UNIT – 1

Number Systems and Logic Gates

Unit-01/Lecture-01

Definitions:

- Base or Radix: The no. of values that a digit (0ne character) can assume is equal to the base of the system. Decimal system has radix 10
- The Largest value of a digit is always one less than the base. For Decimal system it is 10-1 = 9
- Each digit position (place) represents a different multiple of base. i.e. numbers have positional importance.

Radix (Base) r	Characters (r)	Highest number (r – 1)
2 (Binary)	0, 1	1
3	0, 1, 2	2
4	0, 1, 2, 3	3
5	0, 1, 2, 3, 4	4
6	0, 1, 2, 3, 4, 5	5
7	0, 1, 2, 3, 4, 5, 6	6
8 (Octal)	0, 1, 2, 3, 4, 5, 6, 7	7
9	0, 1, 2, 3, 4, 5, 6, 7, 8	8
10 (Decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	9
:	:	:
:	:	:
16 (Hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	F (15)

Decimal

- Radix is 10, Values are 0,1,2,3,4,5,6,7,8,9, largest number is 9

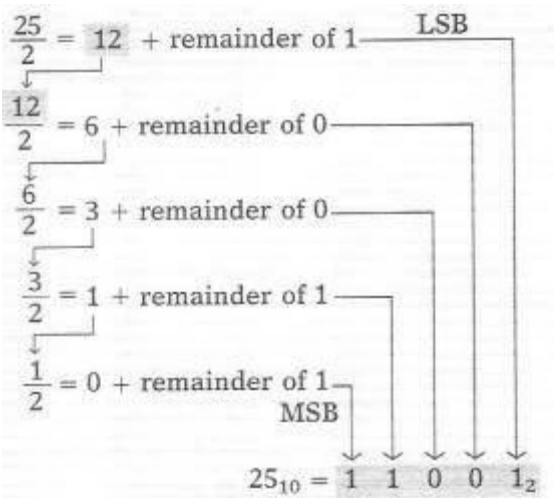
D-B conversion

$$45_{10} = 32 + 8 + 4 + 1 = 2^5 + 0 + 2^3 + 2^2 + 0 + 2^0$$
$$= 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1_2$$

a 0 is placed in the 2^1 and 2^4 positions, since all positions for. Another example is the following:

$$76_{10} = 64 + 8 + 4 = 2^6 + 0 + 0 + 2^3 + 2^2 + 0 + 0$$
$$= 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0_2$$

- Mutltiply the given fractional decimal no by the base (radix) r
- Record the carry generated in this multiplication as MSD
- Multiply only the fractionsl number of the product in step 2 by the base and record the carry as the next bit of MSD
- Repeat steps 2 and 3 upto the end. The last carry will represent the LSD of equivalent binary no.



Binary

- Values are 0,1 and are called bits
- Leftmost digit is MSB (Most Significant Bit) and Rightmost is LSB (Least Significant Bit)
- Applications: Mostly used in Digital Systems like Computers.

Name	Size (bits)	Example
Bit	1	1
Nibble	4	0101
Byte	8	0000 0101
Word	16	0000 0000 0000 0101
Double Word	32	0000 0000 0000 0000 0000 0000 0000 0101

- Write down the number

- Write down the weights for different positions
- Multiply each digit in the given number with the corresponding weight to obtain product numbers
- Add all the product numbers to get the decimal equivalent.

$$\begin{array}{ccccc} 1 & 1 & 0 & 1 & 1_2 \end{array}$$
$$2^4 + 2^3 + 0 + 2^1 + 2^0 = 16 + 8 + 2 + 1 = 27_{10}$$

Let's try another example with a greater number of bits:

$$\begin{array}{cccccccc} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1_2 = \end{array}$$
$$2^7 + 0 + 2^5 + 2^4 + 0 + 2^2 + 0 + 2^0 = 181_{10}$$

Convert 37_{10} to binary. Try to do it on your own before you look at the solution

Solution

$\frac{37}{2} = 18.5 \longrightarrow$ remainder of 1 (LSB)

$\frac{18}{2} = 9.0 \longrightarrow$ 0

$\frac{9}{2} = 4.5 \longrightarrow$ 1

$\frac{4}{2} = 2.0 \longrightarrow$ 0

$\frac{2}{2} = 1.0 \longrightarrow$ 0

$\frac{1}{2} = 0.5 \longrightarrow$ 1 (MSB)

Thus, $37^{10} = 100101_2$.

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Convert (27)10 and, Convert (40.25)10 to ()2, ()8, ()16	Dec 2003	10
Q.2	Convert (350.25)10 to ()2, ()8, ()16	Dec 2005	6
Q.3	Convert (210.25)10 to ()2, ()8, ()16	June 2006	4

Unit-01/Lecture-02

Octal

- Base is 8 and values are from 0 to 7, largest value is 7.
- Applications: Avoids large strings of 0s and 1s as in Binary system.
- It reduces the size of large binary no.

