F	0001 0101	1000

The Byte, Nibble, and Word

- 1 byte = 8 bits
- 1 nibble = 4 bits
- 1 word = size depends on data pathway size.
  - Word size in a simple system may be one byte (8 bits)
  - Word size in a PC is eight bytes (64 bits)

Refer Example 2-9 to 2-13 in text

Alphanumeric Codes

- Represents characters and functions found on a computer keyboard.
- ASCII – American Standard Code for Information Interchange.
  - Seven bit code: $2^7 = 128$ possible code groups
  - Table 2-5(T1) lists the standard ASCII codes
  - Examples of use are: to transfer information between computers, between computers and printers, and for internal storage.

Parity Method for Error Detection

- Binary data and codes are frequently moved between locations. For example:
  - Digitized voice over a microwave link.
  - Storage and retrieval of data from magnetic and optical disks.
  - Communication between computer systems over telephone lines using a modem.
- Electrical noise can cause errors during transmission.
- Many digital systems employ methods for error detection (and sometimes correction).

Parity Method for Error Detection

- The parity method of error detection requires the addition of an extra bit to a code group.
- This extra bit is called the parity bit.
- The bit can be either a 0 or 1, depending on the number of 1s in the code group.
- There are two methods, even and odd.

Parity Method for Error Detection

- Even parity method – the total number of bits in a group including the parity bit must add up to an even number.
  - The binary group 1 0 1 1 would require the addition of a parity bit **1** 1 0 1 1
    - *Note that the parity bit may be added at either end of a group.*
- Odd parity method – the total number of bits in a group including the parity bit must add up to an odd number.
  - The binary group 1 1 1 1 would require the addition of a parity bit **1** 1 1 1 1

Parity Method for Error Detection

- The transmitter and receiver must “agree” on the type of parity checking used.
- Two bit errors would not indicate a parity error.
- Both odd and even parity methods are used, but even is used more often.

Logic Gates and Boolean Algebra(Chapter 3 T1)

- Now that we understand the concept of binary numbers, we will study ways of describing how systems using binary logic levels make decisions.
- Boolean algebra is an important tool in describing, analyzing, designing, and implementing digital circuits.

Boolean Constants and Variables

- Boolean algebra allows only two values; 0 and 1.
- Logic 0 can be: false, off, low, no, open switch.
- Logic 1 can be: true, on, high, yes, closed switch.
- Three basic logic operations: OR, AND, and NOT.

Truth Tables

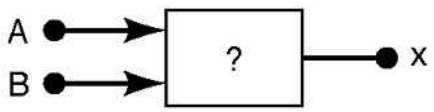
- A truth table describes the relationship between the input and output of a logic circuit.
- The number of entries corresponds to the number of inputs. For example a 2 input table would have $2^2 = 4$ entries. A 3 input table would have $2^3 = 8$ entries.

Truth Tables Examples

- Examples of truth tables with 2, 3, and 4 inputs.

Inputs / Output

A	B	x
0	0	1
0	1	0
1	0	1
1	1	0



(a)

A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(b)

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(c)

OR Operation With OR Gates

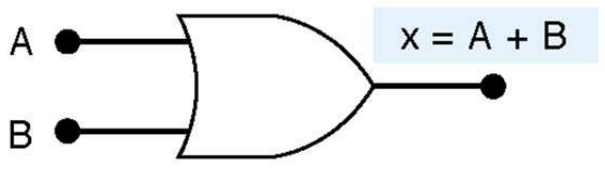
- The Boolean expression for the OR operation is

  X = A + B

  - This is read as "x equals A or B."
  - X = 1 when A = 1 or B = 1.
- Truth table and circuit symbol for a two input OR gate:

OR

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1



OR Gate

OR Operation With OR Gates

- The OR operation is similar to addition but when A = 1 and B = 1, the OR operation produces 1 + 1 = 1.
- In the Boolean expression

