MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

Digital Electronics and Computer Architecture

SCHOOL OF COMPUTER ENGINEERING AND TECHNOLOGY

Digital Electronics and Computer Architecture

Course Objectives:

By participating in and understanding all facets of this Course a student will be able:

1. **To learn number systems, Boolean algebra and to introduce the concepts of digital logic families.**
2. **To develop skills for design and implementation of combinational logic circuits.**
3. **To develop skills for design and implementation of sequential circuits**
4. **To understand the functions, characteristics of various components of computer, instruction set, addressing mode of computer architecture and Pipeline Hazards**
5. **To acquire the knowledge of Control unit architecture and Memory Organization**

Digital Electronics and Computer Architecture

Course Outcomes:

On completion of course, students will be able to

1. **Identify and use the basic logic gates and solve logic equations with various reduction techniques of digital logic circuits.**
2. **Design and implement combinational logic circuits**
3. **Design and implement sequential logic circuits**
4. **Recognize the functions and organization of various blocks of CPU. Describe CPU instruction characteristics used in Pipelining concept and its relevance with Processor's performance.**
5. **Demonstrate computer architecture concepts related control unit and memory**

Syllabus

Module 1 Computer Arithmetic:

- Introduction to digital logic families, Introduction to Number System and Codes, Sign Magnitude representation, 1's and 2's complement, Binary Arithmetic: Addition, Subtraction of signed numbers, multiplication of positive numbers, Integer Arithmetic, Floating point representation and operations – IEEE standard, Boolean Algebra, Logic Gates, Minimization of logic functions using K map (SOP and POS)

Module 2 Combinational Logic Design:

- Design examples: Arithmetic circuits, Comparator, Code converters, Parity generators and checkers, Arithmetic and Logic Unit, design of adder and fast adder, look ahead carry generator, Implementation of SOP and POS using MUX, DMUX, DECODER.

Module 3 Sequential Logic Design:

- 1-bit Memory Cell, Flip flops, Conversion of flip flops, Design of ripple counters and synchronous counters, Modulus of the counter, Sequence generator, Lock out condition. Shift registers, Applications of Shift registers (ring and twisted ring counters), Moore and Mealy Models- State diagram, State Tables and Design Procedure, Sequence detector.

Syllabus

Module 4 Fundamental of Computer Architectures:

- **Structure and Function, basic operational concepts of bus structures Memory locations and addresses functions. Types of computer units (CPU, Memory, I/O, System Bus), Von Neumann & Harvard architecture, Key characteristics of RISC & CISC.**
- **ALU: Multiplication – Block diagram, Hardware implementation of unsigned binary and signed number, Booth's Algorithm, Division Algorithms (Unsigned Binary).**
- **Processor organization, Register organization- user visible registers, control and status registers-Case Study 8086. Instruction Cycle- The machine cycle and Data flow. Types of operands, Types of operations, Types of instructions, Addressing modes and Instruction Formats- instruction length, allocation of bits, variable length instructions instruction formats - Case Study- 8086. Processor and Instruction Pipelining, Pipeline Performance, Pipeline Hazards - Structural, Data, Control.**

Syllabus

Module 5 Control Unit and Memory System:

- **Hardwired control, Micro-programmed control- micro instructions, micro-program sequencing, wide branch addressing, microinstruction with next address field, prefetching microinstructions and emulation.**
- **Memory hierarchy, characteristics, Cache Memory- Cache memory principles, Elements of cache design- cache address, size, mapping functions, replacement algorithms, write policy, line size, number of cache, one level and two level cache, performance characteristics of two level cache- locality & operations, main memory and secondary storage. Pentium IV cache organization**

Syllabus

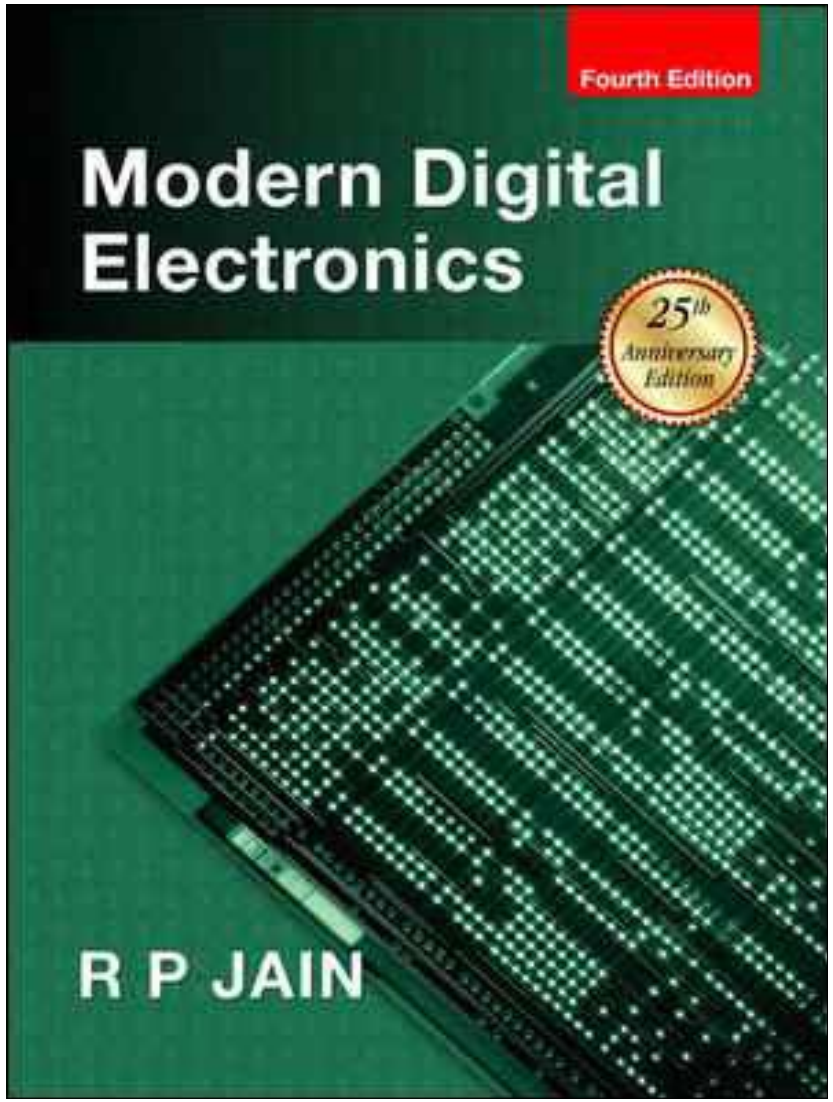
List of Assignments:

1. **Design and Implement Code Converters using logic gates**
2. **Design and Implement Combinational Logic Design using MUX/Decoder ICs**
3. **Design and Implement asynchronous counter using JK- Flip flop.**
4. **Design and Implement MOD-N asynchronous counter using JK- Flip flop /IC7490.**
5. **Design and implement Synchronous Counter**
6. **Design of a sequence detector**
7. **Write an assembly language program (ALP) to display 2-digit and 4-digit hex numbers using 64- bit assembly language programming**
8. **Write an assembly language program (ALP) to accept 2-digit and 4-digit hex numbers user input using 64- bit assembly language programming.**
9. **Write an assembly language program (ALP) to implement addition and subtraction of 8-bit numbers (Using user input, macro and procedure).**
10. **Write an ALP to perform basic string operations. (length and Reverse)**

Module 1 Computer Arithmetic

- **Introduction to digital logic families,**
- **Introduction to Number System and Codes,**
- **Sign Magnitude representation, 1's and 2's complement,**
- **Binary Arithmetic: Addition, Subtraction of signed numbers, multiplication of positive numbers, Integer Arithmetic,**
- **Floating point representation and operations – IEEE standard, Boolean Algebra, Logic Gates, Minimization of logic functions using K map (SOP and POS)**

Text book soft copy available



Digital and Analog Quantities

For many years, applications of digital electronics were limited to computer systems.

Digital Technology have wide range of applications

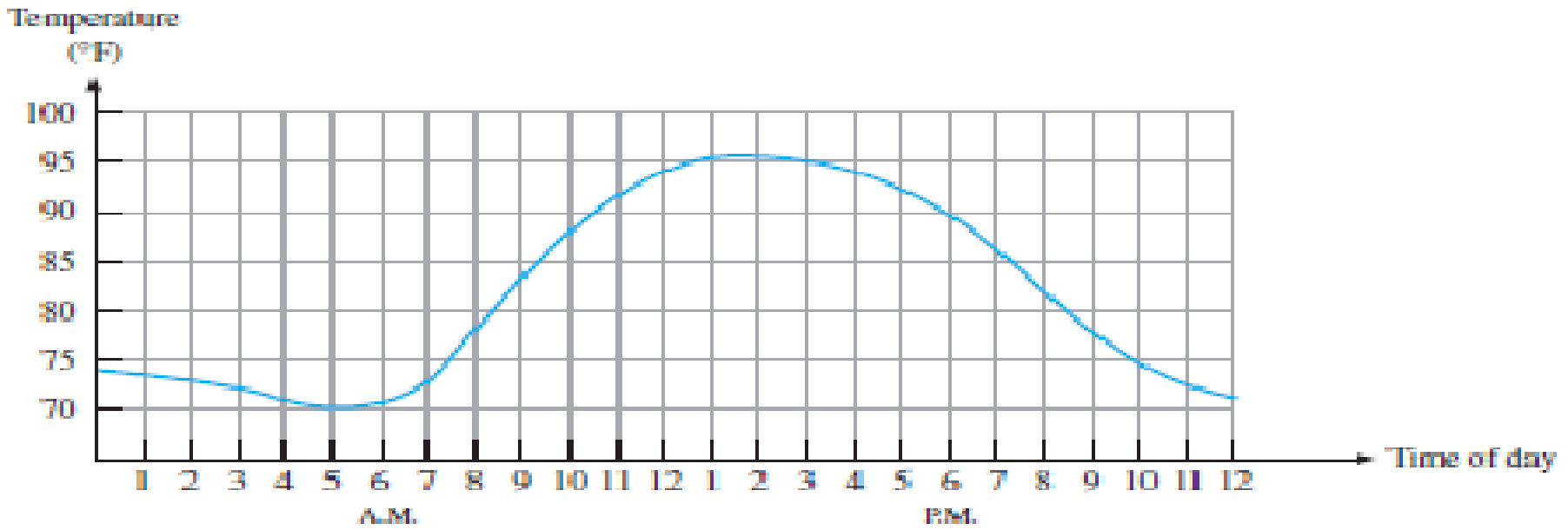
- ✓ Television
- ✓ Communication systems
- ✓ Radar
- ✓ Navigation
- ✓ Military Systems
- ✓ Medical Instrumentations
- ✓ Industrial Process Control
- ✓ Consumer electronics etc.

Electronics circuits can be divided into two broad categories

- ✓ Digital
- ✓ Analog

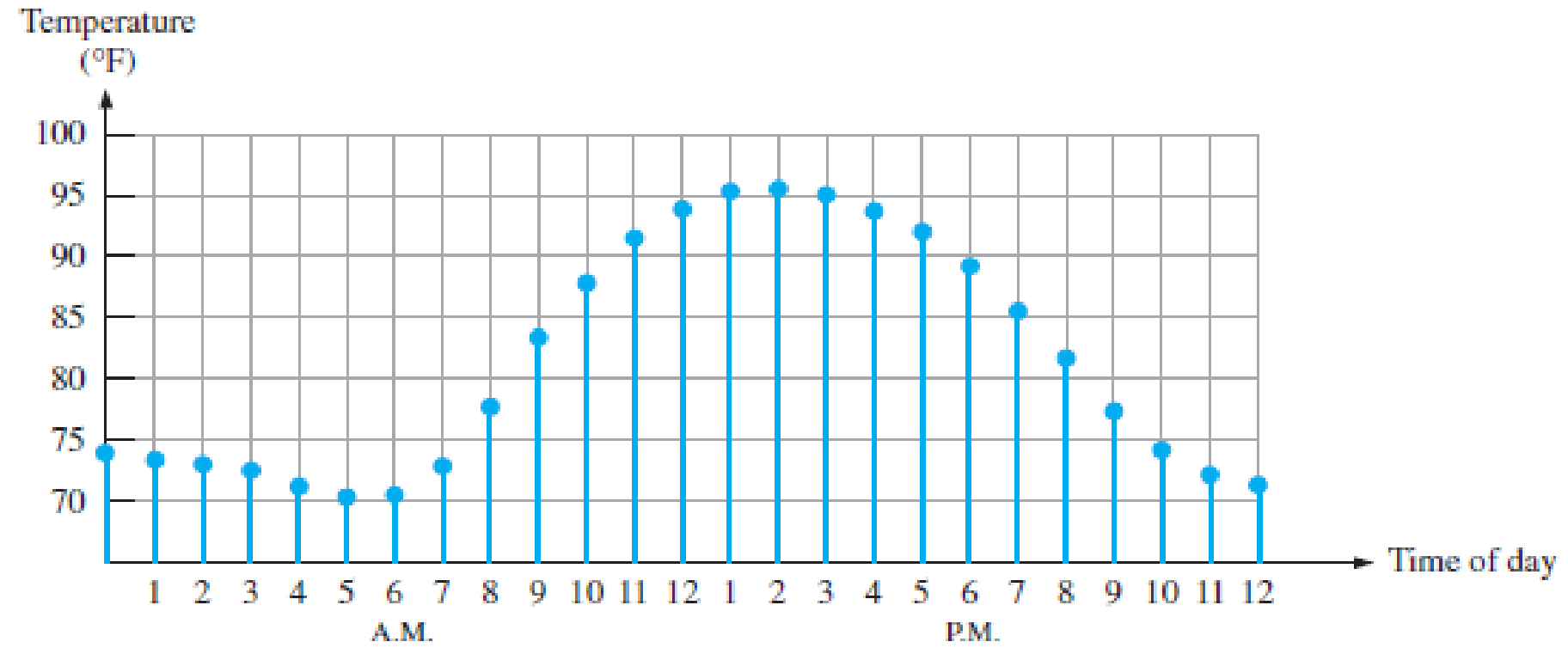
Digital and Analog Quantities

- Analog: is one having continuous values
- Digital: is one having discrete set of values
- Example: Air temperature changes over continuous range of values
- **Analog Signal**



Graph of an analog quantity (temperature versus time).

Digital Code

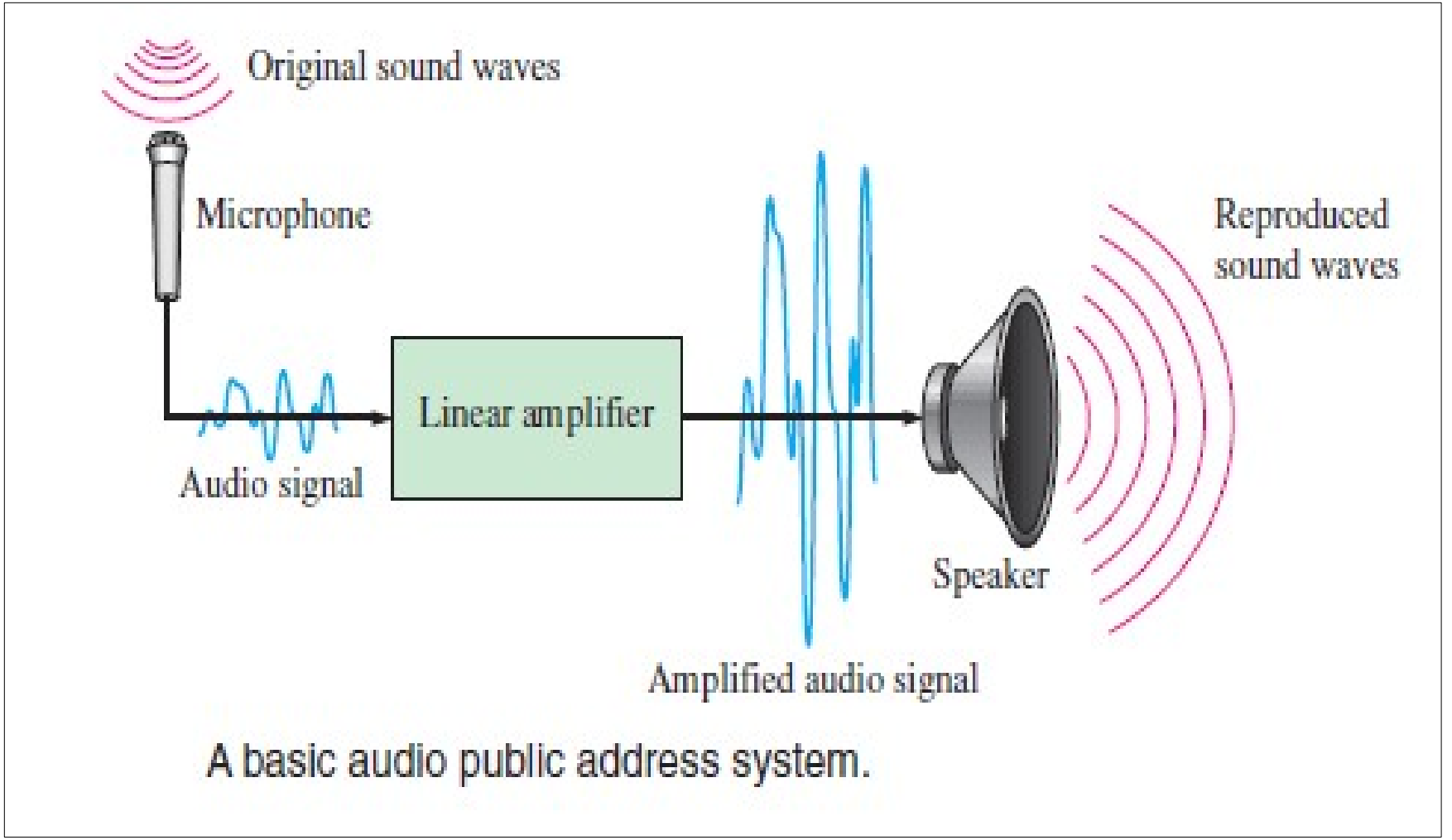


Sampled-value representation (quantization) of the analog quantity in
Each value represented by a dot can be digitized by representing it as a digital code that consists of a series of 1s and 0s.