0	10	20	30	40	......	70	100
1	11	21	31	41	......	71	101
2	12	22	32	42	......	72	102
3	13	23	33	43	......	73	103
4	14	24	34	44	......	74	104
5	15	25	35	45	......	75	105
6	16	26	36	46	......	76	106
7	17	27	37	47	......	77	107

O-D conversion

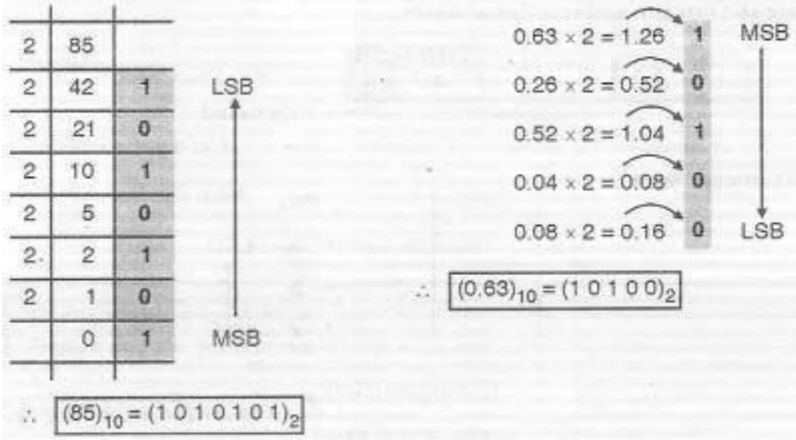
$$372_8 = 3 \times (8^2) + 7 \times (8^1) + 2 \times (8^0)$$
$$= 3 \times 64 + 7 \times 8 + 2 \times 1$$
$$= 250_{10}$$

$$24.68 = 2 \times (8^1) + 4 \times (8^0) + 6 \times (8^{-1})$$
$$= 20.75_{10}$$

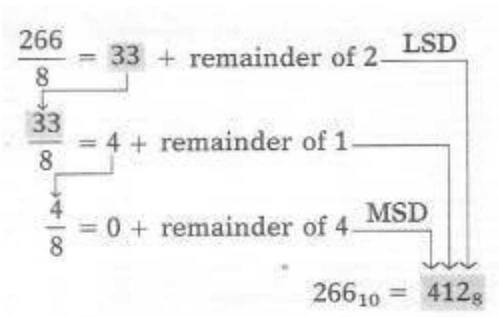
D-O conversion

- Separate the integer and fractional parts of the given decimal no.
- Convert the integer part into desired radix
- Convert the fractional part into desired radix
- Combine the results of steps 2 and 3 to get the final answer.

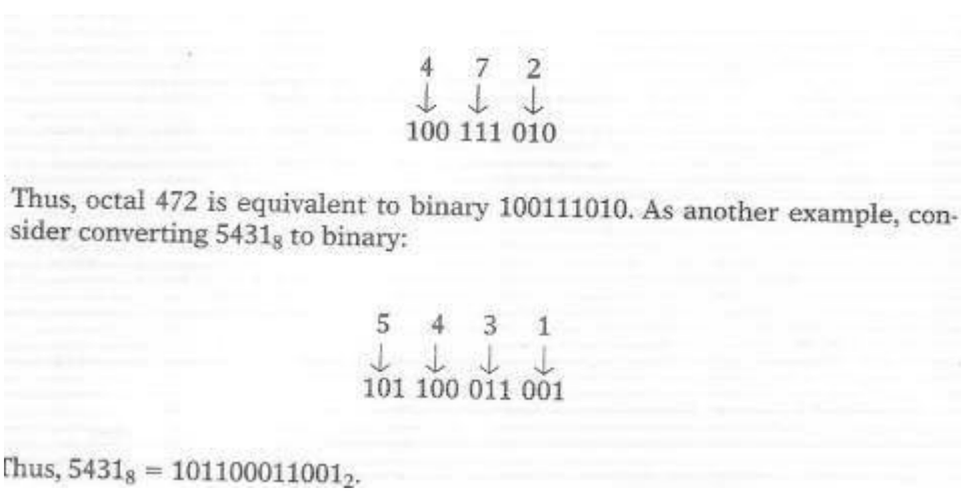
(85.63)10



(85.63)10 = (1010101.10100)2



O-B conversion



4 7 2
↓ ↓ ↓
100 111 010

Thus, octal 472 is equivalent to binary 100111010. As another example, consider converting 5431_8 to binary:

5 4 3 1
↓ ↓ ↓ ↓
101 100 011 001

Thus, $5431_8 = 101100011001_2$.

B-O conversion

- Group binary into 3 digits starting from LSB
- Covert each group into its equivalent decimal. The decimal no is same as octal as each group is restricted to 3.

$$\begin{array}{l} 1 \quad 1 \quad 0 \quad 1 \quad 1_2 \\ 2^4 + 2^3 + 0 + 2^1 + 2^0 = 16 + 8 + 2 + 1 \\ \qquad = 27_{10} \end{array}$$

Let's try another example with a greater number of bits:

$$\begin{array}{l} 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1_2 = \\ 2^7 + 0 + 2^5 + 2^4 + 0 + 2^2 + 0 + 2^0 = 181_{10} \end{array}$$

$$\underbrace{0\ 1\ 1}_{3}\ \ \underbrace{0\ 1\ 0}_{2}\ \ \underbrace{1\ 1\ 0}_{6_8}$$

O-H conversion

- Convert the given octal number into equivalent binary
- Then convert this binary into HEX.