  $x=1+1+1=1$

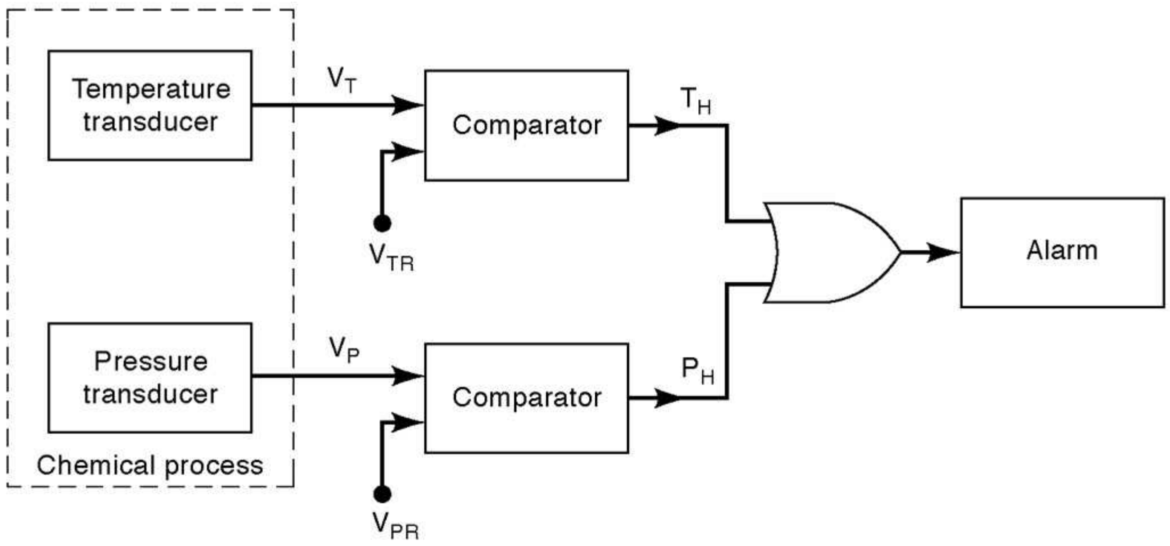
  We could say in English that x is true (1) when A is true (1) OR B is true (1) OR C is true (1).

OR Operation With OR Gates:- Application Example

- In a chemical process it may be desired that an alarm be activated whenever the process temperature exceeds a maximum value or whenever the pressure goes above a certain limit.

OR Operation With OR Gates:- Application Example

- Temperature transducer circuit produces an output voltage proportional to the process tempr.
- This voltage $V_{T,}$ is compared with a temperature reference voltage V_{TR} , in voltage comparator circuit.
- The comparator output ,T_H is normally low voltage (logic 0), but it switches to high voltage when V_T exceeds $V_{TR.}$



AND Operation With AND Gates

- Logic Expression, Truth table, logic symbol covered in tut1
- The AND operation is similar to multiplication.
- In the Boolean expression

  $X = A \bullet B \bullet C$

  X = 1 only when A = 1, B = 1, and C = 1.