Advantages of Digital Signals

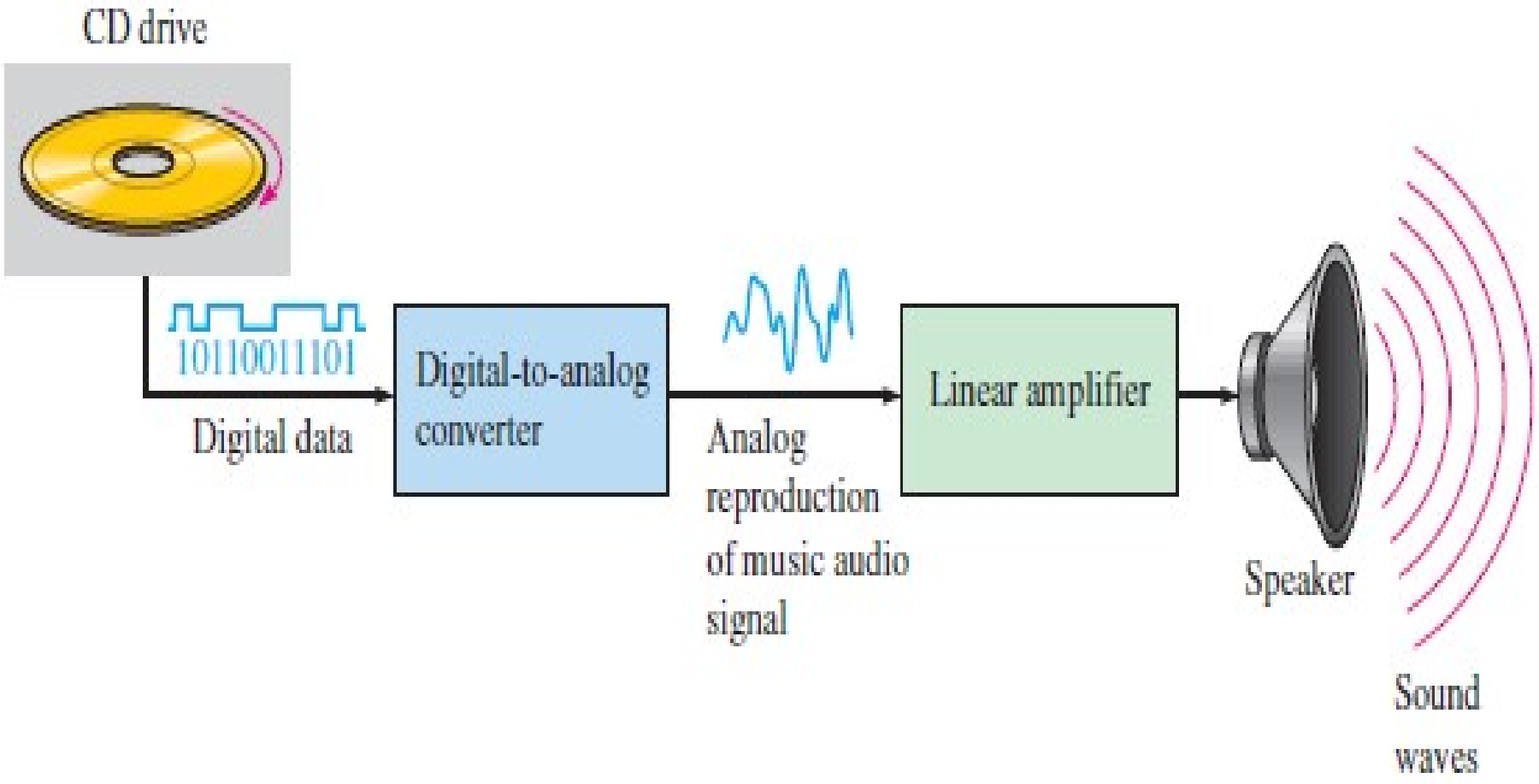
- Data transmission is more effective and reliable
- Digital data has a great advantage when storage is necessary
- Digital data is more accurate and precise
- Its ripple free or noise free

Analog Systems



A basic audio public address system.

A System Using Analog and Digital Methods

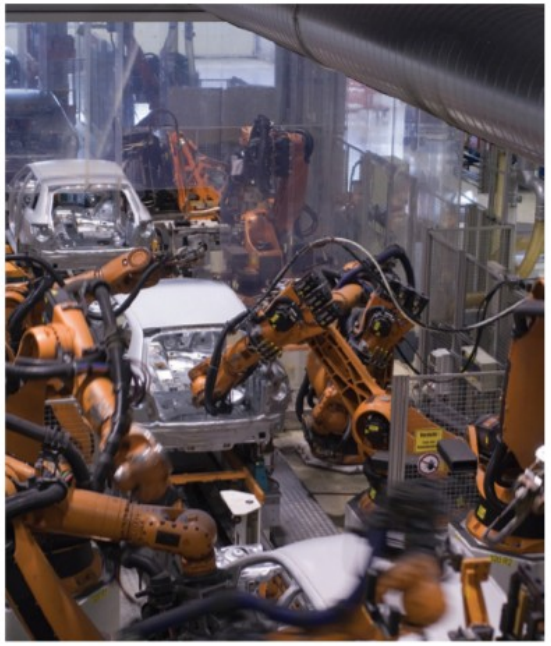


Basic block diagram of a CD player. Only one channel is shown.

Mechatronics

- The interdisciplinary field that comprises both mechanical and electronic components is known as mechatronics
- Both digital and analog electronics are used in the control of various mechanical systems.

(b) Robotic arm



(c) Automotive assembly line

Digital Electronics

- It uses Binary Digits 0 and 1
- Each of the two digits in the binary system, 1 and 0, is called a bit, Two different voltage levels are used to represent the two bits.
- 1 - is represented by the higher voltage, which will be referred as a HIGH,
- 0 - is represented by the lower voltage level, which will be referred as a LOW.
- This is called positive logic and will be used throughout the topic.

HIGH = 1 and LOW = 0

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Number System and Binary Codes

Number System and Binary Codes

- Number systems
- Representation of unsigned and signed integers
- Fixed-point representation of real numbers
- Floating-point representation of real numbers
- Binary representation
- Codes and their conversions
- Binary arithmetic

Number Systems

- **Decimal**

$$1234_{10} = (1 \times 10^3) + (2 \times 10^2) + (3 \times 10^1) + (4 \times 10^0)$$

- **Binary**

$$1101_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

- **Octal**

$$123_8 = (1 \times 8^2) + (2 \times 8^1) + (3 \times 8^0)$$

- **Hexadecimal**

$$123_{16} = (1 \times 16^2) + (2 \times 16^1) + (3 \times 16^0)$$

here we need 16 characters – 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Binary number into Decimal

- Convert the given binary number into decimal equivalent number

1. $(100010)_2=0\times2^0+1\times2^1+0\times2^2+0\times2^3+0\times2^4+1\times2^5$

$=0+2+0+0+0+32$

$=(34)_{10}$

2. $(0.1011)_2=1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+1\times2^{-4}$

$=0.5+0+0.125+0.0625$

$=(0.6875)_{10}$

3. $(10101.011)_2$

Integer part: $(10101)_2=1\times2^0+0\times2^1+1\times2^2+0\times2^3+1\times2^4$

$=1+0+4+0+16=(21)_{10}$

Decimal part: $(0.011)_2=0\times2^{-1}+1\times2^{-2}+1\times2^{-3}$

$=(0.375)_{10}$

$=(21.375)_{10}$

Hexadecimal Number System

Hexadecimal number system	Equivalent binary number
	8 4 2 1 (weights)
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0 = A
11	1 0 1 1 = B
12	1 1 0 0 = C
13	1 1 0 1 = D
14	1 1 1 0 = E
15	1 1 1 1 = F

Hexadecimal Number System:

The hexadecimal number system has base 16. It has 16 distinct digit symbols. It uses the digits 0-9 & letters A,B,C,D,E,F as 16 digit symbols.

- Hexadecimal to Binary Conversion

Hexadecimal C 3

Binary 1100 0011

- Binary to Hexadecimal Conversion

Binary 1110 1010

Hexadecimal E A

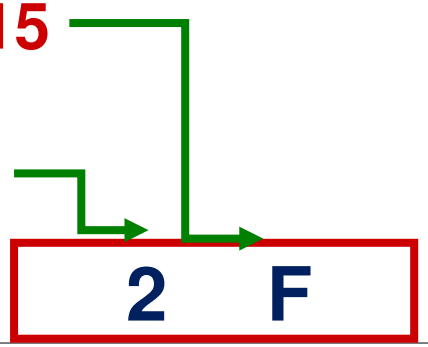
Hexadecimal Number System

- **DECIMAL TO HEXADECIMAL CONVERSION**

Divide by 16 Process

Decimal # 47 ÷ 16 = 2 remainder 15

2 ÷ 16 = 0 remainder 2



HEXADECIMAL NUMBER INTO DECIMAL NUMBER SYSTEM

1. $(2B8D.E2)16 = 2\times16^3+11\times16^2+8\times16^1+13\times16^0+14\times16^{-1}+2\times16^{-2}$

$= 8192+2816+128+13+0.875+0.0078125$

$= (11149.88281)_{10}$

$(269)_{16} = ?$

Answer: $(617)_{10}$

Octal Number System

A number system that uses 8 digit (0-7) is called an octal number system. It has base 8.

Octal number system into Binary number system:

$(7423.245)_8$

7	4	2	3	.	2	4	5
111	100	010	011	.	010	100	101

Conversion from Binary number system to Octal number system:

$$(11101101110.11111)_2$$

011	101	101	110	.	111	110
3	5	5	6	.	7	6

Octal Numbers	Binary Equivalent number
	4 2 1 (weights)
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

Octal Number System

- Octal to Decimal Conversion
- Multiplying each Octal digit by its positional weight

$$(314)_8 = 3\times8^3+3\times8^2+1\times8^1+4\times8^0$$
$$= 192 + 8 + 4$$
$$= (204)_{10}$$

- Decimal to Octal Conversion
- Repeated division or double dabble method

$$(1375)_{10} = (2537)_8$$

remainder

8	1375	
8	171	7
8	21	3
	2	5
		2

MSB

- **Hexadecimal number system to Octal number system:**

First write down the 4 bit binary equivalent of hexadecimal digit and then rearranging into group of three bit each.

$(2AB.9)_{16} = (?)_8$

Hexadecimal equivalent	2 A B . 9 0010 1010 1011 . 1001
Octal equivalent	001 010 101 011 . 100 100 1 2 5 3 . 4 4

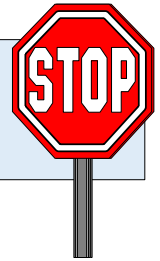
Convert $(1375)_8 = (\quad)_{16}$

1375 = 001 011 111 101

Now 001011111101 **0010 1111 1101**

2 15 13

TEST



- Convert Hexadecimal number A6 to Binary

  1010 0110 (Binary)

- Convert Hexadecimal number 16 to Decimal

  22 (Decimal)

- Convert Decimal 63 to Hexadecimal

  3F (Hexadecimal)

- Convert Decimal 41.915 to Binary

  101001.11101

- Convert hexadecimal 7FD6 to Octal

  77726

Takeaway

- **Digital computers are implemented via logic circuits and thus represent *all* numbers in binary (base 2).**
- **We (humans) often write numbers as decimal and hexadecimal for convenience, so need to be able to convert to binary and back (to understand what computer is doing!).**

Representation of Unsigned and Signed Integers

- **Are the negative numbers just numbers with a minus sign in the front? This is probably true…but there are issues to represent negative numbers in computing systems**
- **Common schemas:**
  - Sign-magnitude
  - Complementary representations:
    - 1's complement
    - ***2's complement – most common & important***

One’s complement

- For example, in 8-bit one’s complement,
- positive 3 is: 00000011
- Negative 3 is: 11111100
- In one’s complement, as with signed magnitude, negative values are indicated by a 1 in the high order bit.

- Complement systems are useful because they eliminate the need for subtraction. The difference of two values is found by adding the minuend to the complement of the subtrahend.

Two's Complement Representation

- **Step 1: First complement all the bits (that is find one's complement) Make all 1s as 0s and all 0s as 1s**
- **Step 2:Then perform increment by 1 Add 0001 b**

Signed Numbers

- **Range of Values**

  Total combinations = 2^n

  2's complement form:

  $$-(2^{n-1}) \text{ to } +(2^{n-1}-1)$$

Range for 8 bit number:

n = 8

$-(2^{8-1}) = -2^7 = -128$ minimum

$+(2^{8-1}) - 1 = +2^7 - 1 = +127$ maximum

Total combination of numbers is $2^8 = 256$.

WHAT IS A BINARY CODE ?

- In the coding, when **numbers, letters or words** are represented by a **specific group of symbols**, it is said that the number, letter or word is being encoded.

- The group of symbols is called as a code. The digital data is represented, stored and transmitted as group of binary bits. This group is also called as **binary code**. The binary code is represented by the number as well as alphanumeric letter.

Classification of Binary Codes

- Weighted Codes
- Non-Weighted Codes
- Reflective Codes
- Sequential Codes
- Alphanumeric Codes
- Error Detecting and Correcting Codes

CODE

- Weighted and Non-Weighted Code
- In weighted code, each digit represents specific weight.
  - Binary code has a weight 8 4 2 1
- In non-weighted code weight is not assigned to position.
  - Excess 3 and Gray Codes.

