UNIT I

Introduction

- Number System
  - Number system defines set of values used to represent quantity
  - Number system in most common use today is the Arbic system
  - For decimal number systems base is 10 ie (0,1,⋯9)

    Hence decimal number is represented like

    $(349.25)_{10}$

    ie $3\times10^2 + 4\times10^1 + 9\times10^0 \cdot 2\times10^{-1} + 5\times10^{-2}$

- Binary number system.
  - binary number system uses base 2 as it has two digits only
  - binary number system uses only two digits namely 0 and 1
  - Weighted value for different positions can be represented as follows.

	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	← position weight
MSB ⋯					•				⋯ LSB

MSB - Most significant bit → with the highest values

LSB - Least significant bit → with the smallest values.

- Octal number system
  - the base used for octal system is 8
  - Each digit in octal system will assume 8 different values (ie 0,1,⋯7)
  - Weighted values can be represented as follows

	8^3	8^2	8^1	8^0		8^{-1}	8^{-2}	8^{-3}	
MSB					•				LSB

- Hexadecimal number system
  - the base of Hexadecimal number system is 16
  - Each digit in Hexadecimal number system will

	16^3	16^2	16^1	16^0		16^{-1}	16^{-2}	16^{-3}		
MSB ...					.				...	LSB

- In binary number system
  - bit → 1 bit
  - Nibble → 4 bit
  - Byte → 8 bit
  - word → 16 bit
  - Double word → 32 bit
- the important drawback is, it requires large or very long string of 1's and 0's to represent decimal number

  ie $(128)_{10} \rightarrow (10000000)_2$

- Conversion
  - Conversion of binary to decimal

Ex. $(1011 \cdot 01)_2$

$= 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 \cdot 0\times2^{-1} + 1\times2^{-2}$

$= 8 + 0 + 2 + 1 + \cdot 0 + 0.25$

$= (11.25)_{10}$

- Conversion of octal to decimal.

Ex. ~~(314)8~~ $(365.24)_8$

$= 3\times8^2 + 6\times8^1 + 5\times8^0 \cdot + 2\times8^{-1} + 4\times8^{-2}$

$= 192 + 48 + 5 + 0.25 + 0.0625$

$= (245.3125)_{10}$

- Conversion of Hex to decimal

Ex $(4C8.2)_{16}$

$= 4\times16^2 + C\times16^1 + 8\times16^0 + 2\times16^{-1}$

$= 1024 + 192 + 8 + 0.125$

$= (1224.125)_{10}$

- Conversion from decimal to other system
  - conversion of decimal to binary number system

Ex $(105)_{10}$

division	number	Reminder
2	105	
2	52	1
2	26	0
2	13	0
2	6	1
2	3	0
2	1	1
	0	1 ↑

Answer is 1101001

$\therefore (105)_{10} = (1101001)_2$

- Conversion of decimal to octal number

Ex $(204)_{10}$

divide	number	Reminder
8	204	
8	25	4
8	3	1
	0	3 ↑

Answer is 314

$(204)_{10} = (314)_8$

- Conversion of decimal to Hexadecimal number.

Ex $(259)_{10}$

divide	number	Reminder
16	259	
16	16	3
16	1	0
		1 ↑

$(259)_{10} = (\cancel{301})\ (103)_{16}$

Date :
Page :

Fractional part conversion

- Conversion of decimal to binary

Ex- $(0.42)_{10}$

Decimal fraction	Base	product	carry
0.42	2	0.84	0
0.84	2	1.68	1
0.68	2	1.36	1
0.36	2	0.72	0
0.72	2	1.44	1

$(01101)_2$

$(0.42)_{10} = (0.01101)_2$

- Conversion of decimal to octal

Ex - $(0.6234)_{10}$

Decimal fraction	Base	product	carry
0.6234	8	4.9872	4
0.9872	8	7.8976	7
0.8976	8	7.1808	7
0.1808	8	1.4464	1
0.4464	8	3.5712	3

~~37714~~

$(0.6234)_{10} = (0.47713)_8$

- Conversion of decimal to Hexadecimal.