Hexadecimal

- Base is 16, Values are 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F, largest value is F
- Applications: HEX no.sare very compact and easy to convert to binary and vice-versa. Used in programming languages.

0	10	20	30	....	90	A0	....	F0
1	11	21	31	....	91	A1	....	F1
2	12	22	32	....	92	A2	....	F2
3	13	23	33	....	93	A3	....	F3
4	14	24	34	....	94	A4	....	F4
.	.	.	.		.	.		.
.	.	.	.		.	.		.
.	.	.	.		.	.		.
.	.	.	.		.	.		.
D	1D	2D	3D	....	9D	AD	....	FD
E	1E	2E	3E	....	9E	AE	....	FE
F	1F	2F	3F	....	9F	AF	....	FF

H-Dconversion

$$
\begin{aligned}
356_{16} &= 3 \times 16^2 + 5 \times 16^1 + 6 \times 16^0 \\
&= 768 + 80 + 6 \\
&= 854_{10}
\end{aligned}
$$

$$
\begin{aligned}
2AF_{16} &= 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 \\
&= 512 + 160 + 15 \\
&= 687_{10}
\end{aligned}
$$

$$
\begin{aligned}
9F2_{16} &= \quad 9 \qquad\quad F \qquad\quad 2 \\
&= 1\ 0\ 0\ 1 \quad 1\ 1\ 1\ 1 \quad 0\ 0\ 1\ 0 \\
&= 100111110010_2
\end{aligned}
$$

D-Hconversion

(a) Convert 423_{10} to hex.

Solution

$$
\begin{aligned}
\frac{423}{16} &= 26 + \text{remainder of } 7 \\
\frac{26}{16} &= 1 + \text{remainder of } 10 \\
\frac{1}{16} &= 0 + \text{remainder of } 1 \\
423_{10} &= 1A7_{16}
\end{aligned}
$$

(b) Convert 214_{10} to hex.

Solution

$$
\begin{aligned}
\frac{214}{16} &= 13 + \text{remainder of } 6 \\
\frac{13}{16} &= 0 + \text{remainder of } 13 \\
214_{10} &= D6_{16}
\end{aligned}
$$

D-Hconversion

Separate the integer and fractional parts :

$$\underbrace{2003}_{\text{Integer}}.\underbrace{31}_{\text{Fractional}}$$

Convert integer part :

				Hex	
16	2003				
16	125	3	→	3	LSD
16	7	13	→	D	↑
	0	7	→	7	MSD

$\therefore (2003)_{10} = (7D3)_{16}$

Fig. P. 1.9.13(a)

Convert the fractional part into hex :

Decimal fraction		Base		Product	Carry	Hex	
0.31	×	16	=	4.96	4 →	4	MSD
0.96	×	16	=	15.36	15 →	F	
0.36	×	16	=	5.76	5 →	5	
0.76	×	16	=	12.16	12 →	C	
0.16	×	16	=	2.56	2 →	2	LSD

Fig. P. 9.13(b)

$\therefore (0.31)_{10} = (0.4F5C2)_{16}$

Combine the results of steps 2 and 3 :

ombining the results of steps 2 and 3 we get

$\therefore (2003.31)_{10} = (7D3.4F5C2)_{16}$

B-Hconversion

- Break binary into 4 bit sections from LSB to MSB
- Convert each group binary no to its HEX equivalent

$$1110100110_2 = \underbrace{0\,0\,1\,1}_{3}\ \underbrace{1\,0\,1\,0}_{A}\ \underbrace{0\,1\,1\,0}_{6}$$
$$= 3A6_{16}$$

H-Bconversion

- Convert each HEX digit to its 4 bit binary equivalent
- Combine the 4 bit sections by removing the spaces.

H-Oconversion

- Represent each HEX digit by a 4 bit binary no.

- Combine these 4 bit binary sections by removing the spaces
- Now group these binary bits into groups of 3 bits starting from LSB side
- Then convert each of this 3 bit group into an octal digit.