Type of Codes

- **Binary - follows 8,4,2,1**
  - Example – decimal 12 – Binary 1100
- **BCD – Binary Coded Decimal** – each digit of decimal number is represented by a group of bits. Follows 8,4,2,1.
  - Example – decimal 12 – BCD 0001 0010
- **Excess -3 Code** – derived from BCD code. Excess -3 is calculated by adding 3 to each BCD code. It is non weighted and sequential code.
  - Example BCD 12 – 0001 0010 its Excess -3 code is – 0100 0101
- **Gray Code**

Codes and their conversions

- Gray Code
- Gray code is a code where only one bit changes at a time while traversing from 0 to any decimal number in sequence.
- Gray Code is nonweighted and cyclic code
- Not arithmetic code
- Optical shaft encoders for disk to reduce errors

Binary to Gray Code Conversion

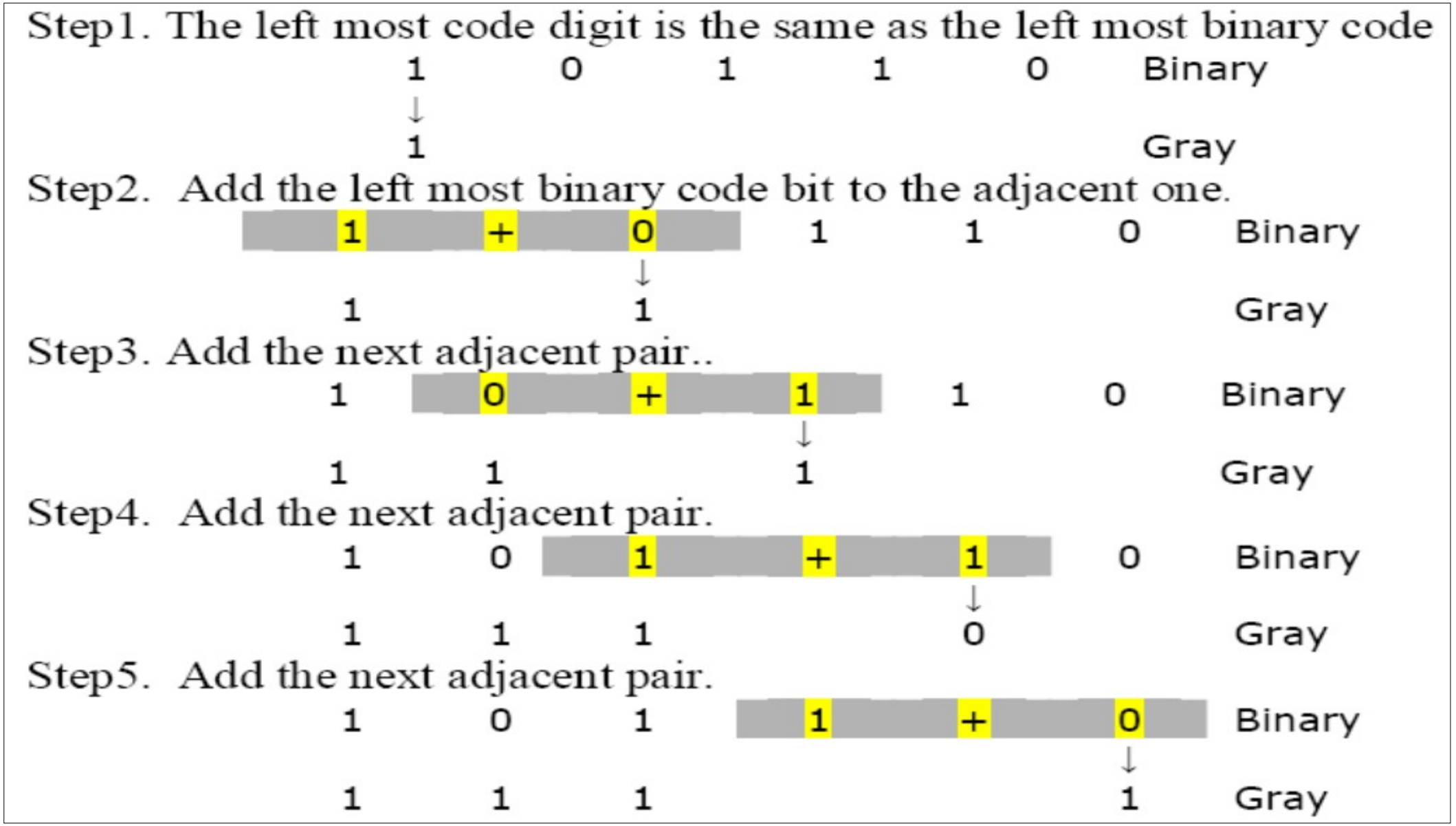
1. Copy the MSB bit as it is
2. Perform EX-OR operation between previous and current bit and write the result
3. Repeat the step 2 for all bits

Example – Binary 1011

1 , (1 EX-OR 0) , (0 EX-OR 1), (1 EX-OR 1)

1, 1, 1, 0

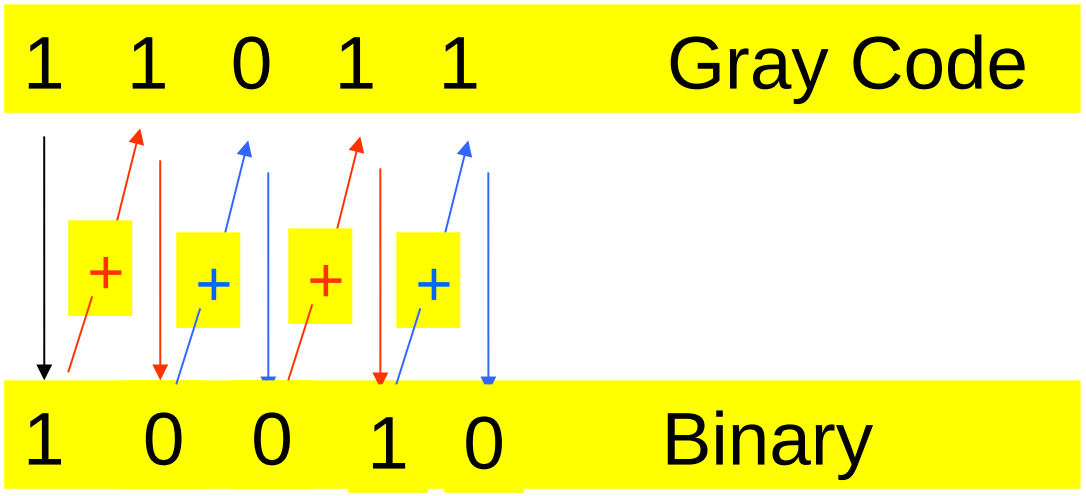
Binary to Gray Code Conversion



Gary to Binary Code Conversion

- MSB of binary is set to the MSB of Gray code
- Going from left to right, add the binary bit to the next Gray code bit. Discard carries.
- Example. Convert 11011 to binary

MSB of Gray Code is 1, so set the MSB of the binary number to 1.



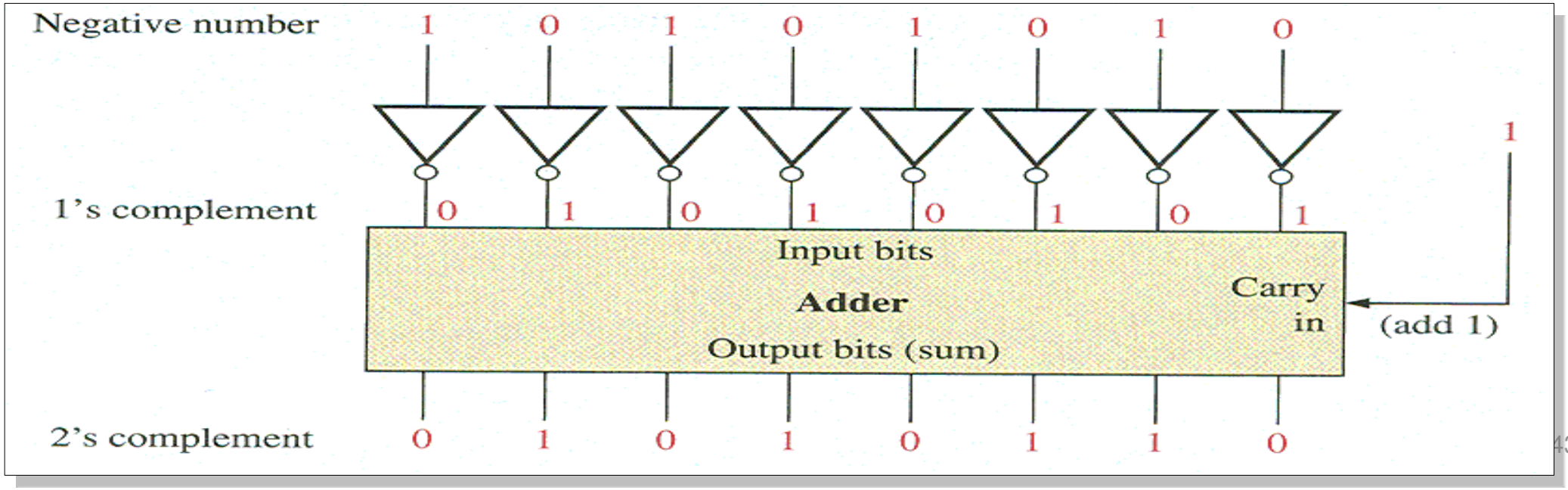
4 Bit Binary To Gray Code Conversion:

Decimal Value	INPUT (BINARY CODE)				OUTPUT (GRAY CODE)			
	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Sign Magnitude
1’s Complement
2’s Complement

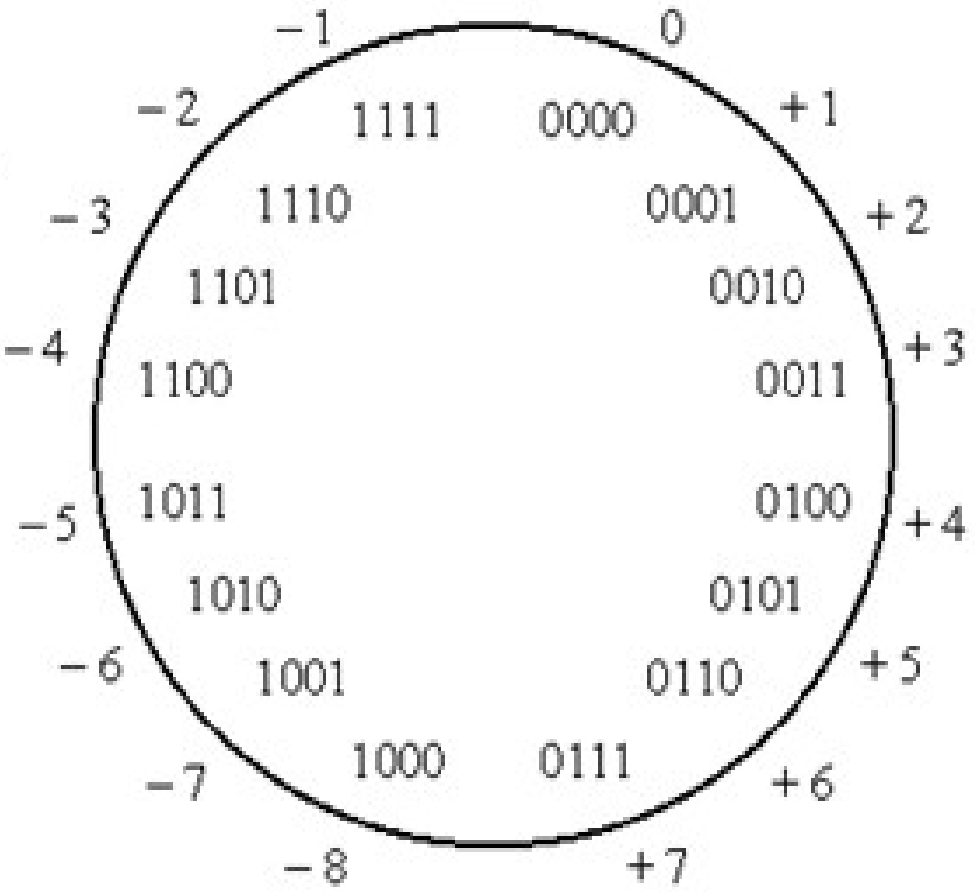
Terminology: One's-Complement and Two's-Complement

- The ***one's-complement*** of a binary number is the binary number that you get when you invert each bit.
- Example: What is the one's-complement of 10101010?
- The ***two's-complement*** of a binary number is the binary number that you get when you invert each bit and then add 1 to the result.
- Example: What is the two's-complement of 10101010?



Twos Complement Representation

- Negative number: Bitwise complement plus one
  - $0011 \equiv 3_{10}$
  - $1101 \equiv -3_{10}$
- Number wheel
- Only one zero!
- msb is the sign digit
  - $0 \equiv$ positive
  - $1 \equiv$ negative



Binary Arithmetic

Binary addition:-

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- If addition is unsinged –keep on adding from LSB to MSB
- In case of Signed Numbers – Sign bit and Concept of Sign bit extension need to be used.

Integer Arithmetic: Addition

+7 = 0111
2's complement of +7= 1001

+4 = 0100
2's complement of +4= 1100

+1 = 0001
2's complement of +1= 1111

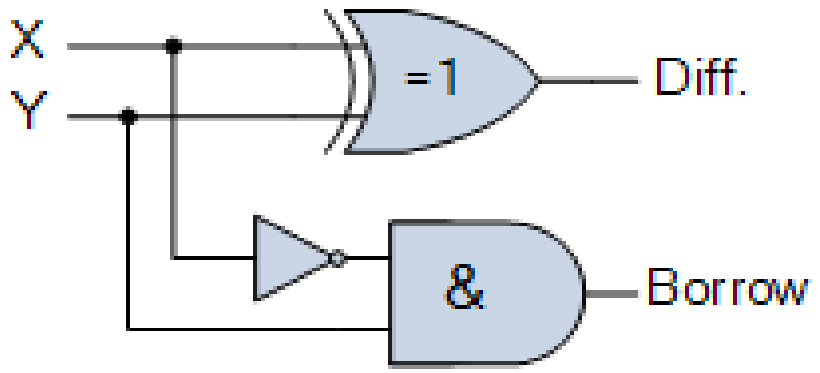
+6 = 0110
2's complement of +6= 1010

1001 = −7 +0101 = 5 1110 = −2 0010 (a) (−7) + (+5)	1100 = −4 +0100 = 4 10000 = 0 (b) (−4) + (+4)
0011 = 3 +0100 = 4 0111 = 7 (c) (+3) + (+4)	1100 = −4 +1111 = −1 11011 = −5 0101 (d) (−4) + (−1)
0101 = 5 +0100 = 4 1001 = Overflow (e) (+5) + (+4)	1001 = −7 +1010 = −6 10011 = Overflow 1101 (f) (−7) + (−6)

If two numbers are added, and they are both positive or both negative, then overflow occurs if and only if the result has the opposite sign.
If A and B are of opposite sign, then overflow cannot occur.

Binary Subtraction

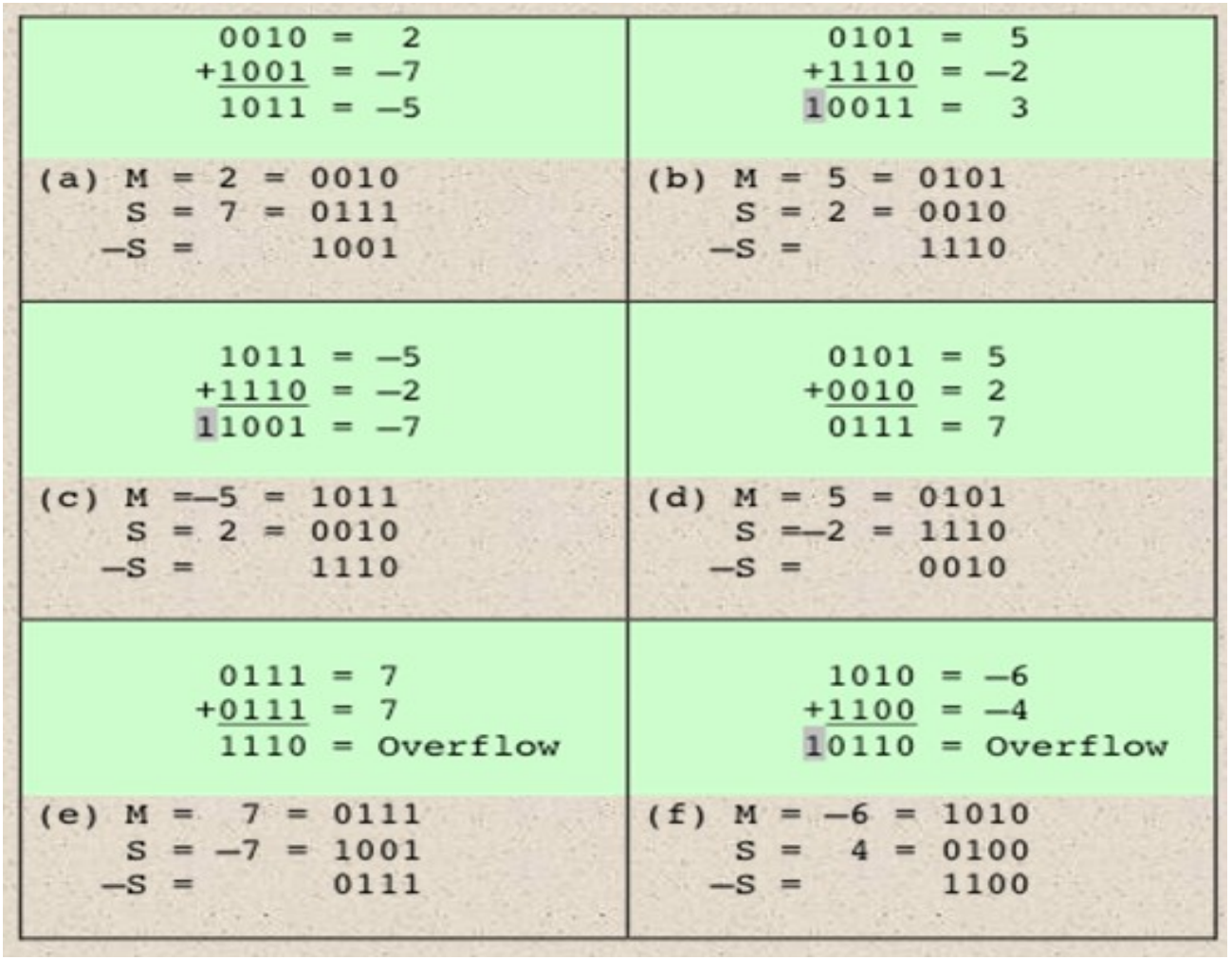
Case	A - B	Subtract	Borrow
1	0 - 0	0	0
2	1 - 0	1	0
3	1 - 1	0	0
4	0 - 1	1	1

Symbol	Truth Table			
	Y	X	DIFFERENCE	BORROW
	0	0	0	0
	0	1	1	0
	1	0	1	1
	1	1	0	0

Binary Subtraction

1. Negate a number (by taking its 2’s-complement).
2. Add binary numbers.

Subtraction



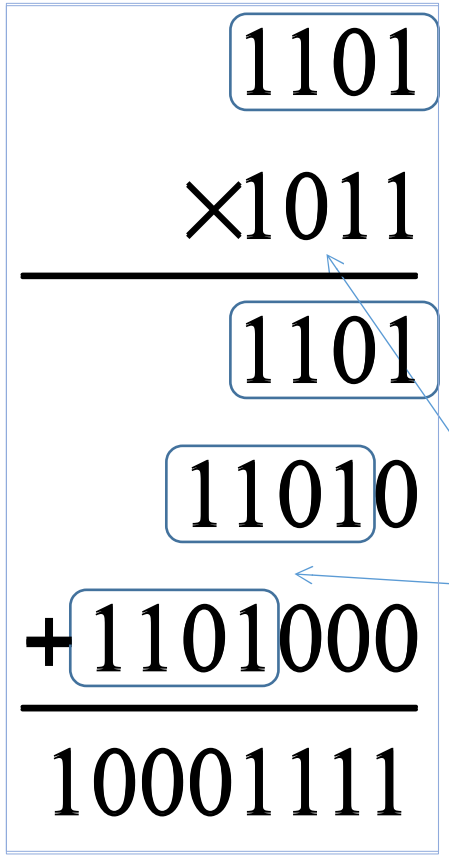
```
      0010 =  2
     +1001 = −7
      1011 = −5
(a) M = 2 = 0010
    S = 7 = 0111
   −S =     1001

      1011 = −5
     +1110 = −2
     11001 = −7
(c) M =−5 = 1011
    S = 2 = 0010
   −S =     1110

      0111 = 7
     +0111 = 7
      1110 = Overflow
(e) M =  7 = 0111
    S = −7 = 1001
   −S =     0111

      0101 =  5
     +1110 = −2
     10011 =  3
(b) M = 5 = 0101
    S = 2 = 0010
   −S =     1110

      0101 = 5
     +0010 = 2
      0111 = 7
(d) M = 5 = 0101
    S =−2 = 1110
   −S =     0010

      1010 = −6
     +1100 = −4
     10110 = Overflow
(f) M = −6 = 1010
    S =  4 = 0100
   −S =     1100
```

SUBTRACTION RULE: a - b = a + (-b)

To subtract one number (Subtrahend) from another (Minuend), take the twos complement (negation) of the subtrahend and add it to the minuend.

