Unit-5 # Microprocessor - II

Q. Write and explain instruction set in 8086 microprocessor

Instruction set in 8086 Microprocessor:-

(i) Data Copy / Transfer Instruction
(ii) Arithmetic / Logical Instruction,
(iii) Branch Instruction.
(iv) Loop Instruction.
(v) Machine Control Instruction.
(vi) Shift / Rotate Instruction.
(vii) Flag Manipulation Instruction.
(viii) String Instruction.

The 8086/8088 instructions are categorized into the following main types. This section explains the function of each of the instructions with suitable examples wherever necessary.

(i) Data Copy / Transfer Instructions:-

This type of instructions are used to transfer data from source operand to destination operand. All the store, move, load, exchange, input and output instructions belong to this category.

(ii) Arithmetic and Logical Instructions:-

All the instructions performing arithmetic, logical, increment, decrement, compare and scan instructions belong to this category.

(iii) Branch Instructions:-
These instructions transfer control of execution to the specified address. All the call, jump, interrupt and return instructions belong to this class.

(iv) Loop Instructions:-
If these instructions have REP prefix with CX used as count register, they can be used to implement unconditional and conditional loops. The LOOP, LOOPNZ and LOOPZ instructions belong to this category. These are useful to implement different loop structures.

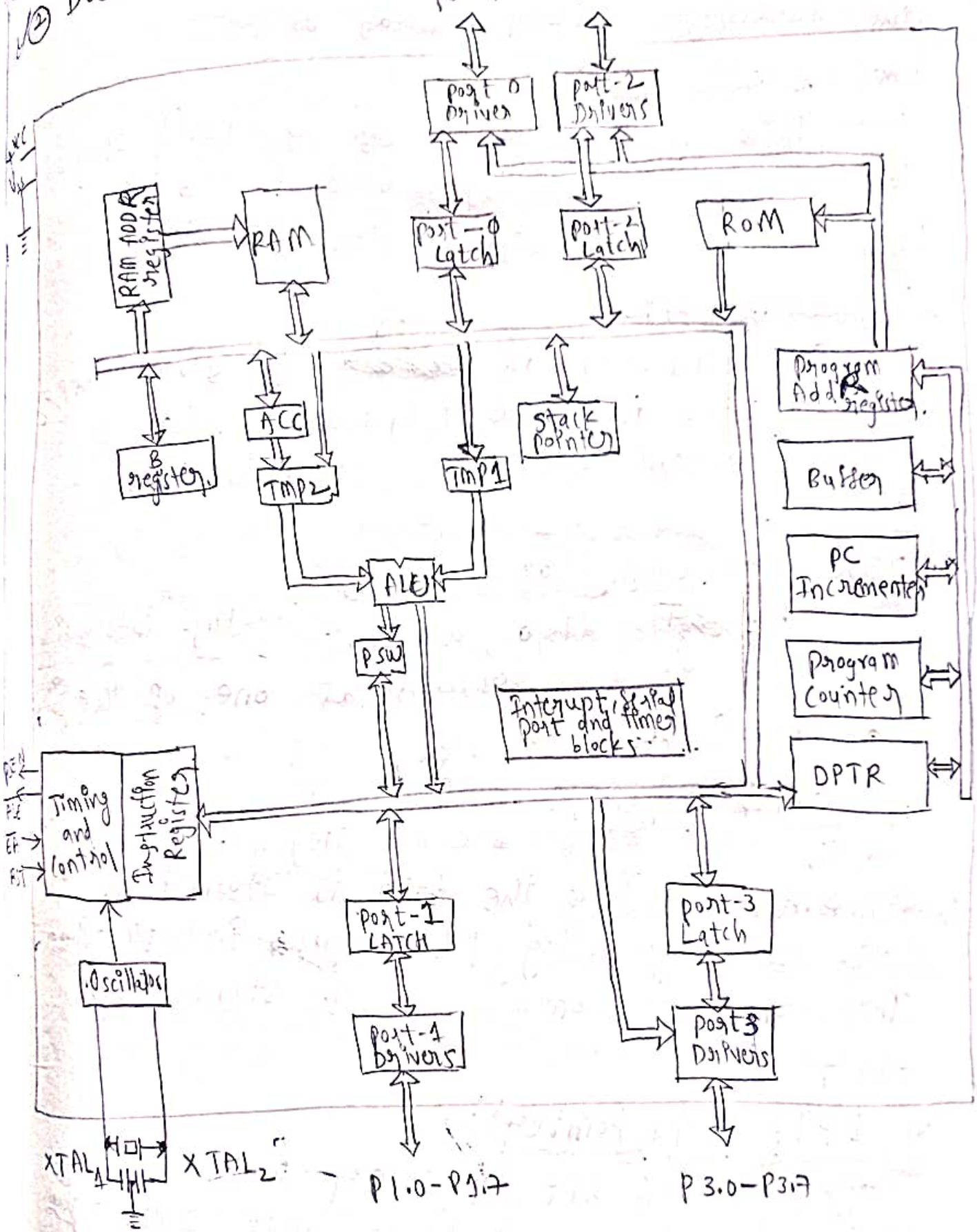
(v) Machine Control Instructions:-
These instructions control the machine status. NOP, HLT, WAIT and LOCK instructions belong to this class.