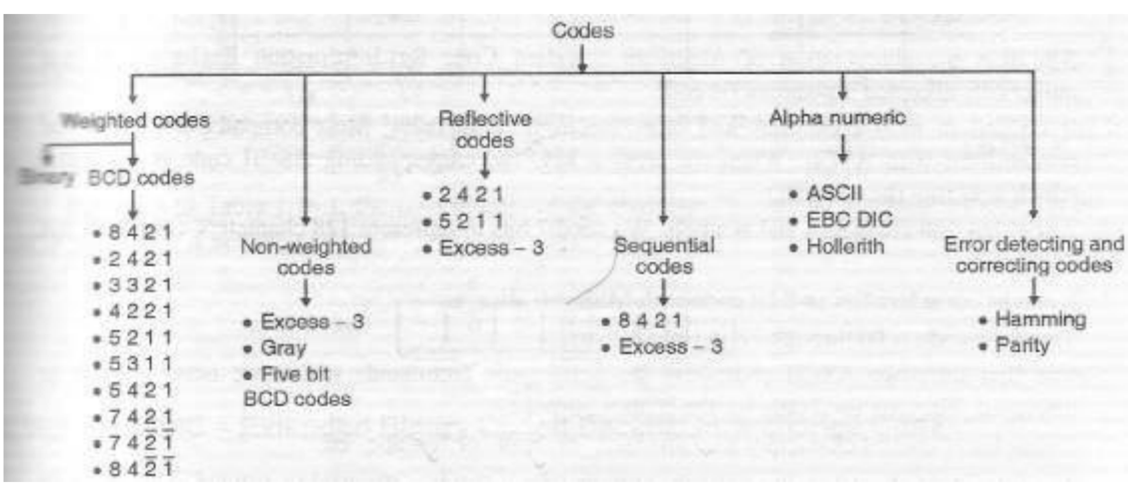
S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Convert (12.0625)10 to Binary	Dec 2006	5
Q.2	Convert (623.77)8 to ()2, ()8, ()16	Dec 2006	5
Q.3	Convert (2AC5.D)10 to ()2, ()8, ()16	June 2008	5

Unit-03/Lecture-03

Codes

- No.s, letters or words are represented by a specific group of symbols, it is said that the no is encoded
- The group of symbols is called code
- Codes can be represented in numbers or alphanumeric letters.

Types of Codes



Weighted Codes

- They obey positional weight principle
- Each position represents specific weight

Non Weighted Codes

- Positional weights are not assigned
- Eg Excess-3 and Gray Codes

BCD

Advanatages:

- Binary equivalents of decimal no.s are only needed to be remembers

Disadvantages:

- The addition and subtraction has different rules
- BCD arithmetic is bit complicated

D-BCD

8	7	4	(decimal)
↓	↓	↓	
1000	0111	0100	(BCD)

As another example, let us change 943 to its BCD-code representation:

9	4	3	(decimal)
↓	↓	↓	
1001	0100	0011	(BCD)

BCD-D

Convert 0110100000111001 (BCD) to its decimal equivalent.

Solution

Divide the BCD number into four-bit groups and convert each to decimal.

0110	1000	0011	1001
6	8	3	9

BCD-XS3

- Convert BCD to decimal
- Add (3)10 to this decimal number
- Convert into Binary to get the excess 3 code.

BCD Addition

```
  47     0100    0111   ← BCD for 47
 +35   + 0011    0101   ← BCD for 35
  82     0111    1100   ← invalid sum in first digit
            1←┐  0110   ← add 6 to correct
         1000 └─0010    ← correct BCD sum
           8       2
```

```
          ┌───┐
          ↓   │
          1   │
 59     0101  │ 1001   ← BCD for 59
+38   + 0011  │ 1000   ← BCD for 38
 97     1001  └─0001   ← perform addition
                0110   ← add 6 to correct
        1001    0111     BCD for 97
          9       7
```

```
 275      0010    0111    0101   ← BCD for 275
+641    + 0110    0100    0001   ← BCD for 641
 916      1000    1011    0110   ← perform addition
        +         0110           ← add 6 to correct second digit
          1001    0001    0110   ← BCD for 916
```

Excess 3

- It is no weighted code
- The excess 3 code word are derived from the 8421 BCD code word by adding (0011)2 or (3)10 to each word in 8421.

```
                                    Add
Decimal Number → 8421 BCD ——→ Excess - 3 code
                                    0011
```

Decimal	BCD				Excess - 3			
	8	4	2	1	BCD + 0 0 1 1			
					8	4	2	1
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Decimal	Binary	Octal	Hexadecimal	BCD	GRAY
0	0	0	0	0000	0000
1	1	1	1	0001	0001
2	10	2	2	0010	0011
3	11	3	3	0011	0010
4	100	4	4	0100	0110
5	101	5	5	0101	0111
6	110	6	6	0110	0101
7	111	7	7	0111	0100
8	1000	10	8	1000	1100
9	1001	11	9	1001	1101
10	1010	12	A	0001 0000	1111
11	1011	13	B	0001 0001	1110
12	1100	14	C	0001 0010	1010
13	1101	15	D	0001 0011	1011
14	1110	16	E	0001 0100	1001
15	1111	17	F	0001 0101	1000

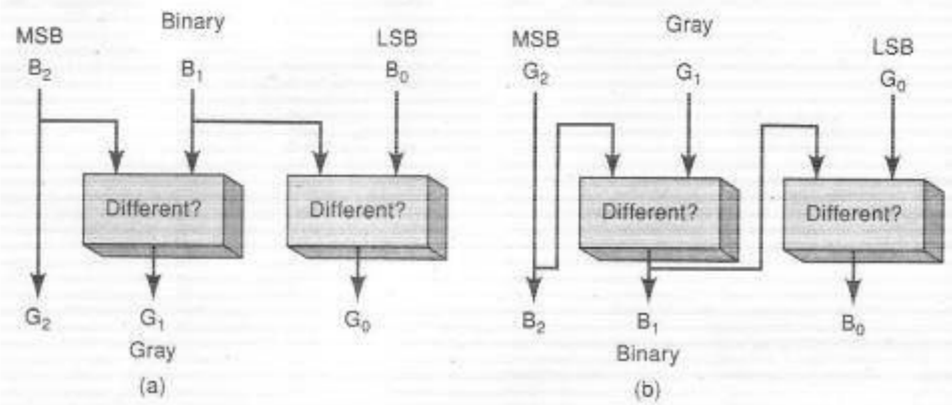
S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Convert (10111101)2 to ()8, (C346)10 to ()2, (3906)10 to ()BCD, (370)8 to ()16	Jun 2005	10
Q.2	Convert (428)10 to ()XS3	Dec 2010	5