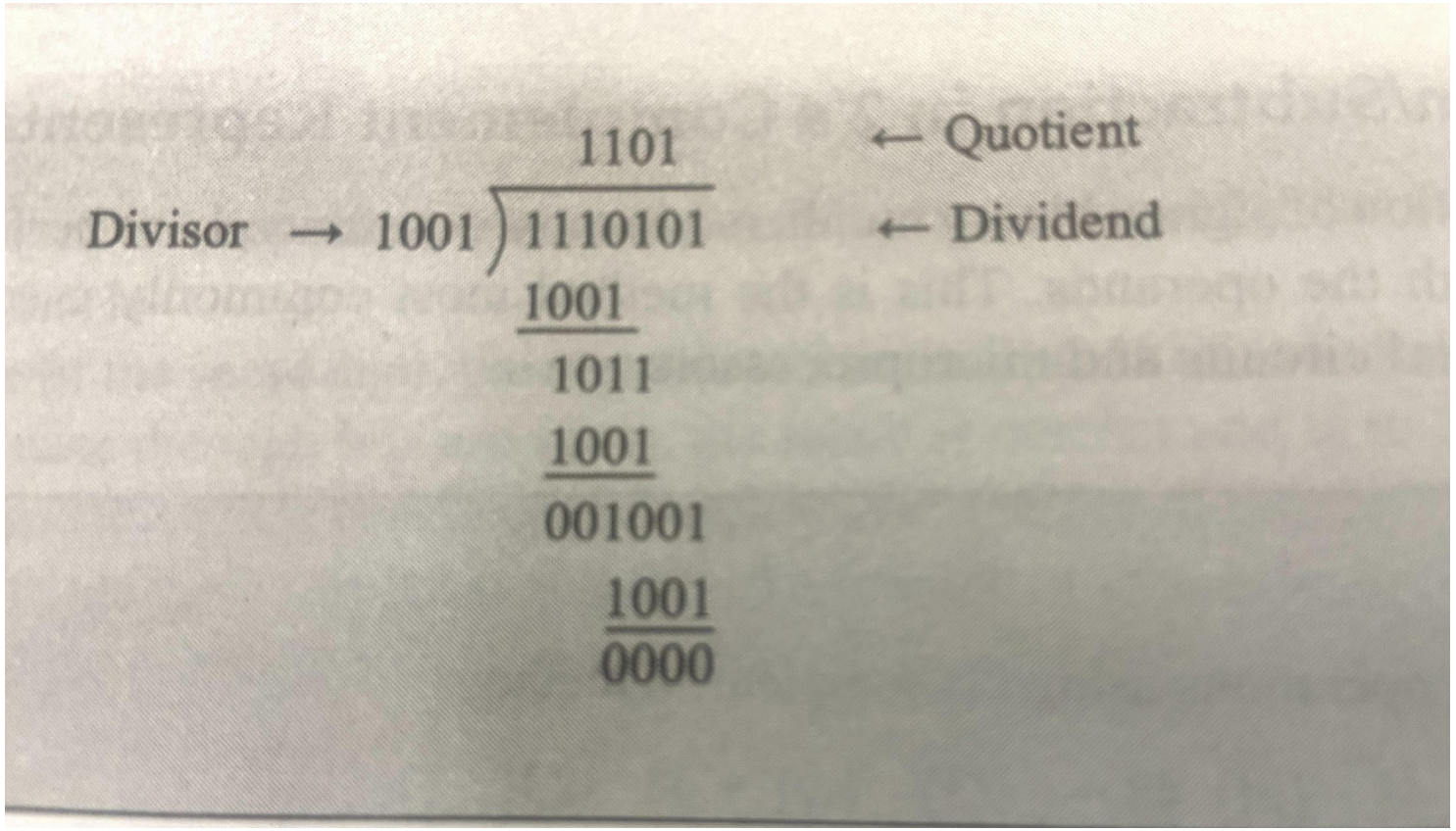
Multiplication (binary)

It's interesting to note that binary multiplication is a sequence of shifts and adds of the first term (depending on the bits in the second term.

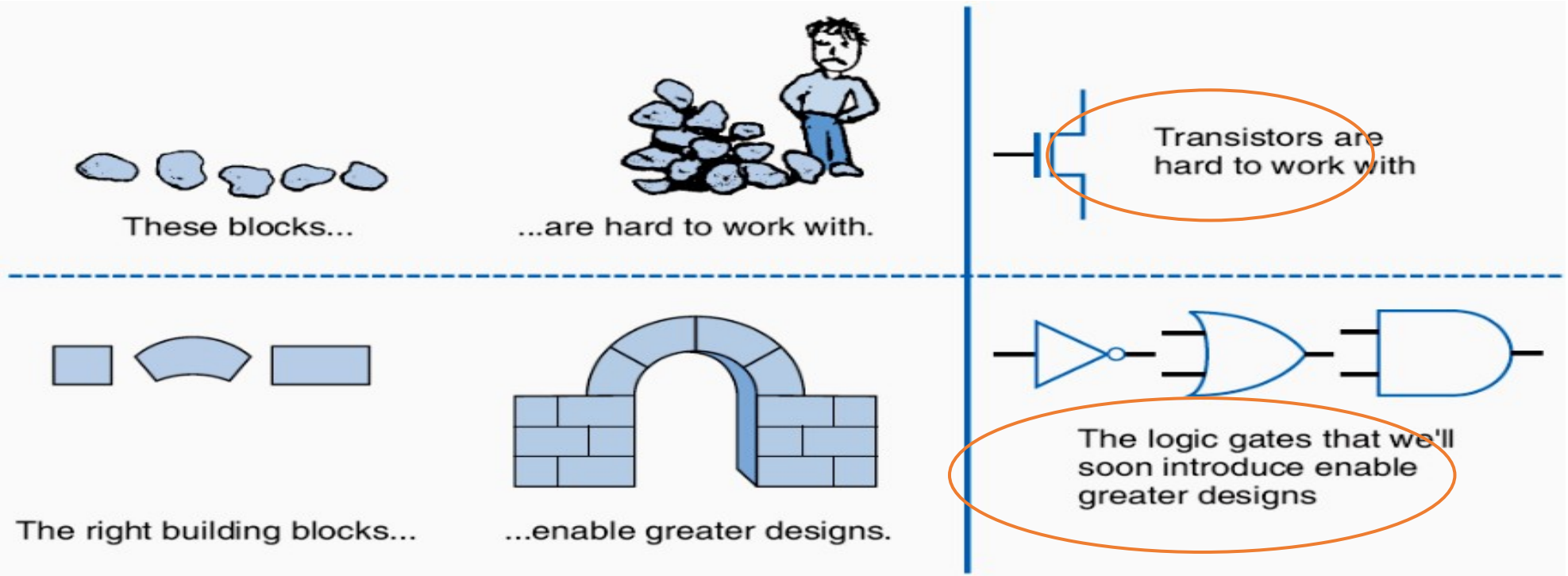


110100 is missing here because the corresponding bit in the second terms is 0.

Binary Division



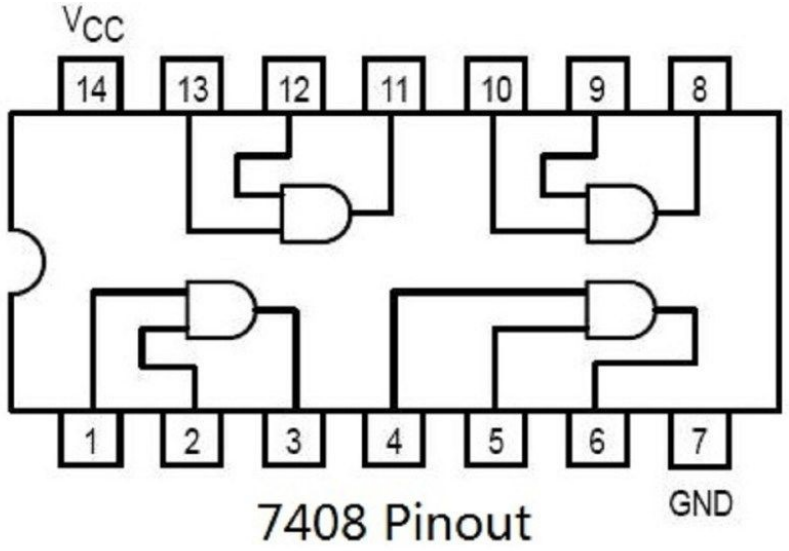
Boolean Logic Gates (Building Blocks for Digital Circuits)
(Because Switches are Hard to Work With)



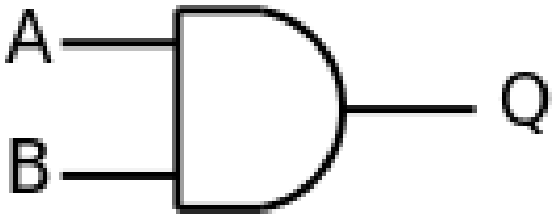
“Logic gates” are better digital circuit building blocks than switches (transistors), Why?...