Ex $(0.122)_{10}$

Decimal fraction	base	product	carry	Hex
0.122	16	1.952	1	1
0.952	16	15.232	15	F
0.232	16	3.712	3	3
0.712	16	11.392	11	B
0.392	16	6.272	6	6
0.272	16	4.352	4	4

1F3B64

$(0.122)_{10} = (0.1F3B64)_{16}$

- Binary to octal conversion

Ex $(11010010)_2$

- make group of three digit

$\underbrace{011}\ \underbrace{010}\ \underbrace{010}$

ie 3 2 2 = $(322)_8$

- Binary to Hex conversion

Ex $(1010\ 1111\ 1011\ 0010)_2$

- make group of four digit

$\underbrace{1010}\ \underbrace{1111}\ \underbrace{1011}\ \underbrace{0010}$

ie A F B 2 = $(AFB2)_{16}$

- Octal to binary

Ex $(364.25)_8$

3 6 4 . 2 5

$\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$

011 110 100 . 010 101 ie $(011110100.010101)_2$

- Conversion of Hex to binary

Ex. $(AFB2)_{16}$

A F B 2

$(1010\ 1111\ 1011\ 0010)_2$

- Octal to Hex conversion

Ex $(436)_8$

4 3 6

100 011 110 — Binary

$\downarrow$ $\downarrow$ $\downarrow$

$(1\ 1\ E)_{16}$ — Hex.

$(436)_8 \rightarrow (11E)_{16}$

- Hex to octal conversion

Ex $(4CA)_{16}$

4 C A

0100 1100 1010

2 3 1 2 ie $(2312)_8$

Fractional Hex to octal conversion

$(0.12E)_{16}$

$0.\underbrace{000}\underbrace{1\,00}\underbrace{10\,1}\underbrace{110}$

0. 0 4 5 6 i.e. $(0.0456)_8$.

One more example

eg. $(68.4B)_{16}$

$\underbrace{001}\underbrace{10\,1}\underbrace{000}.\underbrace{010}\underbrace{0\,10}\underbrace{110}$

1 5 0 . 2 2 6

i.e. $(150.226)_8$

- Signed Magnitude numbers.
  - MSB of binary number is used to represent the sign
  - 0 is used to represent + sign
    - i.e. 01011001 → $(+89)_{10}$
  - 1 is used to represent − sign
    - i.e. 11011001 → $(-89)_{10}$
  - Unsigned 8-bit number covers the decimal range of 255 number starting from 0
  - Signed 8 bit number covers the decimal range of $(-127)_{10}$ to $(+127)_{10}$

- 1's complement
  - the 1's complement of number is found by changing all 1's to 0's and all 0's to 1's
  - eg. 1 0 0 1 0 1 0 0
    ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
    0 1 1 0 1 0 1 1 ← 1's complement
  - the result will be same when we substract the above number from 1111 1111

- 2's complement
  - The 2's complement is obtained by adding 1 to the LSB of 1's complement of number
    ∴ 2's complement = 1's complement + 1

e.g.

```
   1 0 1 1 0 0 1 0
   0 1 0 0 1 1 0 1  ← 1's complement
               + 1
  ----------------
   0 1 0 0 1 1 1 0  ← 2's complement
```

- to convert negative number to a positive number find it's 2's complement
- 2's complement of 2's complement of a number results in the original number itself.

- Binary arithmetic including octal and Hexadec

- **Binary addition**

Carry case

	Result	carry
0 + 0 =	0	
0 + 1 =	1	
1 + 0 =	1	
1 + 1 =	0	1

Ex

```
  1 1 1 1 1  ← carry
    1 0 1 1 1
    1 1 0 0 1
  -----------
  1 1 0 0 0 1
```

Here $1+1+1 = (1+1)+1$
$= 10+1$
$= (11)_2$

- Binary substraction

	substraction	borrow
0 - 0 =	0	
1 - 0 =	1	
1 - 1 =	0	
0 - 1 =	1	1

Ex -

```
       1
    1 1 0 1 1
  - 1 0 1 1 0
    -1
  -----------
    0 0 1 0 1
```

→ $(00101)_2$

Example: Substract decimal numbers $(38)_{10}$ & $(29)_{10}$ by converting them into binary

- Octal arithmetic

Addition - Ex

```
    6 3 4
  + 1 5 2
  -------
   1 1
  1 0 0 6
```

→ $(1006)_8$

Ex ② Add $(354)_8 + (266)_8 + (123)_8$

```
    3 5 4
    2 6 6
    1 2 3
  -------
    1 1
Ans →  7 6 5
```

→ $(765)_8$

nal.