Unit-01/Lecture-04

GRAY

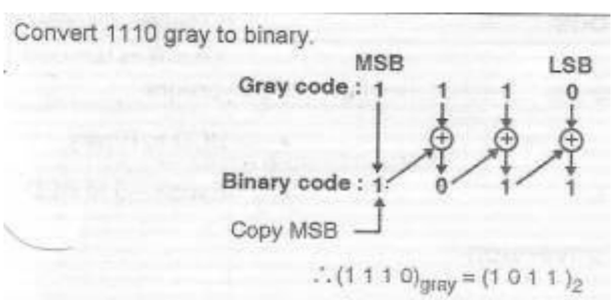
- It is non weighted code. No specific weights are assigned to the bit positions
- Only one bit change each time the decimal number is incremented called unit distance code
- Applications: used in Shaft position encoders (both linear and angular)

B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0



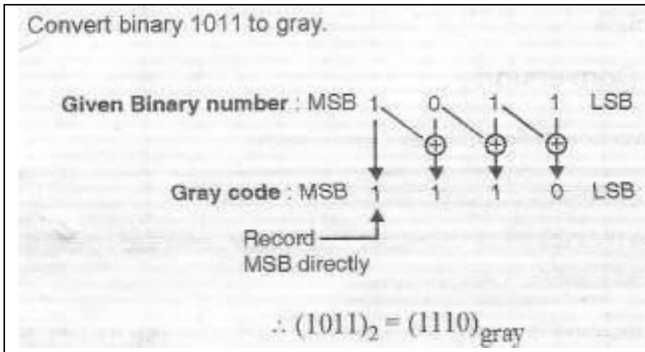
G-B

- The MSB of Gray and Binary are the same. So write it directly
- Add binary MSB to the next bit of Gray code. Record the result and ignore the carries
- Continue this process until the LSB is reached.



B-G

- Record the MSB as it is.
- Add this bit to the next position, recording the sum and neglecting any carry
- Record successive sums until completed.



Aphanumeric Codes

- Represent numbers as well as alphabetic characters
- Some represent symbols and instructions as well.
- Eg ASCII, EBCDIC (Extended Binary Coded Decimal Interchange Code), Hollerith

ASCII Codes (American Standard Code for Information Interchange)

- ASCII has 128 characters and symbols and requires 7 bits
- It can be considered as 8 bit with MSB = 0
- This 8 bit is 00 to 7F
- It has 32 control symbols those are not displayed on screen
- 94 printable characters and SPACE and DEL

Character	HEX	Decimal	Character	HEX	Decimal	Character	HEX	Decimal	Character	HEX	Decimal
NUL (null)	0	0	Space	20	32	@	40	64	`	60	96
Start Heading	1	1	!	21	33	A	41	65	a	61	97
Start Text	2	2	"	22	34	B	42	66	b	62	98
End Text	3	3	#	23	35	C	43	67	c	63	99
End Transmit.	4	4	$	24	36	D	44	68	d	64	100
Enquiry	5	5	%	25	37	E	45	69	e	65	101
Acknowlege	6	6	&	26	38	F	46	70	f	66	102
Bell	7	7	'	27	39	G	47	71	g	67	103
Backspace	8	8	(	28	40	H	48	72	h	68	104
Horiz. Tab	9	9	)	29	41	I	49	73	i	69	105
Line Feed	A	10	*	2A	42	J	4A	74	j	6A	106
Vert. Tab	B	11	+	2B	43	K	4B	75	k	6B	107
Form Feed	C	12	,	2C	44	L	4C	76	l	6C	108
Carriage Return	D	13	-	2D	45	M	4D	77	m	6D	109
Shift Out	E	14	.	2E	46	N	4E	78	n	6E	110
Shift In	F	15	/	2F	47	O	4F	79	o	6F	111
Data Link Esc	10	16	0	30	48	P	50	80	p	70	112
Direct Control 1	11	17	1	31	49	Q	51	81	q	71	113
Direct Control 2	12	18	2	32	50	R	52	82	r	72	114
Direct Control 3	13	19	3	33	51	S	53	83	s	73	115
Direct Control 4	14	20	4	34	52	T	54	84	t	74	116
Negative ACK	15	21	5	35	53	U	55	85	u	75	117
Synch Idle	16	22	6	36	54	V	56	86	v	76	118
End Trans Block	17	23	7	37	55	W	57	87	w	77	119
Cancel	18	24	8	38	56	X	58	88	x	78	120
End of Medium	19	25	9	39	57	Y	59	89	y	79	121
Substitue	1A	26	:	3A	58	Z	5A	90	z	7A	122
Escape	1B	27	;	3B	59	[	5B	91	{	7B	123
Form separator	1C	28	<	3C	60	\	5C	92	|	7C	124
Group separator	1D	29	=	3D	61	]	5D	93	}	7D	125
Record Separator	1E	30	>	3E	62	^	5E	94	~	7E	126
Unit Separator	1F	31	?	3F	63	_	5F	95	Delete	7F	127