Logic Gates

- **AND Gate**



7408 Pinout

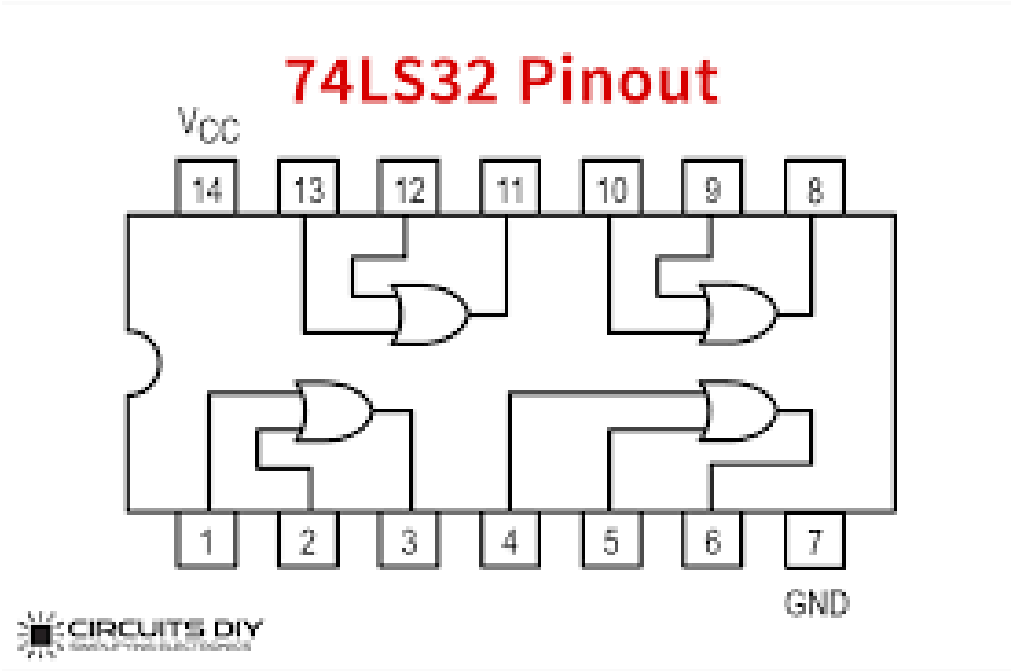


AND Gate

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

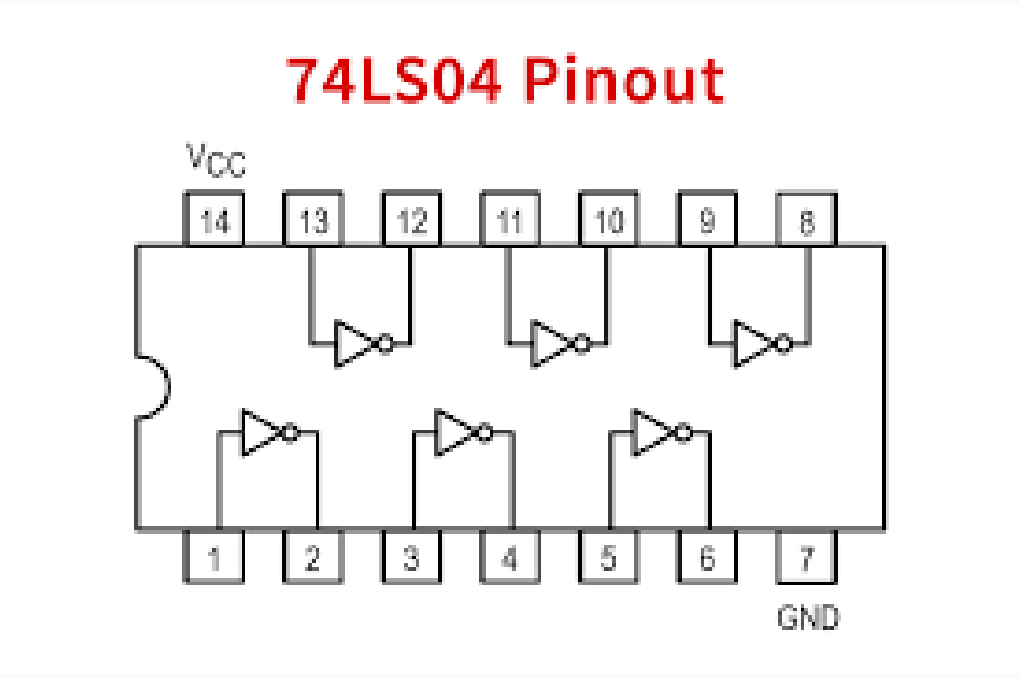
- **OR Gate**



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

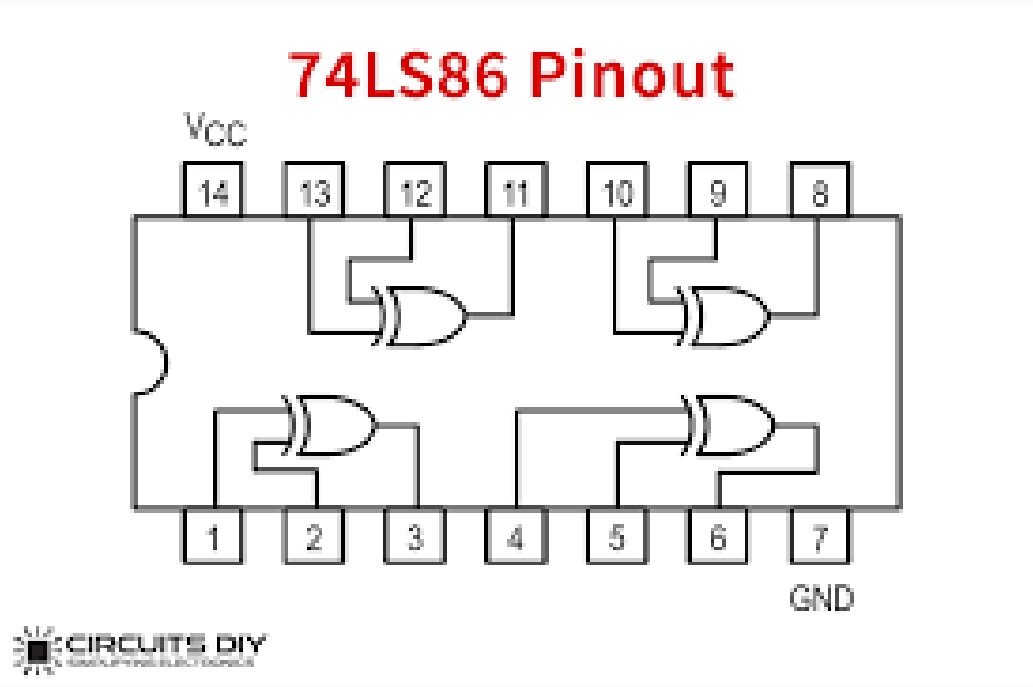
NOT gate

- **NOT Gate**



Input	Output
A	Y
0	1
1	0

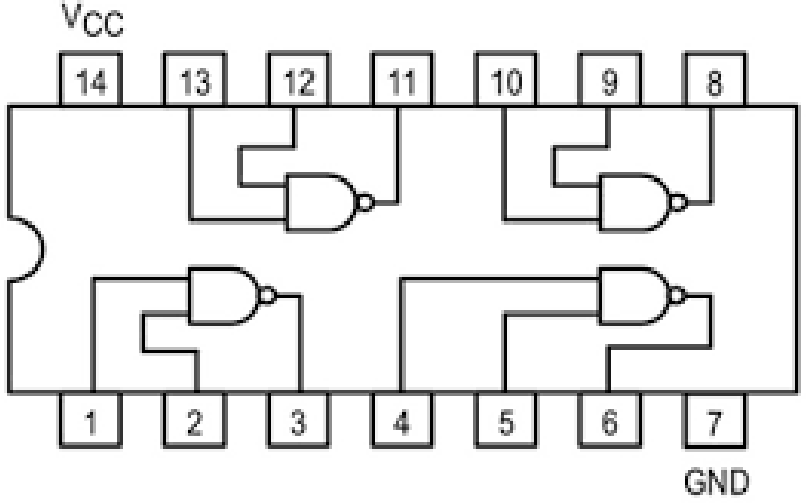
EX-OR

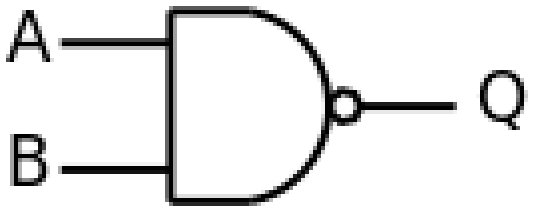
- 

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

NAND Gate

- **NAND Gate IC 7400**



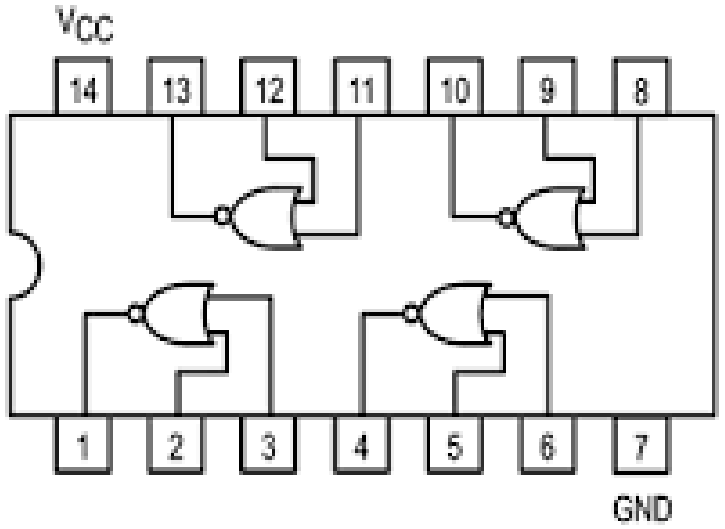


NAND Gate

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate

- **NOR Gate IC 7402**



A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Boolean Algebra Terminology

English Mathematician George Boole develop rules for manipulation of binary variables.

- Variable - an arbitrary element of a Boolean Algebra is known as Variable. It can have either value 0 or 1.
- Constant – it can have fixed value as 0 or 1
- Complement – is represented with the “bar over the letter,”
- Literal – each occurrence of a variable in Boolean Function in complemented or complemented way is called as a Literal
- Boolean Function- Boolean expressions are constructed by connecting variable, constant in Boolean operations which, altogether describes Boolean function.

Boolean Algebra - Properties

- **Closure Property –**
  - With respect to + Operator – When two binary elements are operated with + operator the result is a unique binary element.
  - With respect to . Operator – When two binary elements are operated with . (dot) operator the result is a unique binary element.
- **Identity Property**
- **Cumulative Property**
- **Distributive Property**
- **Complement Property**
- **Idempotency Property (Sameness)**
- **Absorption Property**
- **Involution Property**

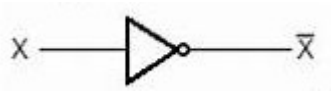
Postulates	(a)	(b)
Postulate 1	Result of each operation is either 0 or 1 $\in$ B	
Postulate 2 (Identity)	$A + 0 = A$	$A \cdot 1 = A$
Postulate 3 (Commutative)	$A + B = B + A$	$AB = BA$
Postulate 4 (Distributive)	$A (B + C) = AB + AC$	$A + BC = (A + B) (A + C$
Postulate 5 (Complement)	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
Theorems	**(a)**	**(b)**
Theorem 1 (Idempotency)	$A + A = A$	$A \cdot A = A$
Theorem 2	$A + 1 = 1$	$A \cdot 0 = 0$
Theorem 3 (Involution)		$\overline{\overline{A}} = A$
Theorem 4 (Absorption)	$A + AB = A$	$A (A + B) = A$
Theorem 5	$A + \overline{A}B = A + B$	$A \cdot (\overline{A} + B) = AB$
Theorem 6 (Associative)	$A + (B + C) = (A + B) + C$	$A (BC) = (AB) C$

Boolean Algebra Postulates/Rules

- Postulates of a mathematical / binary system:
  Minimal set of basic definitions from which all other information can be derived

A1. if $X = 0$ then $X' = 1$ | A1'. if $X = 1$ then $X' = 0$

NOT



A2. $0 \cdot 0 = 0$

A3. $1 \cdot 1 = 1$

A4. $0 \cdot 1 = 1 \cdot 0 = 0$

AND

A2'. $1 + 1 = 1$

A3'. $0 + 0 = 0$

A4'. $0 + 1 = 1 + 0 = 1$

OR

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Useful Properties of Boolean Algebra

Operations with 0 and 1:

1. $X + 0 = X$

1D. $X \cdot 1 = X$

2. $X + 1 = 1$

2D. $X \cdot 0 = 0$

Idempotent Theorem:

3. $X + X = X$

3D. $X \cdot X = X$

Involution Theorem:

4. $(X')' = X$

Theorem of Complementarity:

5. $X + X' = 1$

5D. $X \cdot X' = 0$

Useful Properties of Boolean Algebra

Commutative law:

6. $X + Y = Y + X$ **6D.** $X \bullet Y = Y \bullet X$

Associative law:

7. $(X + Y) + Z = X + (Y + Z) = X + Y + Z$ **7D.** $(X\,Y)Z = X(Y\,Z) = X\,Y\,Z$

Distributive law:

8. $X(Y + Z) = XY + XZ$ **8D.** $X + YZ = (X + Y)(X + Z)$

Simplification theorems:

9. $XY + XY' = X$ **9D.** $(X + Y)(X + Y') = X$

10. $X + XY = X$ **10D.** $X(X + Y) = X$

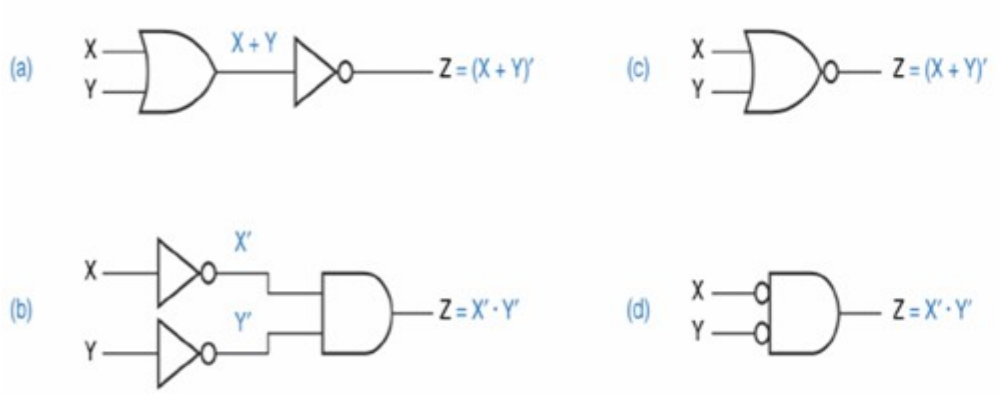
11. $(X + Y')Y = XY$ **11D.** $XY' + Y = X + Y$

DeMorgan's Theorem 1

- $\overline{A + B} = \overline{A} \cdot \overline{B}$

A	B	A + B	$\overline{A + B}$		$\overline{A}$	$\overline{B}$	$\overline{A} \times \overline{B}$
0	0	0	1		1	1	1
0	1	1	0		1	0	0
1	0	1	0		0	1	0
1	1	1	0		0	0	0

EQUAL



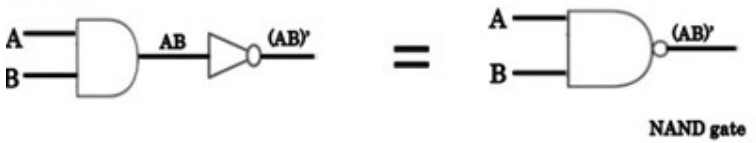
NOR = Bubbled AND

DeMorgan's Theorem 2

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

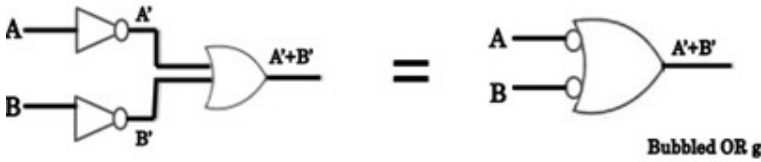
Law 2: (AB)' = A'+B'

LHS



NAND gate

RHS



Bubbled OR gate

A	B	AB	(AB)'
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

=

A	B	A'	B'	A'+B'
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

A	B	A • B	$\overline{A \bullet B}$		$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	0	1		1	1	1
0	1	0	1		1	0	1
1	0	0	1		0	1	1
1	1	1	0		0	0	0

EQUAL

NAND = Bubbled OR

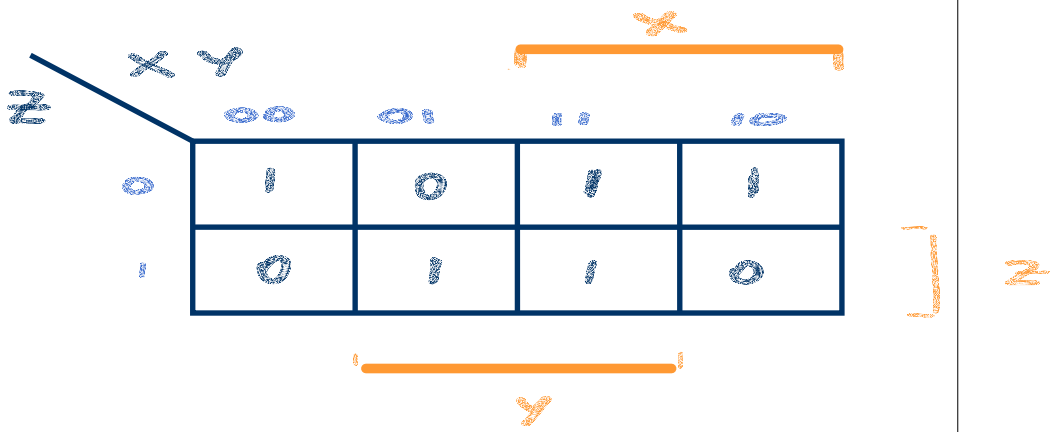
MIT-WPU

।। विश्वशान्तिर्ध्रुवं ध्रुवा ।।

SOP & POS

Representations of Logic Functions

- **Truth Tables**
  Table that shows the output value for each combination of the inputs. The truth table for an n-variable (n-inputs) logic function has 2^n rows.
- **Algebraic expressions**
- **Karnaugh Maps**



Row	X	Y	Z	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

$$F = x \cdot \bar{z} + \bar{y} \cdot \bar{z} + y \cdot z$$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Common Terminology

- **Literal**
  The appearance of a variable or its complement
- **Product Term**
  One or more literals connected by AND operators
- **Sum Term**
  One or more literals connected by OR operators

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Common Terminology(cnt'd)

Standard form of Boolean Expressions

1. **Sum-of-Products (SOP) -** is called a '**minterm**'
    a. first the product (AND) terms are formed then these are summed (OR)
    b. eg: ABC + DEF + GHI
2. **Product-of-sum form (POS) -** is called a '**maxterm**'
    a. first the sum (OR) terms are formed then the products are taken (AND)
    b. eg: (A+B+C) (D+E+F) (G+H+I)
3. **Canonical form**
    a. Canonical form is not efficient but sometimes useful in analysis and design
    b. In an expression in canonical form, every variable appears in every term
    c. eg: f(A, B,C,D) = ABCD +ABCD +ABCD

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Values of logic variables and logic functions are in binary form(1 and 0).

Any logic function can be expressed in the forms of

1. **Sum of Products form (SOP)**

   Y= AB + AC' + BC----------SOP

   Product terms: AB, AC', BC [AND Gate Z = XY]

   Here Summation Terms : 2 terms [OR Gate Z = X + Y]

   ORing of ANDs

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Values of logic variables and logic functions are in binary form(1 and 0).

Any logic function can be expressed in the forms of

2. Product of Sums form (POS)

Y= (A +C)(B + C')----------POS

Sum Terms: A+C , B+C' [OR Gate Z = X + Y]

Here Product Terms: 1 [AND Gate Z = XY]

ANDing of ORs

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Standard Representations for Logic Functions

Canonical Normal Form

- **Standard form for Boolean expression** that provide a **unique signature**
- If **each term** in SOP and POS forms contains **all the literals** then these are known as **canonical SOP and POS** respectively.
- Each **individual term** in **canonical SOP** form is called as **minterm (mo, m1, m2……)**
- Each **individual term** in **canonical POS** form is called as as **maxterm (M0, M1, M2……)**

Canonical or Conventional form

- Consider the following expression

$$W = A\overline{B}C + \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}\overline{B}C$$

Conventional Form of SOP

$$W=f(A,B,C)= A\overline{B}C + \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}\overline{B}C$$

Summation Form of Expression --SOP

- In this case we substitute a "1" for all positive variables & a "0" for all inverted
- Ex: A = 1 and A' or $\bar{A}$ = 0 ; B=1 and B' = 0 and C=1 and C' = 0

- E.g. $F(A,B,C) = A\bar{B}C + \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$ ← Minterms

$$= 101 + 000 + 010 + 001$$
$$= 5 + 0 + 2 + 1$$
$$= \sum m(5,0,2,1)$$
$$= \sum m(0,1,2,5)$$

A	B	C	Decimal Value	Minterm
0	0	0	0	m0
0	0	1	1	m1
0	1	0	2	m2
0	1	1	3	m3
1	0	0	4	m4
1	0	1	5	m5
1	1	0	6	m5
1	1	1	7	m6

Canonical Form and Summation form

Conversion Process – Canonical form to summation form

- **It takes 3 steps:**

1. Each **minterm is converted into it's binary equivalent.**

$$\text{E.g. } F(A,B,C) = A\overline{B}C + \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}\overline{B}C$$
$$= 101 + 000 + 010 + 001$$

2. **Each minterm is converted from binary to decimal.**

$$= 101 + 000 + 010 + 001$$
$$= 5 + 0 + 2 + 1$$
$$= \sum m(5,0,2,1)$$

3. **Finally the min terms are placed in ascending order**

$$= \sum m(0,1,2,5)$$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Advantages of Summation form

- Useful for implementation in Programmable Logic
- Very long & complex expressions can be easily described
- Useful for implementation with hardware devices

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Conversion from Conventional to Summation Form

- E.g. $X = f(A,B,C,D) = \overline{A}CD + AB\overline{C} + \overline{C}\overline{D}$

Conversion from Conventional to Summation Form

- E.g. $X = f(A,B,C,D) = \bar{A}CD + AB\bar{C} + \bar{C}\bar{D}$

$$= \bar{A}CD(B+\bar{B}) + AB\bar{C}(D+\bar{D}) + \bar{C}\bar{D}\,(A+\bar{A})(B+\bar{B})$$

$$= \bar{A}BCD + \bar{A}\bar{B}CD + AB\bar{C}D + AB\bar{C}\bar{D} + \bar{C}\bar{D}(\bar{A}\bar{B}+A\bar{B}+\bar{A}B+AB)$$

$$= \bar{A}BCD + \bar{A}\bar{B}CD + AB\bar{C}D + AB\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + AB\bar{C}\bar{D}$$

$$= \sum m\,(7,3,13,12,0,8,4,12)$$

$$= \sum m\,(0,3,4,7,8,12,13)$$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Conversion from Conventional to Summation Form

- E.g. $X = f(A,B,C,D) = \bar{A}CD + AB\bar{C} + \bar{C}\bar{D}$

$$= \bar{A}CD(B+\bar{B}) + AB\bar{C}(D+\bar{D}) + \bar{C}\bar{D}\,(A+\bar{A})(B+\bar{B})$$

$$= \bar{A}BCD + \bar{A}\bar{B}CD + AB\bar{C}D + AB\bar{C}\bar{D} +$$

$$\bar{C}\bar{D}(\bar{A}\bar{B}+A\bar{B}+\bar{A}B+AB)$$

$$= \bar{A}BCD + \bar{A}\bar{B}CD + AB\bar{C}D + AB\bar{C}\bar{D} +$$

$$\bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + AB\bar{C}\bar{D}$$

$$= \sum m\,(7,3,13,12,0,8,4,12)$$

$$= \sum m\,(0,3,4,7,8,12,13)$$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

- **Consider Following Truth Table**

	A	B	C	F	
m0 0	0	0	0	0	← Low
m1 1	0	0	1	0	
m2 2	0	1	0	1	← High
m3 3	0	1	1	0	
m4 4	1	0	0	1	
m5 5	1	0	1	1	
m6 6	1	1	0	1	
m7 7	1	1	1	1	

$F = \bar{A}.B.\bar{C}$

$F = A.\bar{B}.\bar{C}$

$F = A.B.\bar{C}$

$F = A.B.\bar{C}$

$F = A.B.C$

In SOP Form always remember

$1 = A$

$0 = \bar{A}$

$F = \bar{A}B\bar{C} + A\bar{B}\bar{C} + \bar{A}BC + AB\bar{C} + ABC$

Practice Examples(Contd..)