(vi) Flag Manipulation Instructions:-
All the instructions which directly affect the flag register, come under this group of instructions. Instructions like CLD, STD, CLI, STI, etc. belong to this category of instructi

(vii) Shift and Rotate Instructions:-
These instructions involve the bitwise shifting (or rotation either direction with (or without a count in CX.

(viii) String Instructions:- These instructions involve various string manipulation operations like load, move, scan, compare, store, etc. These instructions are only to be operated upon the strings.

(2) Draw and explain 8051 micro controller ?.



The internal architecture of 8051 is representing the functional description. as follows

(i) ACC [Accumulator]:-

This may be acts as operand register either implicit or explicit. are specified in the instructions This is on chip special function register.

(ii) B Register:-

This register is used to store one of the operands for multiply and division instruction and considered as special function register.

(iii) PSW (Program Status) Word:-

The set of flags contains the status information and is considered as one of the special function registers.

(iv) Stack Pointer:-

This is 8-bit wide register it is incremented before the data is stored on to the stack using push / pop instructions. This register contain 8-bit stack top address.

(v) DPTR (Data Pointer):-

This is 16 bit register contains a higher byte DPH and lower byte DPL of a 16 bit external data RAM address.

(vii) Port '0' to 3 Latches and Drivers:-
These 4 latches & drivers pairs are alloted to each of the 4 on chip IO ports. These latches have been alloted addresses in the special function register. using the allowted addresses. The user can communicate and identified as P_0, P_1, P_2, and P_3

(x) Serial Data Buffer:-
This consisting of (a) Transmit Buffer:- which is necessary a parallel in serial out (PISO) register. (b). Receive Buffer:- This is a serial in parallel out register. (SIPO)

(viii) Timer Register:-
The two 16-bit registers can be accessed as their lower and upper bytes tl0', th0' TL0, TH0 for register '0'.
lly TL1 and TH1 represents lower and higher bytes of timing register '1'.

(ix) Control Register:-
The special function register IP, IE, TMOD, T-COM, S-COM and P-COM contain control and status information for interrupts/. Timers | Counters

(X) Timing and Control Unit

This derives all the necessary timing and control signals required for the internal operation of the circuit. It also derives control signals required for controlling the external system bus.

(XI) Oscillator :-

The circuit generates the basic timing clock signals for the operation of the circuit using the crystal oscillator.

Instruction Register :-

This register decodes the opcode of an instructi to be executed and gives information to the timing and control unit to generate necessary signals for the execution of the instruction.

EPROM and program address Register :-

This provide onchip EPROM and mechanism to internally address it.

RAM and RAM address register :-

This block provide internal 128-bytes RAM. and a mechanism to address it internally.

ALU :-

The Arithmetic and Logic Unit performs 8-bit arithmetic and logical operations over the operands held by the temporary register (temp-1 and temp2 But user can not access these temporary register

SFR Register Bank:-

SFR = Special Function Register.

This is a set of SFRs, which can be addressed using their respective addresses, which lie in the range 80H to FFH.

Finally the interrupt, serial port and timer units control and perform their specific functions under the control of timing and control unit.

Q Write 8051 Instruction set supports addressing modes.