Unit-01/Lecture-05

Theorems and Properties

BOOLEAN THEOREMS

1. $x \cdot 0 = 0$
2. $x \cdot 1 = x$
3. $x \cdot x = x$
4. $x \cdot \bar{x} = 0$
5. $x + 0 = x$
6. $x + 1 = 1$
7. $x + x = x$
8. $x + \bar{x} = 1$
9. $x + y = y + x$
10. $x \cdot y = y \cdot x$
11. $x + (y + z) = (x + y) + z = x + y + z$
12. $x(yz) = (xy)z = xyz$

3a. $x(y + z) = xy + xz$
13b. $(w + x)(y + z) = wy + xy + wz + xz$
14. $x + xy = x$
5a. $x + \bar{x}y = x + y$
15b. $\bar{x} + xy = \bar{x} + y$
16. $\overline{x + y} = \bar{x}\,\bar{y}$
17. $\overline{xy} = \bar{x} + \bar{y}$

Boolean Functions

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 Laws	$A \cdot 0 = 0$ $A \cdot 1 = A$ $A \cdot A = A$ $A \cdot \bar{A} = 0$
5.	OR Laws	$A + 0 = A$ $A + 1 = 1$ $A + A = A$ $A + \bar{A} = 1$
6.	Inversion Law	$\bar{\bar{A}} = A$
7.	Other Important Laws	$A + BC = (A + B)(A + C)$ $\bar{A} + AB = \bar{A} + B$ $\bar{A} + A\bar{B} = \bar{A} + \bar{B}$ $A + AB = A$ $A + \bar{A}B = A + B$

To prove that A + AB = A :

$$LHS = A + AB = A(1 + B)$$

But $(1 + B) = 1$ $\quad \therefore LHS = A \cdot 1$ $\quad$ But $A \cdot 1 = A$

$$\therefore \quad LHS = A = RHS$$

$\therefore \quad A + AB = A$ **...Proved**

To prove that $A + \bar{A}B = A + B$:

$LHS = A + \bar{A}B = A + AB + \bar{A}B$ ... since $A = A + AB$

$$\therefore \quad LHS = A + B(A + \bar{A})$$

it $(A + \bar{A}) = 1$ $\quad \therefore \quad LHS = A + (B \cdot 1) = A + B$

$$\therefore \quad LHS = RHS$$

$\therefore \quad A + \bar{A}B = A + B$ **... Proved.**

To prove that (A + B) (A + C) = A + BC :

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

$\therefore \quad LHS = AA + AC + BA + BC$ .... According to the distributive law.

But $AA = A$, $\quad \therefore \quad LHS = A + AC + BA + BC$

$$\therefore \quad LHS = A(1 + C) + AB + BC$$

But $(1 + C) = 1$

$$\therefore \quad LHS = A + AB + BC = A(1 + B) + BC$$

But $(1 + B) = 1$

Unit-01/Lecture-06

Canonical Forms

- The meaning of Canonical is conforming to a general rule
- This rule states that each term used in a switching equation must contain all the available input variables.
- It can be expressed in SOP or POS forms
- When we simplify sometimes input variables are eliminated
- Canonical expressions are not simplified hence contains redundancies.
- These equations are not in the minimized form because each term contains all the literals

Standard Form

- Each term can contain 1,2 or any no of literals.
- Can be expressed in SOP or POS forms.

1.	$Y = AB + AB\bar{C} + \bar{A}BC$	Standard SOP
2.	$Y = AB + A\bar{B} + \bar{A}\,\bar{B}$	Canonical SOP
3.	$Y = (\bar{A} + B) \cdot (A + B) \cdot (A + \bar{B})$	Canonical POS
4.	$Y = (\bar{A} + B) \cdot (A + B + C)$	Standard POS

1. $Y = \underbrace{ABC}_{m_7} + \underbrace{\bar{A}BC}_{m_3} + \underbrace{A\bar{B}\bar{C}}_{m_4}$ ← Given logic expression / ← Corresponding minterms

$\therefore Y = m_7 + m_3 + m_4$

$\therefore Y = \Sigma m(3, 4, 7)$ ← other way of representation

where Σ denotes sum of products.

2. $Y = \underbrace{(A + \bar{B} + C)}_{M_2} \underbrace{(A + B + C)}_{M_0} \underbrace{(\bar{A} + \bar{B} + C)}_{M_6}$ ← Given expressic / ← Corresponding ma

$\therefore Y = M_2\ M_0\ M_6$

$\therefore Y = \Pi M(0, 2, 6)$ ← Other way of represent

where Π denotes product of sums.