{Law of distribution: A+BC = (A+B)(A+C)}
so B' + AB = (B' +A) (B' + B) = B' + A}

- $F(A,B,C) = \sum m\,(0,4,6,1,5,7)$

$$F = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + \bar{A}\bar{B}C + A\bar{B}C + ABC$$

$$= \bar{B}\bar{C}(\bar{A}+A) + AB\,(\bar{C}+C) + \bar{B}C(\bar{A}+A)$$

$$= \bar{B}\bar{C}(1) + AB(1) + \bar{B}C(1)$$

$$= \bar{B}\bar{C} + AB + \bar{B}C$$

$$= \bar{B}(\bar{C}+C) + AB$$

$$= \bar{B} + AB$$

$$= \bar{B} + A$$

A	B	C	Decimal Value	Minterm
0	0	0	0	m0
0	0	1	1	m1
0	1	0	2	m2
0	1	1	3	m3
1	0	0	4	m4
1	0	1	5	m5
1	1	0	6	m5
1	1	1	7	m6

Practice Examples(Contd..)
010, 100, 101, 110, 111

- $F(A,B,C)= \sum m\ (2,4,5,6,7)$

Practice Examples(Contd..) A'X +A = (A+A')(A+X)= A+X = A + BC'

- F(A,B,C)= $\sum$m (2,4,5,6,7)

$$F = \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} + ABC$$

(m2, m4, m5, m6, m7)

$$= \overline{A}B\overline{C} + A\overline{B}(\overline{C}+C) + AB\,(\overline{C}+C)$$

$$= \overline{A}B\overline{C} + A\overline{B}(1) + AB(1)$$

$$= \overline{A}B\overline{C} + A\overline{B} + AB$$

$$= \overline{A}B\overline{C} + A(\overline{B}+B)$$

$$= \overline{A}B\overline{C} + A$$

If $B\overline{C} = X$

$$= X + A$$

If $\overline{A}X = X$

$$= A + X$$

$$= A + B\overline{C}$$

Practice Examples(Contd..)

- Simplify the Expression

$Y(AB) = \sum m(0,2,3)$

Practice Examples(Contd..)

m0 = A'B'
m1 = A'B
m2 = AB'
m3 = AB

- Simplify the Expression

$$Y(AB) = \sum m(0,2,3)$$

$$= m0 + m2 + m3$$

$$= \bar{A}\bar{B} + A\bar{B} + AB$$

$$= (\bar{A}+A)\bar{B} + AB$$

$$= \bar{B} + AB \leftarrow \quad \bar{A} + AB = \bar{A} + B$$

$$= \bar{B} + A$$

A	B	Y= $\bar{B}$ + A
0	0	
0	1	
1	0	
1	1	

Product of Sum (POS)

- It is a Conjunctive Normal form or Maxterm Expansion
- E.g.

  (A+B) (C+D)

- (A OR B) AND (C OR D)

i.e. AND of OR terms

POS Contd..

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

A+B+C

A+B+C'

A+B'+C'

In POS Form always remember

$1 = \overline{A}$

$0 = A$

SOP	POS
$0 \rightarrow \overline{A}$	$0 \rightarrow A$
$1 \rightarrow A$	$1 \rightarrow \overline{A}$

POS (Contd..)

- Why 0 = A & 1=$\bar{A}$
- E.g. $Y = (A+B+C)\,(A+B+\bar{C})\,(A+\bar{B}+\bar{C})$

$$\bar{Y} = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC$$

$$\overline{\bar{Y}} = \overline{[\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC]} \leftarrow$$ **Taking complement of both sides**

$$Y = (\bar{A}\bar{B}\bar{C})' \,.\, (\bar{A}\bar{B}C)' \,.\, (\bar{A}BC)'$$ ← **Using De-Morgan's Law (A+B) = A'. B'**

$$Y = (A+B+C).(A+B+\bar{C}).(A+\bar{B}+\bar{C})$$ ← **Again by Using De-Morgan's Law**

But it is complicated to use demorgan's law every time, so it will be easy to use following convention

0 → A

1 → $\bar{A}$

POS (Contd..)

- $Y = (A+B+C)\ (A+B+\overline{C})\ (A+\overline{B}+\overline{C})$

$= (X + C\overline{C})\ (A+\overline{B}+\overline{C})$ ←

From Distributive law
$(X + C)\ (X + \overline{C}) = X + C\overline{C}$

$= (X+0)\ (A+\overline{B}+\overline{C})$

$= X\ .\ (A+\overline{B}+\overline{C})$

$= (A+B)\ (A+\overline{B}+\overline{C})$

$= A+B(\overline{B}+\overline{C})$

$= A + B\overline{B} + B\overline{C}$

$= A + B\overline{C}$ ←

From Distributive law
$A + BC = (A + B)\ (A + C)$

$= (A+B)\ (A+\overline{C})$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Practice Example

- For given truth table minimize POS expression

	A	B	Y
M0	0	0	1
M1	0	1	0
M2	1	0	1
M3	1	1	0

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Practice Example

- For given truth table minimize POS expression

	A	B	Y
M0	0	0	1
M1	0	1	0
M2	1	0	1
M3	1	1	0

$$Y = (A + \overline{B})\,(\overline{A} + \overline{B})$$
$$= \overline{B} + A\overline{A}$$
$$= \overline{B} + 0$$
$$Y = \overline{B}$$

From Distributive law
(A + B) (A + C) = A + BC

Practice Example (Contd..)

- **POS Representation**

Y = (M1, M3)

Y= M(1,3)

- **SOP Representation**

Y=∑m(0,2)

Practice Example (Contd..)

- For following truth table find minterm & maxterm i.e. SOP & POS forms

	A	B	C	F
m0 0	0	0	0	1
M1 1	0	0	1	0
m2 2	0	1	0	1
m3 3	0	1	1	1
M4 4	1	0	0	0
M5 5	1	0	1	0
m6 6	1	1	0	1
m7 7	1	1	1	1

Practice Example (Contd..)

- For following truth table find summation & product forms i.e. SOP & POS form

	A	B	C	F
m0 0	0	0	0	1
M1 1	0	0	1	0
m2 2	0	1	0	1
m3 3	0	1	1	1
M4 4	1	0	0	0
M5 5	1	0	1	0
m6 6	1	1	0	1
m7 7	1	1	1	1

$Y(A,B,C) = \sum m(0,2,3,6,7)$

$Y(A,B,C) = \pi M(1,4,5)$

Practice Example (Contd..)

- $Y(A, B, C) = \sum m(0,2,3,6,7)$

Practice Example (Contd..)

- $Y(A, B, C) = \sum m(0,2,3,6,7)$

$$
\begin{aligned}
Y &= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C} + ABC \\
&= \bar{A}\bar{B}\bar{C} + \bar{A}B(\bar{C}+C) + AB(\bar{C}+C) \\
&= \bar{A}\bar{B}\bar{C} + \bar{A}B + AB \\
&= \bar{A}\bar{B}\bar{C} + (\bar{A}+A)B \\
&= \bar{A}\bar{B}\bar{C} + B \\
&= X.\bar{B} + B \qquad \text{Let } X = \bar{A}\bar{C} \\
&= X + B \\
&= \bar{A}\bar{C} + B
\end{aligned}
$$

Practice Example (Contd..)

- $Y(A,B,C) = \pi M(1,4,5)$

Practice Example (Contd..)

- $Y(A,B,C) = \pi M(1,4,5)$

  $\pi M(001,100,101)$

  $= (A+B+C')(A'+B+C)(A'+B+C')$

  Consider A'+B = X and C = Y and C'=Z

$= (A+B+C')(A' + B + CC')$-------$(A+B)(A+C)=A+BC$—Distributive Law

$= (A + B + C')(A' + B)$

$= B + (A+C')A'$

$=B + A.A' + A'C'$

$= B+ A'C'$

$=(B+A')(B+C')$

$Y =(A'+B)(B+C')$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Karnaugh Map
K-MAP

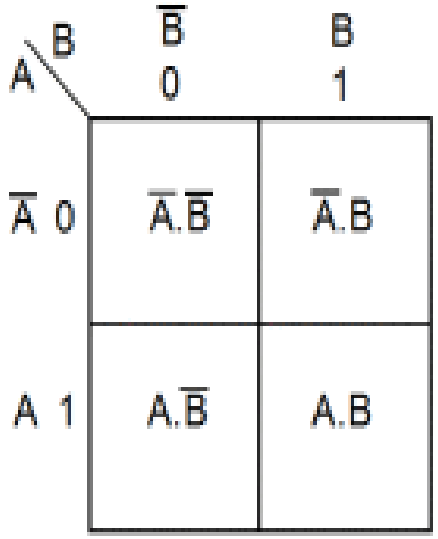
MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Introduction

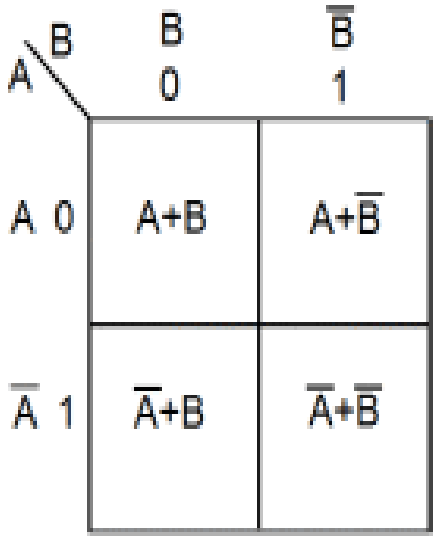
- **Expression reduction through basic rules of Boolean algebra is time consuming for higher variables.**
- **Solution is K-Map - graphical method**
- **A Karnaugh map (K-Map) is a pictorial method used to minimize Boolean expression without having to use Boolean algebra theorems & equation manipulation**
- **It is a special version of truth table**
- **Using K-Map, expressions with 2 to 4 variables are easily minimized**

2 – Variables K- Map

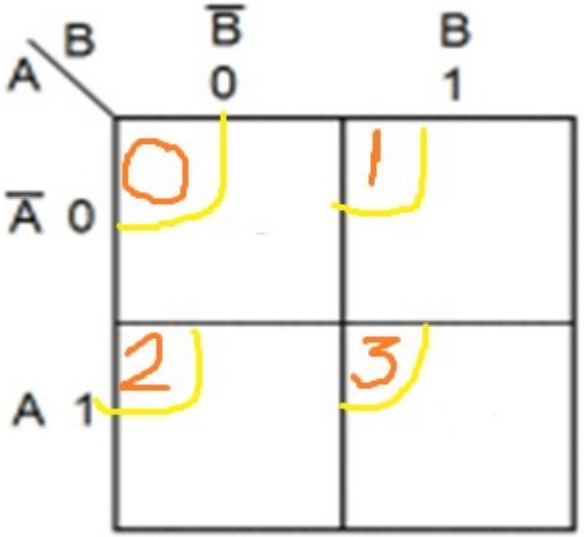
A. SOP: -



B. POS: -



A. SOP: -



3 – Variables K- Map

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

A \ BC	00	01	11	10
0	A’B’C’	A’B’C	A’BC	A’BC’
1	AB’C’	AB’C	ABC	ABC’

SOP Form

A \ BC	00	01	11	10
0	A+B+C	A+B+C’	A+B’+C’	A+B’+C
1	A’+B+C	A’+B+C’	A’+B’+C’	A’+B’+C

POS Form

4 – Variables K- Map

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

AB \ CD	00	01	11	10
00	A’B’C’D’	A’B’C’D	A’B’CD	A’B’CD’
01	A’BC’D’	A’BC’D	A’BCD	A’BCD’
11	ABC’D’	ABC’D	ABCD	ABCD’
10	AB’C’D’	AB’C’D	AB’CD	AB’CD’

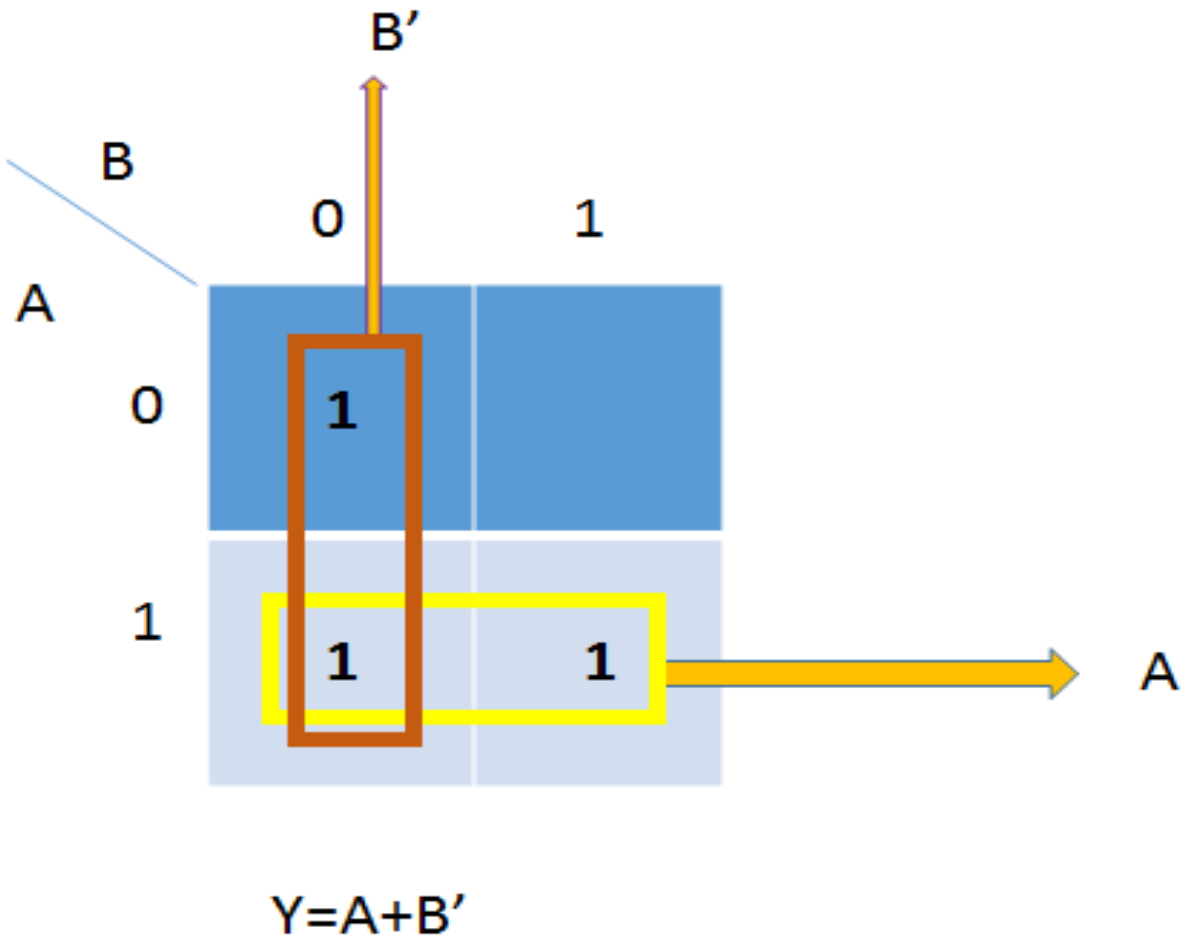
SOP Form

AB \ CD	00	01	11	10
00	A+B+C+D	A+B+C+D’	A+B+C’+D’	A+B+C’+D
01	A+B’+C+D	A+B’+C+D’	A+B’+C’+D’	A+B’+C’+D
11	A’+B’+C+D	A’+B’+C+D’	A’+B’+C’+D’	A’+B’+C’+D
10	A’+B+C+D	A’+B+C+D’	A’+B+C’+D’	A’+B+C’+D

POS Form

Practice Examples

m0	m1
m2	m3



$$Y(AB) = \sum m(0,2,3)$$

$$= m0 + m2 + m3$$

$$= \bar{A}\bar{B} + A\bar{B} + AB$$

$$= (\bar{A}+A)\bar{B} + AB$$

$$= \bar{B} + AB \leftarrow \bar{A} + AB = \bar{A} + B$$

$$= \bar{B} + A$$

Practice Examples(Contd..)