Sol:- (i) Direct Addressing Mode
(ii) Indirect Addressing Mode
(iii) Register Addressing Mode
(iv) Register Specific Addressing Mode
(v) Immediate Addressing Mode
(vi) Indexed Addressing Mode

DIR² I².

Direct Addressing Mode:-

In this the operands are specified using the 8-bit address field in the instruction format only. Internal data RAM and SFRs can directly address.

Ex:- MOV R_0, 89H.

Here 89H is address of a special function register.

T-mode

4 Indirect Addressing Mode:-

In this mode of addressing the 8-bit address of an operand is stored in a register and the 8-bit address is present in a register. This specified in the instruction the register R_0 and R_1 of the selected bank of register or SP can be used as the address registers. for storing the 8-bit address. The address register for 16-bit address can only be data pointer (DPTR

Ex:- ADD A, @ R_0

Register Addressing Mode:-

In this addressing mode operands are stored in the registers $R_0 - R_7$ of the selected register bank one of these 8-registers R_0 to R_7 is specified in the instruction using 3-bit specification field of the opcode format.

Ex:- ADD A, R_3 or MOV A, R_3.

Register Specific Instruction Addressing Mode:-

In this type of instructions the operand implicitly specified. using one of the register

Ex:- RLA (ROL).

This instruction rotates accumulator content left.

Immediate Addressing Mode:-

In this an immediate data that is a constant is specified in the instruction after the opcode byte. Ex:- MOV A, #100H

Indexed Addressing Mode:-

Only program memory accessed using this addressing mode. This addressing mode is used in 8051 for look up table manipulations.

Program counter and DPTR are allowed 16-bit storage registers. These point to the base of lookup table and the accumulator register contents a code to be converted. using look up table..

Ex:- MOV CA, @ A+DPTR

JMP @ A+DPTR

---

(4) Describe Instruction set of 8051

Sol:- (i) Data Copy Instruction (or Data Transfer Instruction

(ii) Arithmetic Instruction

(iii) Logical Instruction

(iv) Branch Instruction

(v) Bit Processing Instruction

Data Copy Instruction:-

Data transfer instructions move the content of one register to another. The register content of which is moved remains unchanged. If they have suffix 'X'

MOV x The data is exchanged with external Memory.

Ex:- MOV A, R_n

MOV A, @Ri

MOV @DPTR, A

MOV @DPTR, A
This indicates accumulator to external RAM 16 bit address.

Arithmetic Instruction:-
This performs several basic operations such as addition, subtraction, division, multiplication etc. After execution of this instructions the result is stored in the first operand.

Eg:- ADD A, R_1
The result of addition $A + R_1$ will be stored in the accumulator.

Eg:- ADD A, #DATA

INCA .
INC R_n

Logical Instruction:-
These performs logic operations upon corresponding bits of two registers - AND, OR, X-OR. etc. on logic instructions. After execution the result is stored in the first operand.

Eg:- ORL A, @ R_i.

XRL A, R_n. ⇒ X-OR register to accumulator performed by this instruction

Branch Instruction:-
In this branch instruction
⇒ Unconditional jump :-
Upon their execution of a jump to a new location from where the program

continuous execution is executed.

c) Conditional Jump:-

This is a jump to a new program location is executed only if a specified condition is met. Otherwise the program normally proceeds with the next instruction.

Ex:- A call address 11.

[Absolute subroutine call] [Long subroutine call]

L call address 16

NOP (No operation).

Bit Processing Instruction:-

Similar to logic instructions bit oriented instructions perform logic operations. The difference is that these are performed upon single bit.

Ex:- CLRC (clear carry flag)

SETBC (sets the carry flag)

CPLC (Compliment the carry flag).