Variables			Minterms	Maxterms
A	B	C	m_i	M_i
0	0	0	$\bar{A}\bar{B}\bar{C} = m_0$	$A+B+C = M_0$
0	0	1	$\bar{A}\bar{B}C = m_1$	$A+B+\bar{C} = M_1$
0	1	0	$\bar{A}B\bar{C} = m_2$	$A+\bar{B}+C = M_2$
0	1	1	$\bar{A}BC = m_3$	$A+\bar{B}+\bar{C} = M_3$
1	0	0	$A\bar{B}\bar{C} = m_4$	$\bar{A}+B+C = M_4$
1	0	1	$A\bar{B}C = m_5$	$\bar{A}+B+\bar{C} = M_5$
1	1	0	$AB\bar{C} = m_6$	$\bar{A}+\bar{B}+C = M_6$
1	1	1	$ABC = m_7$	$\bar{A}+\bar{B}+\bar{C} = M_7$

Ex: Express f(A,B,C) = (A'+B)(B'+C) in sum of min terms and product of max terms Dec 06, 5M

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

Simplify the expression to get canonical POS form :

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

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

$$= M_4 M_5 M_2 M_6$$

$$\therefore f(A, B, C) = \Pi M(2, 4, 5, 6)$$

Simplify the expression to get canonical SOP form :

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

$$= \bar{A}\bar{B} + \bar{A}C + BC \ (\because A\bar{A} = 0)$$

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

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

$$= \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + \bar{A}BC + ABC = m_1 + m_0 + m_3 + m_7$$

$$f(A, B, C) = \Sigma m(0, 1, 3, 7)$$

De Morgans Theorem

$$\overline{(x + y)} = \bar{x} \cdot \bar{y}$$
$$\overline{(x \cdot y)} = \bar{x} + \bar{y}$$

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

Example 1	*Example 2*
$z = \overline{A + \overline{B} \cdot C}$	$\omega = \overline{(A + BC) \cdot (D + EF)}$
$= \overline{A} \cdot (\overline{\overline{B} \cdot C})$	$= \overline{(A + BC)} + \overline{(D + EF)}$
$= \overline{A} \cdot (\overline{\overline{B}} + \overline{C})$	$= (\overline{A} \cdot \overline{BC}) + (\overline{D} \cdot \overline{EF})$
$= \overline{A} \cdot (B + \overline{C})$	$= [\overline{A} \cdot (\overline{B} + \overline{C})] + [\overline{D} \cdot (\overline{E} + \overline{F})]$
	$= \overline{A}\,\overline{B} + \overline{A}\,\overline{C} + \overline{D}\,\overline{E} + \overline{D}\,\overline{F}$

$$
\begin{aligned}
x &= \overline{\overline{AB} \cdot \overline{CD} \cdot \overline{EF}} \\
&= \overline{\overline{AB}} + \overline{\overline{CD}} + \overline{\overline{EF}} \\
&= AB + CD + EF
\end{aligned}
$$

Digital Logic Gates

Truth Table:

Consists of all the possible combinations of the inputs and the corresponding state of output of a logic gate.

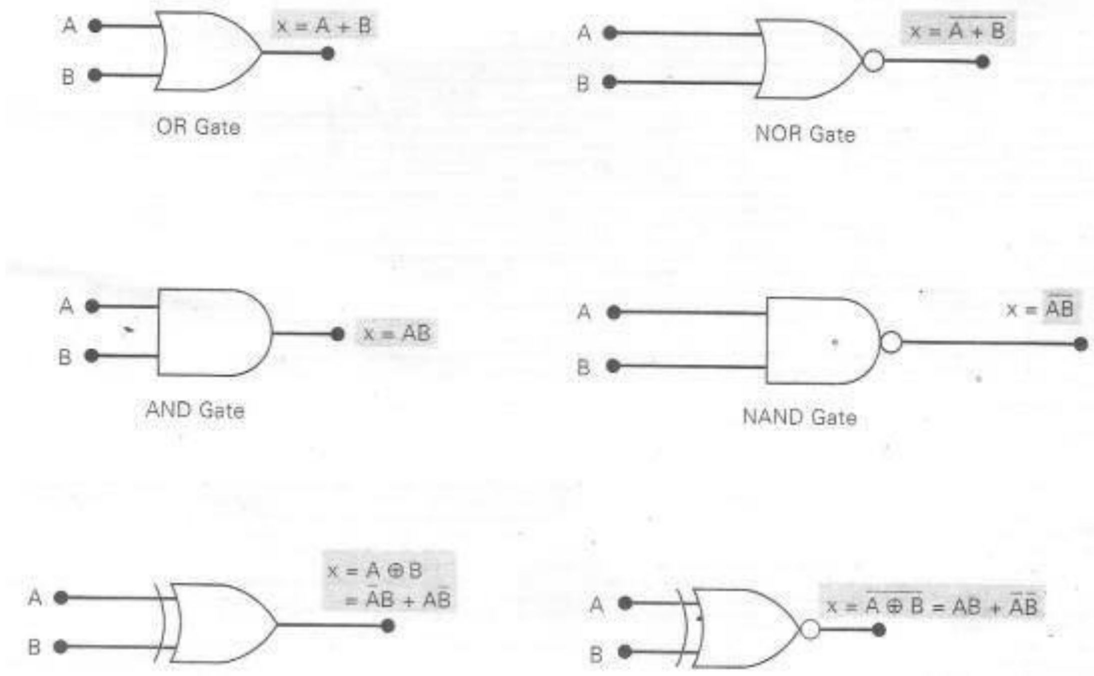
LOGIC GATE TRUTH TABLES

A	B	OR $A+B$	NOR $\overline{A+B}$	AND $A\cdot B$	NAND $\overline{A\cdot B}$	XOR $A\oplus B$	XNOR $\overline{A\oplus B}$
0	0	0	1	0	1	0	1
0	1	1	0	0	1	1	0
1	0	1	0	0	1	1	0
1	1	1	0	1	0	0	1