- Substraction using ~~8's~~ 7's compliment

Ex - Find 7's compliment of $(512)_8$

$(512)_8$

7's complement = 777 - 512

= $(265)_8$

Ex - perform substraction using 7's compliment

$(536)_8 - (345)_8$

7's compliment of 345 = 777 - 345

= 432

Add this to 1st number 536

+ 432

(1170)

Final carry

~~516~~ 176 - ~~413~~ 157

7's → 777 - 157

= 620

Ans = 620 + 176

= 1016

Add +1

(017)

As final carry is 1 answer is positive i.e. $(171)_8$

Ex. - perform substraction $(161)_8 - (243)_8$ using 7's complement

7's complement of 243 = 777 - 243

= 534

Add this to 1st number = 161

534

0715

No final carry to result is negetive..

No carry so take 7's complement

= 777 - 715

= $(062)_8$ ← Answer but the result is negetive.

- Multiplication

Ex: perform $(12)_8 * (7)_8$

convert into binary 001 010 × 000 111

i.e.

1010

× 1111

1010

1010+

0000++

1000110

- Hexadecimal arithmetic

- Hex addition

① C 2 — $(12)_{10}$ $(2)_{10}$

\+ 3 E — $(3)_{10}$ $(14)_{10}$

1 $(16)_{10}$

$(16)_{10}$

1 0 0

→ $(100)_{16}$

②

DDCC $(13)_{10}$ $(13)_{10}$ $(12)_{10}$ $(12)_{10}$

BBAA $(11)_{10}$ $(11)_{10}$ $(10)_{10}$ $(10)_{10}$

1 1 1

1 9 9 7 6

→ $(19976)_{16}$

- Hex substraction

perform the substraction $(D8A)_{16} - (426)_{16}$

1. obtain 15's complement

15 15 15 − 4 2 6 = BD9

2. Add this to first number

$(D\ 8\ A)_{16}$ $(13)_{10}$ $(8)_{10}$ $(10)_{10}$

$(B\ D\ 9)_{16}$ $(11)_{10}$ $(13)_{10}$ $(9)_{10}$

1 1

carry → 1 ; 9 6 3

(indicates that the result is in positive form)

(963)

\+1

964

- Hex Multiplication

- Multiply $(72)_{16}$ and $(39)_{16}$

72 − $(0111\ 0010)_2$

39 − $(0011\ 1001)_2$

Multiply these two

1 1001 0110 0010

i.e $(1962)_{16}$

Date :
Page :

- Binary arithmetics using 1's complement & 2's complement
  - We can represent substraction as A + (-B)
  - We can represent 1's complement or 2's complement form and use addition instead of substraction.
  - there are four possible cases.

Case 1: if A > B

perform $(9)_{10} - (4)_{10}$

a) convert 4 into binary - $(0100)_2$

1's complement - $(1011)_2$

b) Add it to first number i.e. 1001 + 1011

= 1 0100

c) Add this carry bit to number i.e. 0100 + 1

= 0101

Case 2: if A < B

perform $(4)_{10} - (9)_{10}$

a) convert 9 into binary - $(1001)_2$

1's complement - $(0110)_2$

b) Add it to first number i.e. 0100 + 0110

= 0 1010

c) As carry bit 0, it means answer is negetive & in 1's complement form.

∴ (1010) → ~~1's complement~~ invert it

$(0101)_2 \to (5)_{10}$ but answer is negetive

= (-5)

Case 3: both numbers are negetive

perform $(-4)_{10} - (-8)_{10}$

$= (-4)_{10} + (8)_{10}$

as $(-4)_{10}$ is negetive and B is positive

so we have to take the 1's complement of A

- rest of the process same.