8051 IO ports and memory organization:-

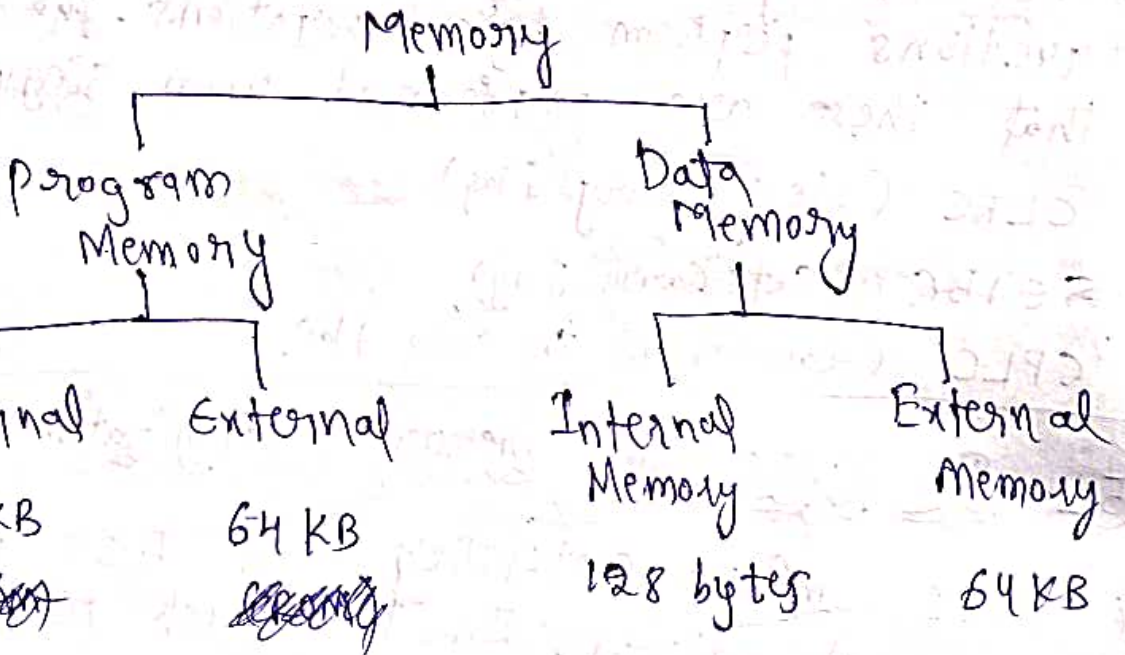
The 8051 micro controller has 4 IO ports drivers and latches. In this each port is of 8-bit and this is configured as input and output ports. By this there is 32 input/output pins allowed to the micro controller to be connected with the pheripheral devices. The pin can be configured as "one" for input and "zero" for o/p.

as the logic state.

Memory Organization :-

The memory is organized logically into program memory and data memory separately. The technique of organizing program and data memory with separate physical addresses is called Harward Architecture.

The program memory is read only type, the data memory is organized as read/write memory again program and data memories can be within the chip (or outside.



Intel 8051 has 128 bytes of RAM, 4KB of ROM within the chip.

The address bus of 8051 is 60 bits wide. So, it can access $2^{16} = 64$ KB of memory in program and in data.

A user can configure program memory 4KB, 60KB inside and outside respectively in the chip. Internal data memory accessed with 8-bit address and external data memory with 16-bit address. So, the maximum data memory that can be connected to the 8051 system is 64KB..

Procedures and Macros:

Procedure and Macros are two concepts in assembly language programming.

Macro:

Macro is a set of instructions and small in less than 10. The programmer can use it any where in the program by its name and used multiple times whenever required with the help of macro. These are not follow call-return method. Assembler directive MACRO is used to define macro, ENDM used after completion.

Procedure:

This is also like macro but they are used for largest set of instructions. Macro is useful for small set of instructions, which performs a specific task. Procedure contain procedure body which contains set of instructions and RET statements. Which denotes return statement. Procedure follow CALL-RETURN method.

Execution time of procedure is high compare to macro. Assembler directive is PROC for starting, ENDP for ending used.

Assembler Directive:-

An assembler is a program to convert an assembly language program in to equalent machine code modules which may be further converted to executable code.

Hints are given to the assembler using some predefined alphabetical strings called assembler directives. These help the assembler to correctly understand the assembly language programs to prepare the code.

DB - Define Byte:-

This reserve byte of memory locations in the memory.

Eg:- RANK DB 01 02 03
MESSAGE DB "Good Morning".

DW - Define Word:-

This reserves 16-bits instead of bytes.

WORDS DW 1 2 3 4

DQ - Define Quadwords:-

Reserves 4 words.

DT-Define Tenbytes :-

ASSUME logical segment name.