NOT Operation

- Covered in tut

$$X = \overline{A}$$

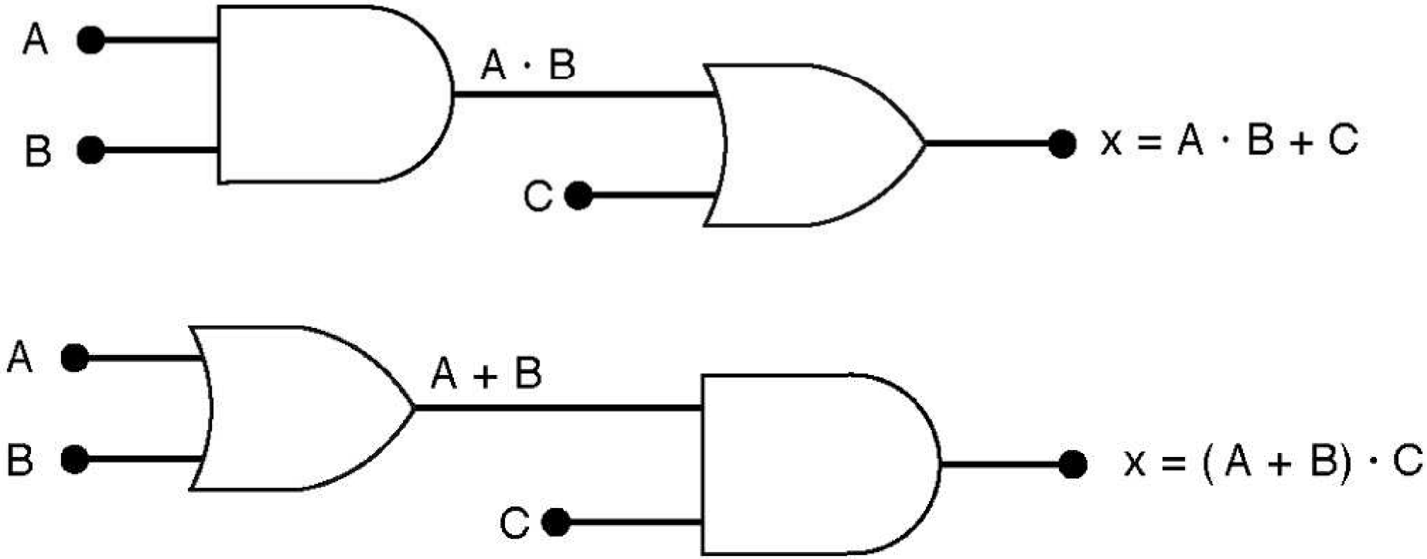
$$X = A'$$

Describing Logic Circuits Algebraically

- The three basic Boolean operations (OR, AND, NOT) can describe any logic circuit.
- If an expression contains both AND and OR gates the AND operation will be performed first, unless there is a parenthesis in the expression.
- Boolean algebra helps in analyzing and synthesizing logic circuits

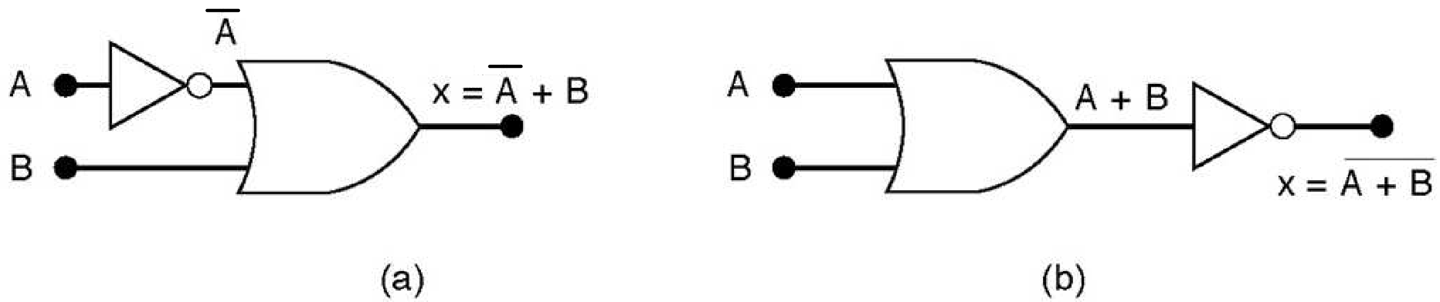
Describing Logic Circuits Algebraically

- Examples of Boolean expressions for logic circuits:



Describing Logic Circuits Algebraically

- The output of an inverter is equivalent to the input with a bar over it. Input A through an inverter equals $\overline{A}$.
- Examples using inverters.



Evaluating Logic Circuit Outputs

- Rules for evaluating a Boolean expression:
  - Perform all inversions of single terms.
  - Perform all operations within parenthesis.
  - Perform AND operation before an OR operation unless parenthesis indicate otherwise.
  - If an expression has a bar over it, perform the operations inside the expression and then invert the result.

Evaluating Logic Circuit Outputs

- Evaluate Boolean expressions by substituting values and performing the indicated operations:

$$A = 0, B = 1, C = 1, \text{and } D = 1$$

$$\begin{aligned}
x &= \overline{A}BC\overline{(A + D)} \\
&= \overline{0} \cdot 1 \cdot 1 \cdot \overline{(0 + 1)} \\
&= 1 \cdot 1 \cdot 1 \cdot \overline{(0 + 1)} \\
&= 1 \cdot 1 \cdot 1 \cdot (\overline{1}) \\
&= 1 \cdot 1 \cdot 1 \cdot 0 \\
&= 0
\end{aligned}$$

Evaluating Logic Circuit Outputs

- Output logic levels can be determined directly from a circuit diagram.
- Technicians frequently use this method.
- The output of each gate is noted until a final output is found.

Microcomputer Application:-The logic circuit in fig generates an output MEM, that is used to activate a Memory IC in a particular Microcomputer. Determine the I/P conditions necessary to activate MEM

1.MEM is active LOW, and it will go LOW only when x and y are HIGH
2.X will be HIGH only when RD=0.
3.Y will be high when either W or V is HIGH
4. W will be high when either ROM-A or ROM-B=0
5. V will be HIGH when RAM=0
6. Putting this all together, MEM will go LOW only when RD=0 and at least one of the three inputs ROM-A,ROM-B, or RAM is low