- $F(A,B,C) = \sum m\,(0,4,6,1,5,7)$

$F = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + \bar{A}\bar{B}C + A\bar{B}C + ABC$

$= \bar{B}\bar{C}(\bar{A}+A) + AB\,(\bar{C}+C) + \bar{B}C(\bar{A}+A)$

$= \bar{B}\bar{C}(1) + AB(1) + \bar{B}C(1)$

$= \bar{B}\bar{C} + AB + \bar{B}C$

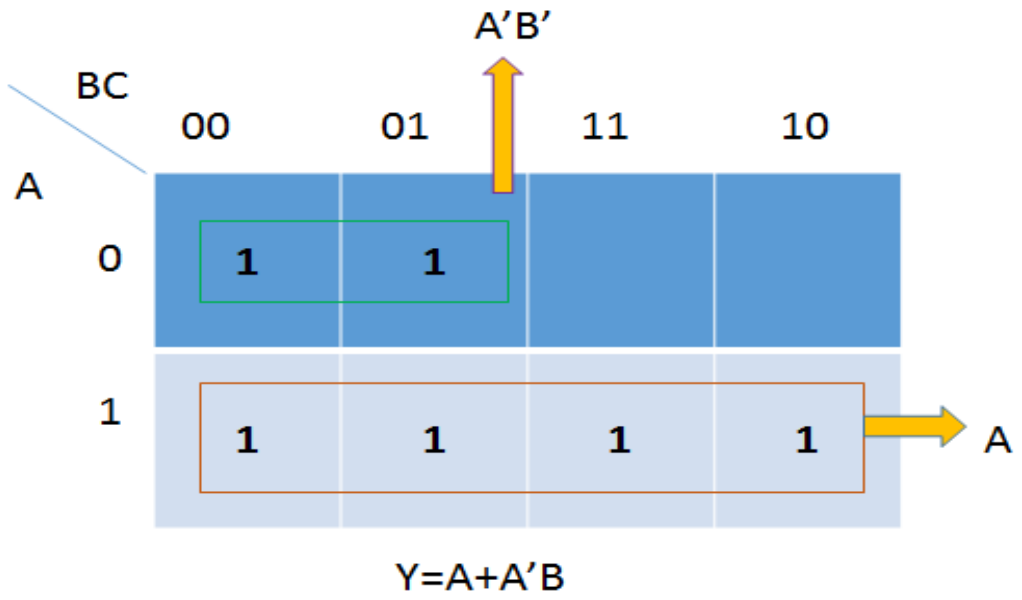
$= \bar{B}(\bar{C}+C) + AB$

$= \bar{B} + AB$

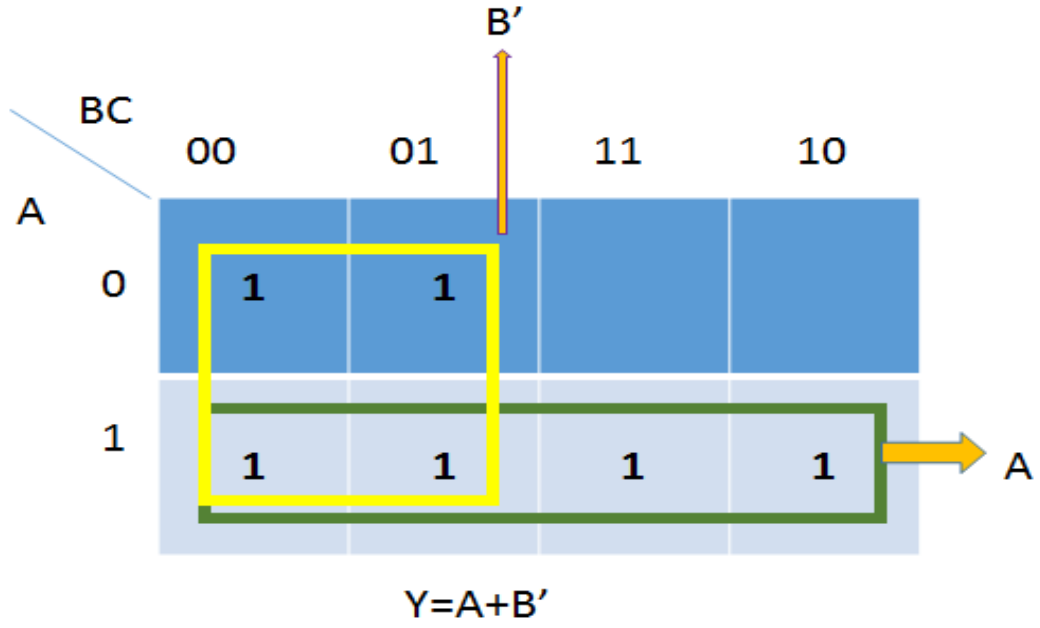
$= \bar{B} + A$

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

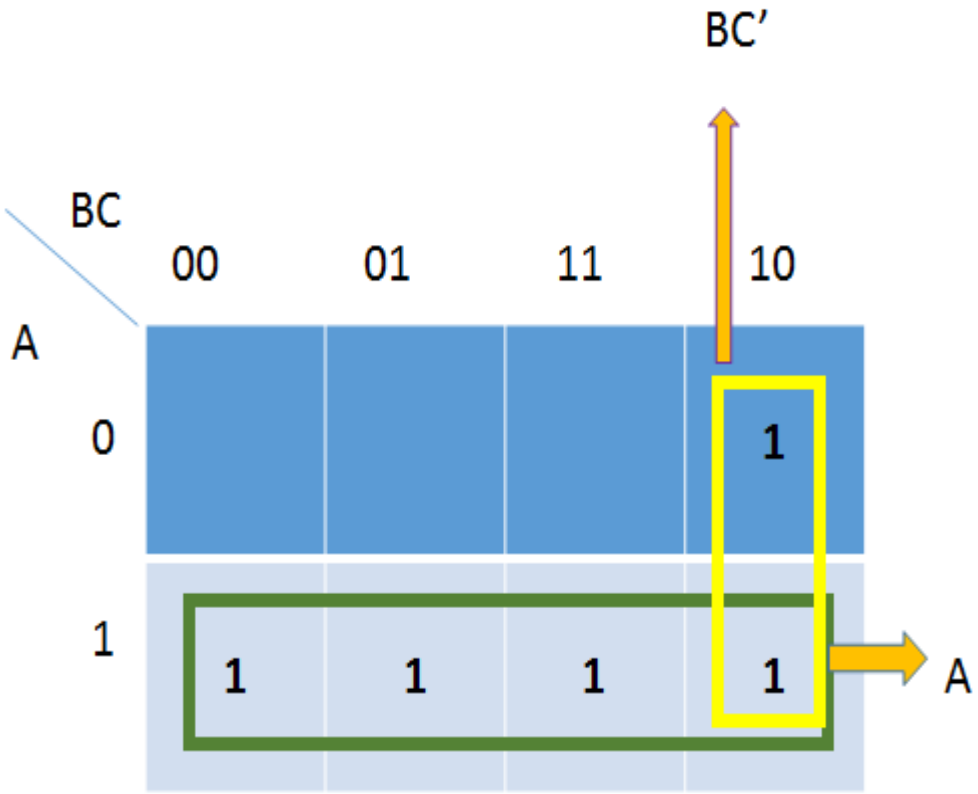
Way 1



Way 2



Practice example



Y=A+B
C'

- **F(A,B,C)= ∑m (2,4,5,6,7)**

$$F = \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$
(m2, m4, m5, m6, m7)

$$= \bar{A}B\bar{C} + A\bar{B}(\bar{C}+C) + AB(\bar{C}+C)$$
$$= \bar{A}B\bar{C} + A\bar{B}(1) + AB(1)$$
$$= \bar{A}B\bar{C} + A\bar{B} + AB$$
$$= \bar{A}B\bar{C} + A(\bar{B}+B)$$
$$= \bar{A}B\bar{C} + A \quad \text{If } B\bar{C} = X$$
$$= X + A \quad \text{If } \bar{A}X = X$$
$$= A + X$$

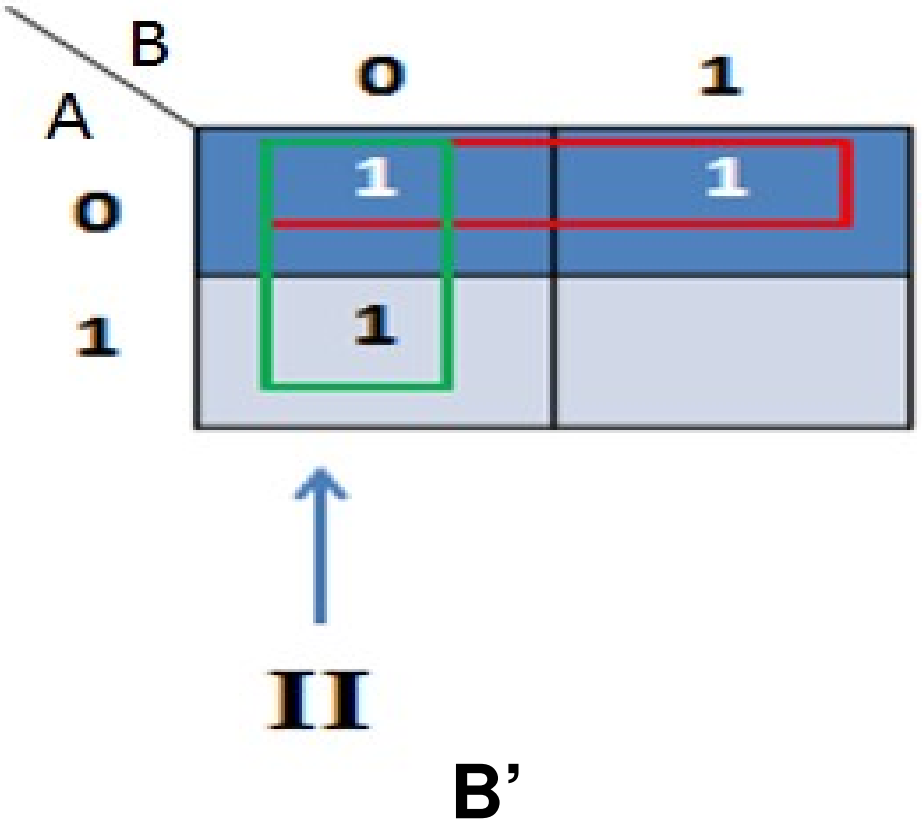
F = A + BC'

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Practice Example

$$F(A,B) = \bar{A}\bar{B} + A\bar{B} + \bar{A}B$$

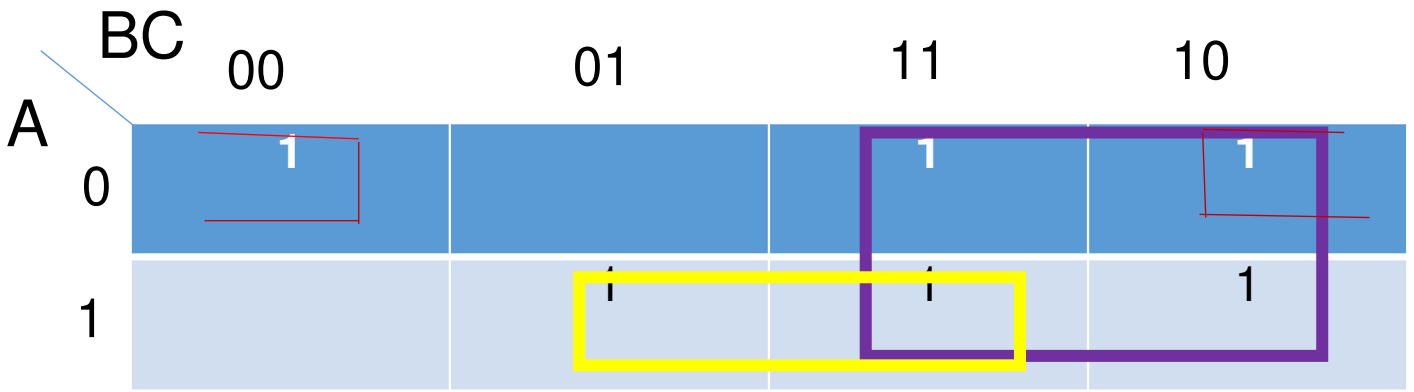
Minterms: 0,2,1



$$Y = A' + B'$$

Example (Contd..)

- $Z=f(A,B,C)=\bar{A}\bar{B}\bar{C}+\bar{A}B\bar{C}+AB\bar{C}+\bar{A}BC+ABC+A\bar{B}C$