ASSUME CS : CODE

ASSUME DS : DATA

ENDP :-

End of procedure.

END S :-

End of segment.

Extra

① Solve for x

(i) $(1256)_8 = (x)_2$

Sol:-

$= (\underline{001}\,\underline{010}\,\underline{101}\,\underline{110})_2$

$\therefore (1256)_8 = (686)_{10}$ //

(ii) $(19.125)_{10} = (x)_8$

Sol:-

$8 \mid 19 \quad -3$

2

$(23)_8$

0.125×8

$1.000 \longrightarrow 1$

0.000×8

$0 \longrightarrow 0$

$\therefore (19.125)_{10} = (23.10)_8$ //

(iii) $(AC.2)_{16} = (x)_8$

Sol:-

$(1010\ \ 1100\ .\ 0010)_{16}$

$= (010\ \ 101\ \ 100\ \ .\ 001\ \ 000)_8$

$= (254.10)_8$

(iv) $(10011.11)_2 = (x)_{16}$

Sol:-

$(0001\ \ 0011\ .\ 1100)_2$

$= (13.C)_{16}$

② Each of the following arithmetic operation is correct at least one number system. Determine the possible bases of the numbers in each operation.

(i) $1234 + 5432 = 6666$

Sol $(1234)_{10} + (5432)_{10} = (6666)_{10}$

(ii) $\sqrt{41} = 5$

Sol $\sqrt{41} = 5$

$b^1\ b^0 \qquad b^0$

$\sqrt{4(b) + 1(b^0)} = 5b^0$

S.O.B.S.

$(\sqrt{4b+1})^2 = (5)^2 \Rightarrow 4b+1 = 25$

$4b = 24$

$\boxed{b = 6}$ //

$\therefore (\sqrt{41})_6 = (\sqrt{25})_{10} = 5$ //

(iii) $\frac{302}{20} = 12.1$

Sol $\frac{3b^2 + 0(b) + 2(b^0)}{2b + 0(b^0)} = \frac{1(b) + 2(b^0)}{b} + \frac{1}{b} \times b^{-1}$.

$\frac{3b^2+2}{2b} = b + 2 + \frac{1}{b}$

$\frac{3b^2+2}{2b} = \frac{b^2+2b+1}{b}$

$2b^2 + 4b + 2 = 3b^2 + 2$

$3b^2 - 2b^2 - 4b = 0$

$b^2 = 4b$

$\boxed{b = 4}$ //.

(iv) $23+44+14+32=223$

Sol: $2b+3+4b+4+b+4+3b+2 = 2b^2+2b+3$

$10b+13 = 2b^2+2b+3$

$2b^2-10b+2b-13+3=0$

$2b^2-8b-10=0$

$b^2-4b-5=0$

$b=5,-1 \Rightarrow \boxed{b=5}$ //

③. If $(789)_{16} + (473)_8 = (x)_4$.

Sol: given $(789)_{16} + (473)_8 = (x)_4$

$(1929)_{10} + (315)_{10} = (2244)_{10}$

$\therefore x = (2244)_{10} = (203010)_4$.

4 | 2244 – 0
4 | 561 – 1
4 | 140 – 0
4 | 35 – 3
4 | 8 – 0
2

BCD Addition:- (or) (8421).

184 + 576

	1	8	4
	5	7	6
	7	6	0
	0111	0110	0000

$\Rightarrow$ 0001 1000 0100
$\Rightarrow$ 0101 0111 0110
0110 1111 1010

↑ Incorrect BCD result.

Incorrect BCD results ని 9 కంటే ఎక్కువ ఉంటే 6 add చేయాలి.

$\boxed{6=0110}$

0110	1111	1010
	0110	0110
0111	0110	0000

← Correct BCD result.

① Write 4 different binary codes in a table for the decimal digits.

Sol:-

Decimal digit	BCD 8421	2421	Excess-3 code	84-2-1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	0101	1000	1011
6	0110	0110	1001	1010
7	0111	0111	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
Six unused Combinations	1010	1000	1101	0001
	1011	1001	1110	0010
	1100	1010	1111	0011
	1101	1011	0001	1100
	1110	1100	0010	1101
	1111	1101	0000	1110

② Explain weighted code and self complimenting property with example.