Case 4: A = B

- the answer is ~~negat~~ zero

- Algebra for logic variable
  - Logic variables:

Name of gate   symbol   Expression   truth table.

1. NOT gate

$Y = \bar{A}$

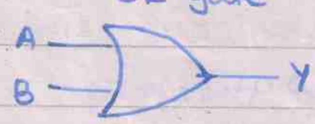
A	Y
0	1
1	0

2. AND gate

$Y = AB$

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

3. OR gate



$Y = \overline{A+B}$

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

4. Ex-Or gate

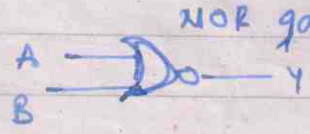
$Y = A \oplus B$

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

5. NAND gate

$Y = \overline{AB}$

6. NOR gate



$Y = \overline{A+B}$

* Nor & Nand gates are called as universal gate

- Boolean Laws.
  - There are three logic operators
    1. AND operator
    2. OR operator
    3. NOT operator

Logic gates:

- Logic gates are classified into three categories
  1. Basic gates a) NOT gate
     b) AND gate
     c) OR gate
  2. Universal gate a) NAND gate
     b) NOR gate
  3. Special purpose gates a) Ex-OR gate
     b) Ex- NOR gate.

Boolean Law:

1. Commutative law - $A \cdot B = B \cdot A$
   $A+B = B+A$
2. Associative law - $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
   $(A+B)+C = A+(B+C)$
3. Distributive law $A \cdot (B+C) = AB + AC$
4. AND law $A \cdot 0 = 0$
   $A \cdot 1 = A$
   $AA = A$
   $A \cdot \bar{A} = 0$
5. OR law $A+0 = A$
   $A+1 = 1$
   $A+A = A$
   $A+\bar{A} = 1$
6. Inversion law $\bar{\bar{A}} = A$
7. Important other laws $A + BC = (A+B)(A+C)$
   $\bar{A} + AB = \bar{A} + B$
   $\bar{A} + A\bar{B} = \bar{A} + \bar{B}$
   $A + AB = A$
   $A + \bar{A}B = A+B$

- De-Morgan's Theorems.
  1. Complement of a product is equal to the addition of complements
     ie $\overline{AB} = \overline{A} + \overline{B}$
  2. Complement of a sum is equal to product of complements.
     ie $\overline{A+B} = \overline{A} \cdot \overline{B}$

* Simplification of boolean algebra

1. Prove that

$$(A+\overline{B}+AB)(A+\overline{B})(\overline{A}B) = 0$$
$$(A+\overline{B})(A+\overline{B})(\overline{A}B) = 0$$
$$(A+\overline{B})(\overline{A}B) = 0$$
$$A\cdot\overline{A}B + \overline{B}\cdot\overline{A}\cdot B = 0$$
$$0+0 = 0$$

2. Simplify

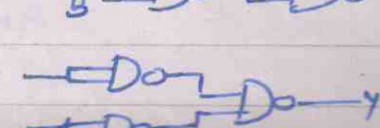
$$y = \overline{(\overline{AB}+\overline{A}+AB)}$$
$$= \overline{(\overline{A}+\overline{B}+\overline{A}+AB)}$$
$$= \overline{(\overline{A}+\overline{B}+AB)}$$
$$= \overline{\overline{A}} \cdot \overline{\overline{B}} \cdot \overline{AB}$$
$$= A\cdot B\cdot\overline{AB}$$
$$= A\cdot B\cdot(\overline{A}+\overline{B})$$
$$= A\cdot\overline{A}\cdot B + AB\cdot\overline{B}$$
$$= 0+0$$
$$= 0.$$

* Realization using gates.