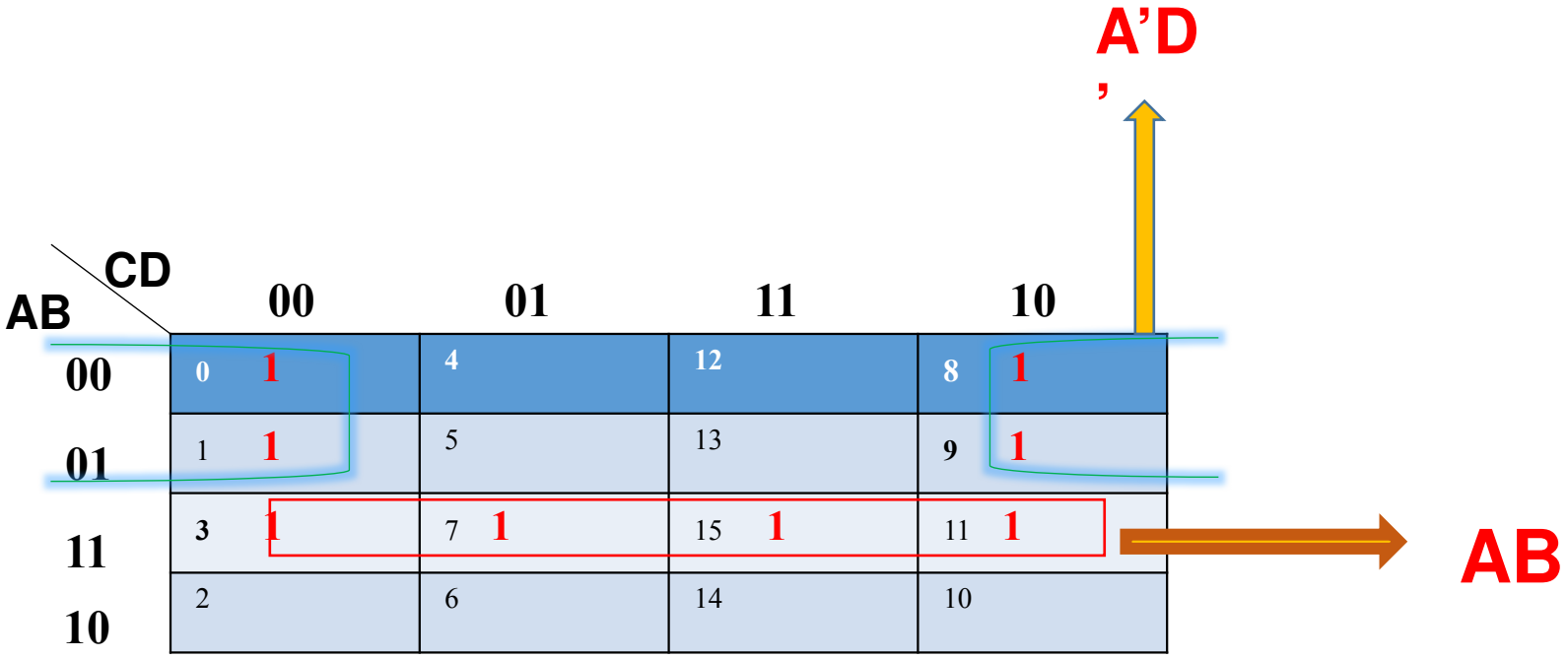
$$Z = A'C' + AC + B$$

Minterms: 0,2,6,3,7,5

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

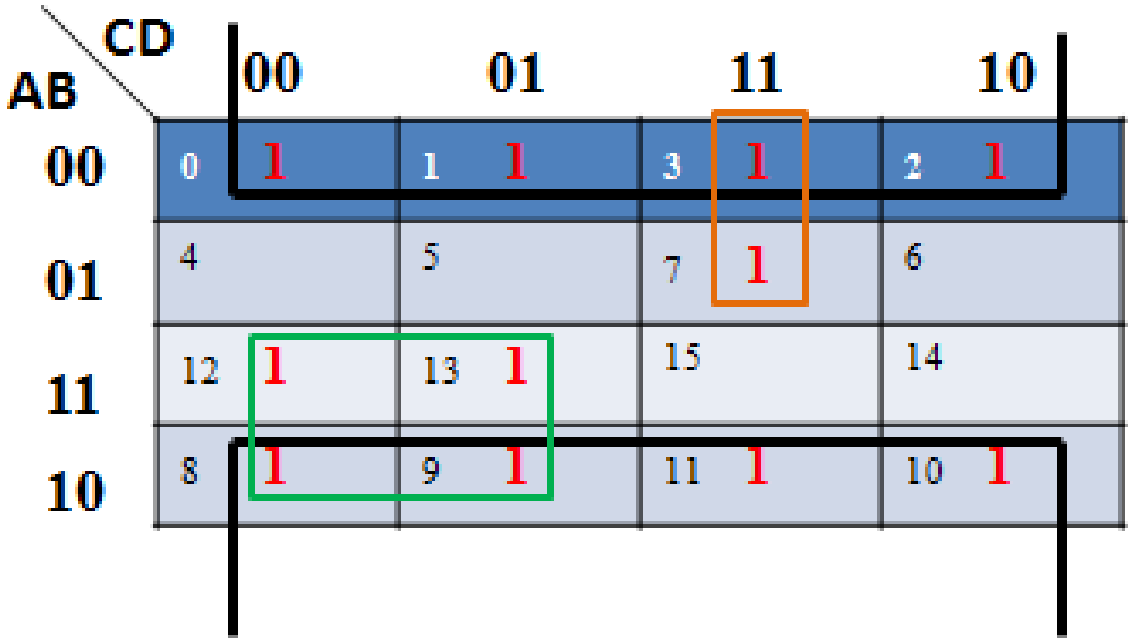
MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Four Adjacent ones



$$Y = A'D' + AB$$

Practice Example



$$Y = \overline{B} + A\overline{C} + \overline{A}CD$$

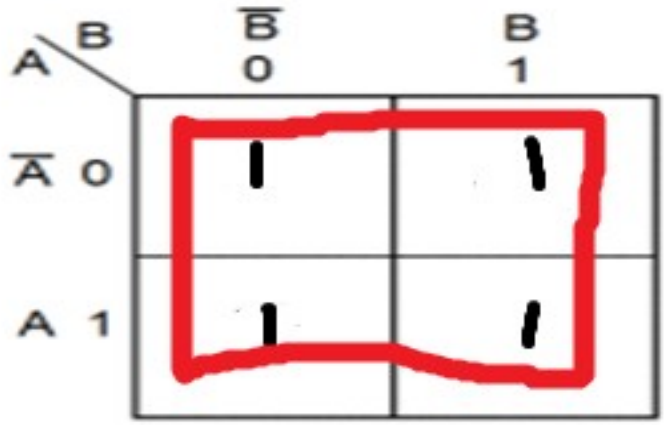
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Example

What is the solution of k map if all entries are 1?. We want SOP form solution.

SOP: -

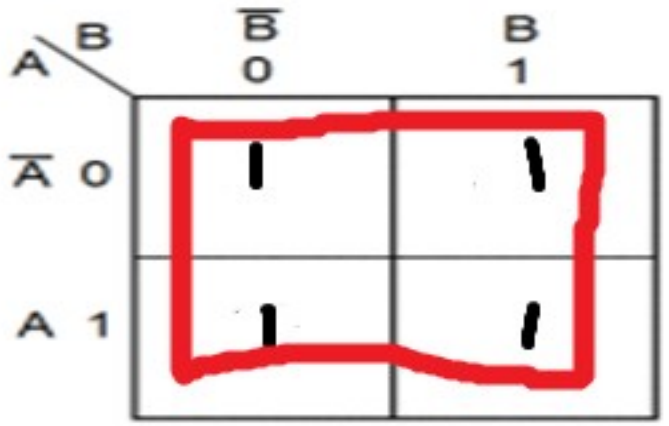


What is the solution of k map if all entries are 1?. We want SOP form solution.

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

SOP: -



$$Y(A,B) = A'B' + A'B + AB' + AB$$

$$Y = A'(B'+B) + A\,B'+B)$$

$$Y = A' + A$$

$$Y = 1$$

Folding K map in horizontal direction

$Y = \sum m(0,2,3,6)$

A \ BC	B'C'	B'C	BC	BC'
A'	1	0	1	1
A	0	0	0	1

$Y = BC' + A'B + A'C'$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Folding K map only in horizontal direction

Y = ∑m(0,1,3,6,10,11 14)

AB \ CD	_00	_01	11	10
_00	1	0	0	1
_01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

Y=A'B'D' + AB'D' + BD

Folding K map only in Vertical direction

Y = ∑m(0,1,3,6,10,11 14)

AB \ CD	_00	_01	11	10
_00	1	0	0	1
_01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

Y=B'C'D' + B'CD' + BD

Folding K map in both horizontal as well as vertical direction

$Y = \sum m(0,1,3,6,10,11\ 14)$

AB \ CD	_00	_01	11	10
_00	1	0	0	1
_01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

$Y = B'D' + BD$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

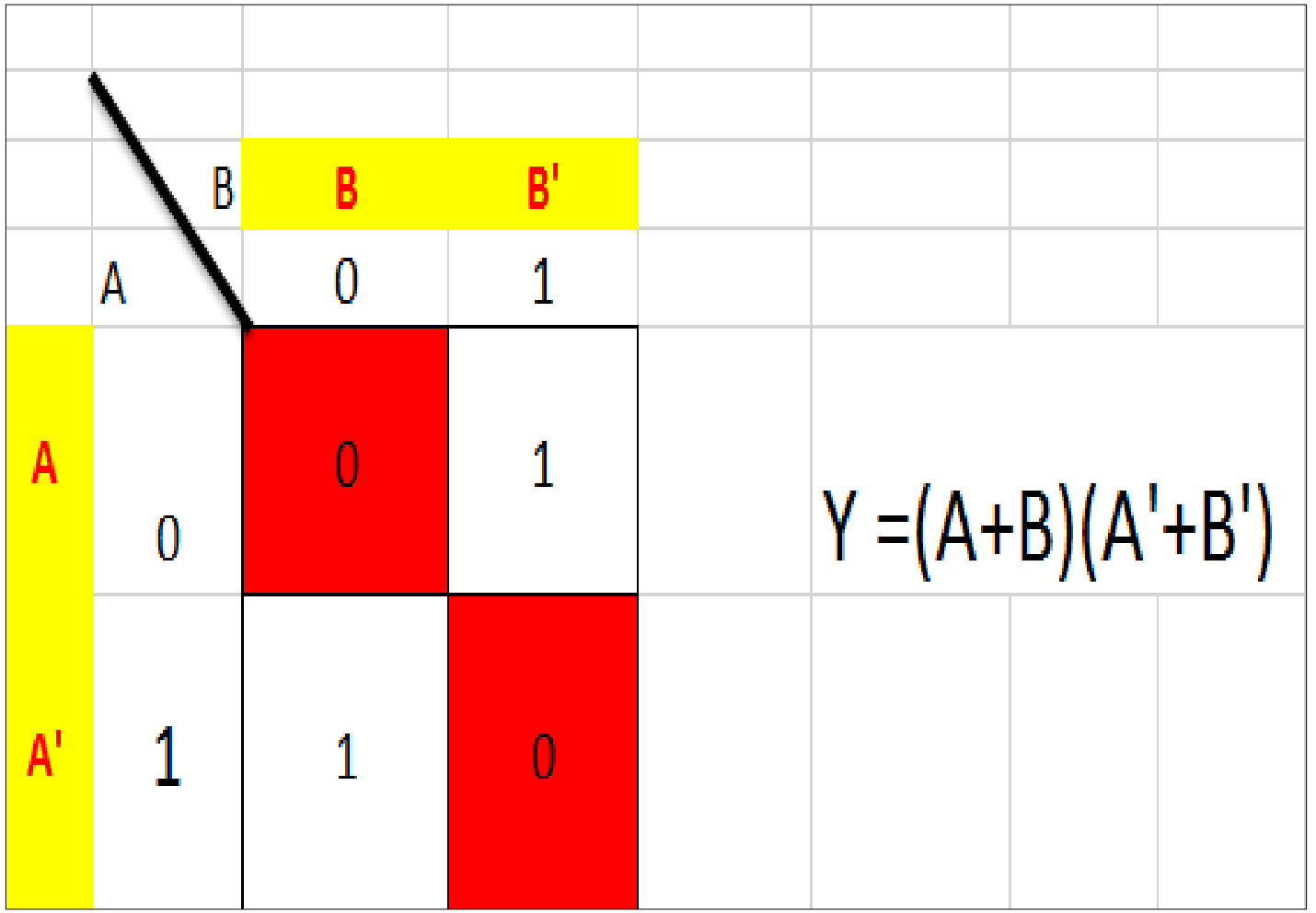
Folding K map - AN EXAMPLE

$Y = \sum m(3, 5, 8, 11, 12, 13, 14)$

AB \ CD	_00	_01	11	10
_00	0	0	1	0
_01	0	1	0	0
11	1	1	0	1
10	1	0	1	0

$Y = AC'D' + BC'D + B'CD + ABD'$

K map for POS form Boolean expressions



POS form – k map simplification

$Y = \pi\ M(0,2,3,6)$

A \ BC	00	01	11	10
	B+C	B+C'	B'+C'	B'+C
0 A	0	1	0	0
1 A'	1	1	1	0

$Y=(A+C)(B'+C)(A+B')$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

POS form – k map simplification

$Y = \pi M(0,1,3,6,10,11\ 14)$

AB \ CD	00 C+D	01 C+D'	11 C'+D'	10 C'+D
00 A+B	0	0	0	1
01 A+B'	1	1	1	0
11 A'+B'	1	1	1	0
10 A'+B	1	1	0	0

$Y=(A+B+C)(A+B+D')(B'+C'+D)(A'+B+C')$

POS form – k map simplification

$Y = \pi M(1,2,4,5)$

		00	01	11	10
		B+C	B+C'	B'+C'	B'+C
0	A	1	0	1	0
1	A'	0	0	1	1

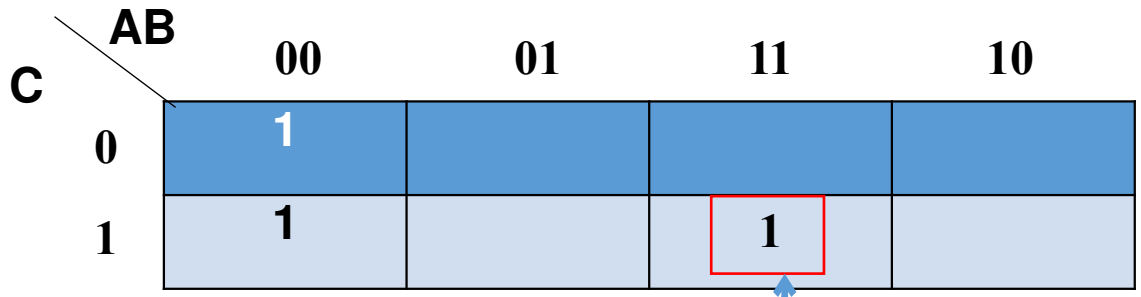
$Y=(B+C')(A'+B)(A+B'+C)$

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

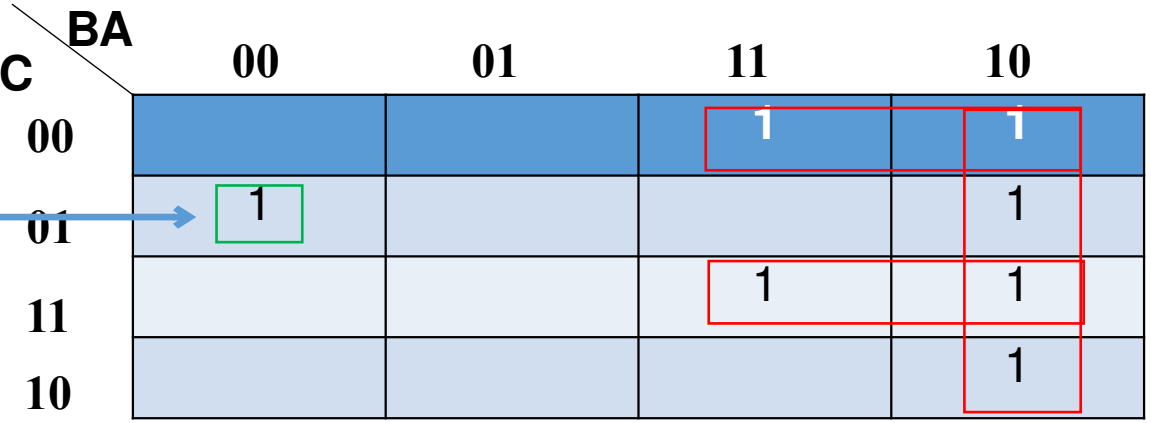
Prime Implicant

- **The process of simplification involves grouping of Minterms & identifying Prime Implicants (PI) & Essential Prime Implicants (EPI)**
- **Prime implicant is a group of ones (minterms) that cannot be combined with any other ones (minterms) or groups**
- **EPI is PI, in which one or more minterms are unique**
  - i.e. it contain atleast one minterm which is not contained in any other PI

Prime Implicant (Contd..)



Prime Implicant



MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

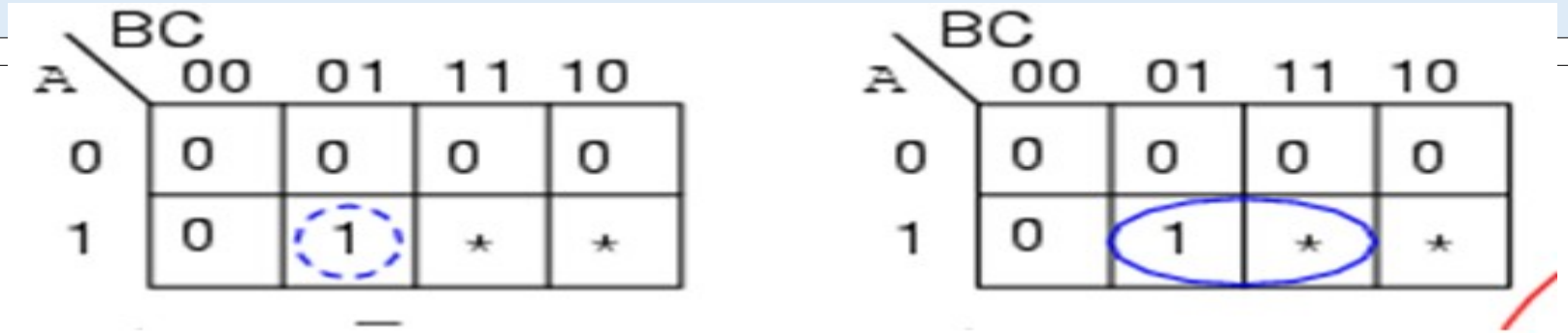
Don't Care Condition (X)

- The K-Maps are simplified using either 1's or 0's, therefore we make the entries in the map for either 0 or 1
- The cell which do not contain 1 are assumed to contain 0 & vice versa
- This is not always true, since there are cases in which certain combinations of I/P variables do not matter

Don’t Care Condition (X)

- In such a situation the designer has a flexibility & it is left to him whether to assume 0 or 1 as O/P for each of these combinations
- This condition is known as Don’t Care Condition and can be represented on K-Map as a X mark in the corresponding cell
- The X mark in a cell may be assumed to be a 1 or 0 depending upon which one leads to a simpler expression

Don’t Care Condition (X)

- 

Without Considering don’t care conditions.

Y= AB’C

Assuming don’t care as 1.

Y=AC

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Example

- $f(A,B,C,D) = \sum m(1,3,7,11,15) + d(0,2,5)$

Example

- $f(A,B,C,D) = \sum m(1,3,7,11,15) + d(0,2,5)$

A'D + CD

Change in position of variable (Important to observe)

Y(A,B,C,D = ∑m(3, 5, 8,11, 12,13,14)

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Y(A,B,C,D = ∑m(3, 5, 8,11, 12,13,14)

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10