Sol:- Weighted Code:-

In this each bit pattern is assigned a weighting factor in such a way that each digit can be evaluated by adding the weights

of all the ones in the coded information.

Ex:- BCD code and 2421 code.

Self Complimenting:-

The property that the 9's compliment of decimal number is obtained directly by changing 1's to 0's and 0's to 1's.

i.e, by complimenting each bit.

Ex:- 2421 code and excess-3 code (or self complimenting code.

⇒ 6 → 1001 ↘ 1's
9's ↘ 3 → 0110 ↗ 1's

Gray Code:-

X-OR

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

1001
1101
1 ⇒ 1
10 ⇒ 1
00 ⇒ 0
01 ⇒ 1

1101
1 0 0 1

The code which exhibits only a single bit change from one code number to the next is known as gray code.

i.e, each gray code number differs from the preceding number by a single bit.

1. (i) Convert binary number to a gray code.

1001101

Sol: 1 0 0 1 1 0 1 ← binary

1 1 0 1 0 1 1 ← gray code.

(ii) Convert graycode to binary number.

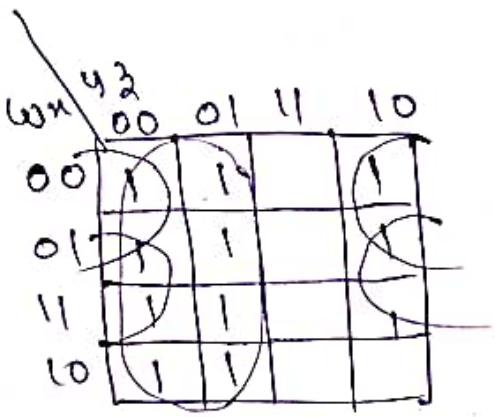
Sol: 1 1 0 1 0 1 1 ← gray code

1 0 0 1 1 0 1 ← binary.

2. Simplify the boolean function

(i) $F(w,x,y,z) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$.

Sol:



wxyz
0000
0001
0100
0101
1100
1101
1000
1001
$\bar{y}$

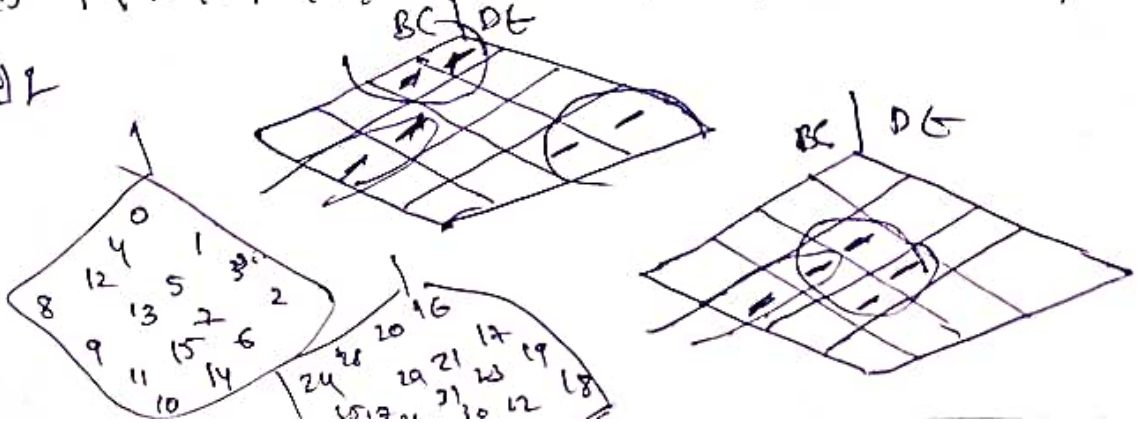
wxyz
0010
0110
0000
0100
$w\bar{z}$

wxyz
0100
1100
0110
1110
$x\bar{z}$

$\therefore F(w,x,y,z) = \bar{y} + w\bar{z} + x\bar{z}$ (or) $y' + w'z' + xz'$.

(ii) $F(A,B,C,D,E) = \Sigma(0,2,4,6,9,13,21,23,25,29,31)$.

Sol:



$$= A'(B'C') + ACE + (A+A')\cdot BD'E //.$$

$$= A'B'C' + ACE // + BD'E //.$$