- NAND gate – 

  - As NOT gate 

  - As AND gate 

  - As OR gate 

* SOP (sum of product form)

  - The sum of product in SOP form are OR and AND function

    e.g. $Y = AB + AC + BC$

    $Y = \bar{P}Q + PQR + PQ\bar{R}$

  - Standard SOP form contains i.e. Each literals consists of all literals in the complemented or uncomplemented form

    e.g. $Y = ABC + A\bar{B}C + \bar{A}\bar{B}C$

* POS (product of the sums form)

  - The product of sum in POS form are AND and OR function

    e.g. $Y = (A+B)\cdot(B+\bar{C})\cdot(A+C)$

  - standard POS form contains i.e. each literals consists of all literals in the uncomplemented or complemented form.

    $Y = (A+B+C)(A+B+\bar{C})\cdot(A+\bar{B}+C)$

* SOP form can be converted into standard SOP form by adding a variable and complement variable which ~~a~~ is not present in that term.

  e.g. $Y = (\cancel{ABC}) + (\cancel{A\bar{B}C}) + \bar{A}\cancel{B}C$

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

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

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

in same way pos form can be converted into standard pos form by Adding variable and complement of variable

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

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

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

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

Date :
Page :

Minterm and Max term

Minterm - Each individual term in standard sop is ~~is~~ called as minterm.

- represented by m_i
- i represent equivalent of natural binary number corrospondiy to minterm with
- normal variable taken as 1's

Maxterm - Each individual term in standard pos ~~for~~ called as max term

- represented as M_i
- i represent equivalent of natural binary number corrospondiy to maxterm with normal variable taken as 0

Minterm and Maxterm for four variable

variable				minterm m_i	maxterm M_i
A	B	C	D		
0	0	0	0	$\bar{A}\bar{B}\bar{C}\bar{D} = m_0$	$A+B+C+D = M_0$
0	0	0	1	$\bar{A}\bar{B}\bar{C}D = m_1$	$A+B+C+\bar{D} = M$
0	0	1	0	$\bar{A}\bar{B}C\bar{D} = m_2$	$A+B+\bar{C}+D = M$
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		

* Representation of SOP and POS form using notation

Example for SOP

$$Y = \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC$$
$$= m_3 + m_4 + m_6 + m_7$$
$$= \Sigma M(3, 4, 6, 7)$$

Example for POS form

$$Y = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)\cdot(\bar{A}+B+\bar{C})$$
$$= M_0 \cdot M_1 \cdot M_2 \cdot M_5$$
$$= \Pi M(0, 1, 2, 5)$$

* K-Map representation of logical function

- Design of K-map
  - In an n-variable K-map there are 2^n cells.
  - Gray code has been used for identification of cells.
  - for four variable the K map will be as follows.

CD \ AB	00	01	11	10
00	0 $\bar{A}\bar{B}\bar{C}\bar{D}$	1 $\bar{A}B\bar{C}\bar{D}$	3 $AB\bar{C}\bar{D}$	2 $A\bar{B}\bar{C}\bar{D}$
01	4 $\bar{A}\bar{B}\bar{C}D$	5 $\bar{A}B\bar{C}D$	7 $\bar{A}BCD$	6 $A\bar{B}CD$
11	$\bar{A}\bar{B}C\bar{D}$	$\bar{A}BC\bar{D}$	$ABC\bar{D}$	$A\bar{B}C\bar{D}$
10	8 $\bar{A}\bar{B}CD$	$\bar{A}BCD$	$ABCD$	$ABCD$

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

For
- SOP form you need to write $\bar{A}\bar{B}\bar{C}\bar{D}$ to $ABCD$ in respective cell.
- for POS form you need to write $A+B+C+D$ to $\bar{A}+\bar{B}+\bar{C}+\bar{D}$ in respective cell.

* Representation of standard sop form

$$y = \bar{A}\bar{B}\bar{C}\bar{D} + AB\bar{C}\bar{D} + \bar{A}B\bar{C}D + A\bar{B}\bar{C}D$$
$$+ \bar{A}BCD + ABCD + ABC\bar{D}$$
$$= \Sigma m(0, 5, 7, 9, 12, 14, 15)$$

k- map representation

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

* Representation of standard pos form on k map

$$y = (A+B+C+\bar{D})(A+B+\bar{C}+D)(A+B+\bar{C}+\bar{D})$$
$$(A+\bar{B}+C+D)(A+\bar{B}+\bar{C}+D)(\bar{A}+B+C+D)$$
$$(\bar{A}+B+\bar{C}+D)(\bar{A}+B+\bar{C}+\bar{D})(\bar{A}+\bar{B}+C+\bar{D})$$
$$= \Pi M(1, 2, 3, 4, 6, 8, 10, 11, 13)$$

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

Trooth table for sop and pos.

Dec	A	B	C	D	Y	A	B	C	D	Y
0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	1	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	0	1	0
4	0	1	0	0	0	0	0	0	0	0
5	0	1	0	1	1 *	0	0	0	1	1
6	0	1	1	0	0	0	0	1	0	0
7	0	1	1	1	1	0	0	1	1	1
8					0					

k- map grouping.

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

- grouping can be either of
  - 2 cells
  - 4 cells
  - 8 cell.

2 cell reduces 1 variable
4 cell reduces 2 variable
8 cell reduces 3 variable.

Example $f(A,B,C,D) = \Sigma m(0,1,2,3,5,7,8,9,11,$

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

$f(A,B,C,D) = ABC\bar{D} + \bar{B}\bar{C} + \bar{B}D + \bar{A}D +$

Date :
Page :

Quine-Mcclaskey minimization technique.

- Requirement of logic minimization
  1. Should have capability of handling large number of variables.
  2. Should not depend on humen recognising
  3. Should ensure minimized expression
  4. Should be suitable computer solution
- Method.

Example

$Y(A,B,C,D) = \Sigma m(0,1,3,7,8,9,11,15)$

step 1: Should be converted into minterm for if not.

step 2: group based upon contaiing of no. of ones one 1, two 1s, three 1s and so on

- First write binary presentation

	A	B	C	D
0	0	0	0	0
1	0	0	0	1
3	0	0	1	1
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
11	1	0	1	1
15	1	1	1	1

- grouping

Group	minterm	variable A B C D	tickmark
0	0	0 0 0 0	✓
1	1	0 0 0 1	✓
	8	1 0 0 0	✓
2	3	0 0 1 1	✓
	9	1 0 0 1	✓
3	7	0 1 1 1	✓
	11	1 0 1 1	✓
4	15	1 1 1 1	✓

Step 3: The boolean algebraic thm $A + \bar{A} = 1$ is applied to the pairs of minterms in w[hich] only one variable is different and all oth[er] variables are same.

- Combination of minterms of groups of two

Group	minterm	Variables A B C D	tickemark
0	0, 1	0 0 0 –	✓
	0, 8	– 0 0 0	✓
1	1, 3	0 0 – 1	✓
	1, 9	– 0 0 1	✓
	8, 9	1 0 0 –	✓
2	3, 7	0 – 1 1	✓
	3, 11	– 0 1 1	✓
	9, 11	1 0 – 1	✓
3	7, 15	– 1 1 1	✓
	11, 15	1 – 1 1	✓

Step 4: Repeat step 3

Group	minterm	variable A B C D	tickemark
0	0, 1, 8, 9	– 0 0 –	
	0, 8, 1, 9	– 0 0 –	
1	1, 3, 9, 11	– 0 – 1	
	1, 9, 3, 11	– 0 – 1	
2	3, 7, 11, 15	– – 1 1	
	3, 11, 7, 15	– – 1 1	

Step 5: Repeat step 4 but both dash and one element should be same. as there is no such term we will stop.

Step 6: All non-checked term are prime-implicant so.

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

Step 7: make table of prime implicant.

PI terms	Decimal numbers	0,	1,	3,	7,	8,	9,	11,	15
$\bar{B}\bar{C}$ →	0, 1, 8, 9.	X	X			X	X		
$\bar{B}D$	1, 3, 9, 11		X	X			X	X	
CD →	3, 7, 11, 15			X	X			X	X

observe the table

- $\bar{B}\bar{C}$ and CD are EPI
- observe the other minterms and see weather these are contained in the EPI or not.
  as 1, 3, 9, 11 are all contained in EPI
- so expression will be
  $Y(A, B, C, D) = \bar{B}\bar{C} + CD$