Basic Gates: NOT, AND, OR

Universal Gates: NAND, NOR

Derived Gates: EX-OR, EX-NOR


A
B
$x = A + B$
OR Gate
A
B
$x = \overline{A + B}$
NOR Gate
A
B
$x = AB$
AND Gate
A
B
$x = \overline{AB}$
NAND Gate
A
B
$x = A \oplus B$
$= \bar{A}B + A\bar{B}$
A
B
$x = \overline{A \oplus B} = AB + \bar{A}\bar{B}$

Unit-01/Lecture-07

Karnaugh Maps (K Maps)

SOP

1. $ABC + \bar{A}B\bar{C}$
2. $AB + \bar{A}B\bar{C} + \bar{C}\,\bar{D} + D$
3. $\bar{A}B + C\bar{D} + EF + GK + H\bar{L}$

POS

1. $(A + \bar{B} + C)(A + C)$
2. $(A + \bar{B})(\bar{C} + D)F$
3. $(A + C)(B + \bar{D})(\bar{B} + C)(A + \bar{D} + \bar{E})$

A	B	X
0	0	1 → $\bar{A}\bar{B}$
0	1	0
1	0	0
1	1	1 → AB

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

	$\bar{B}$	B
$\bar{A}$	1	0
A	0	1

(a)

A	B	C	X
0	0	0	1 → $\bar{A}\bar{B}\bar{C}$
0	0	1	1 → $\bar{A}\bar{B}C$
0	1	0	1 → $\bar{A}B\bar{C}$
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1 → $AB\bar{C}$
1	1	1	0

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

	$\bar{C}$	C
$\bar{A}\bar{B}$	1	1
$\bar{A}B$	1	0
AB	1	0
$A\bar{B}$	0	0

(b)

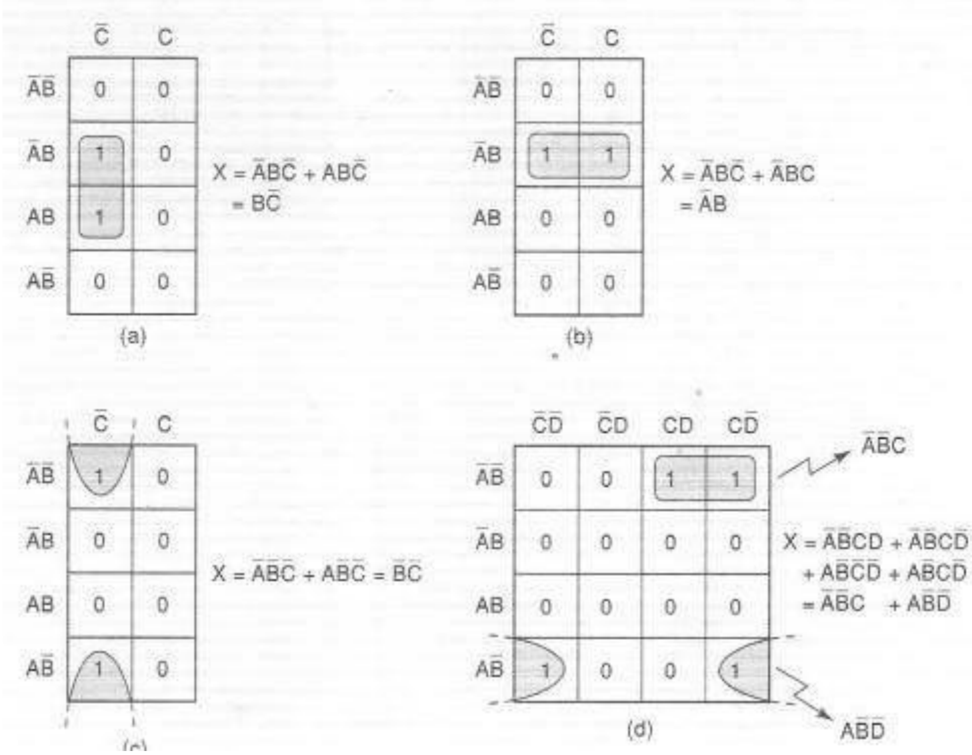
A	B	C	D	X
0	0	0	0	0
0	0	0	1	1 → $\bar{A}\bar{B}\bar{C}D$
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1 → $\bar{A}B\bar{C}D$
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1 → $AB\bar{C}D$
1	1	1	0	0
1	1	1	1	1 → ABCD

$$X = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + ABCD$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
$A\bar{B}$	0	0	0	0

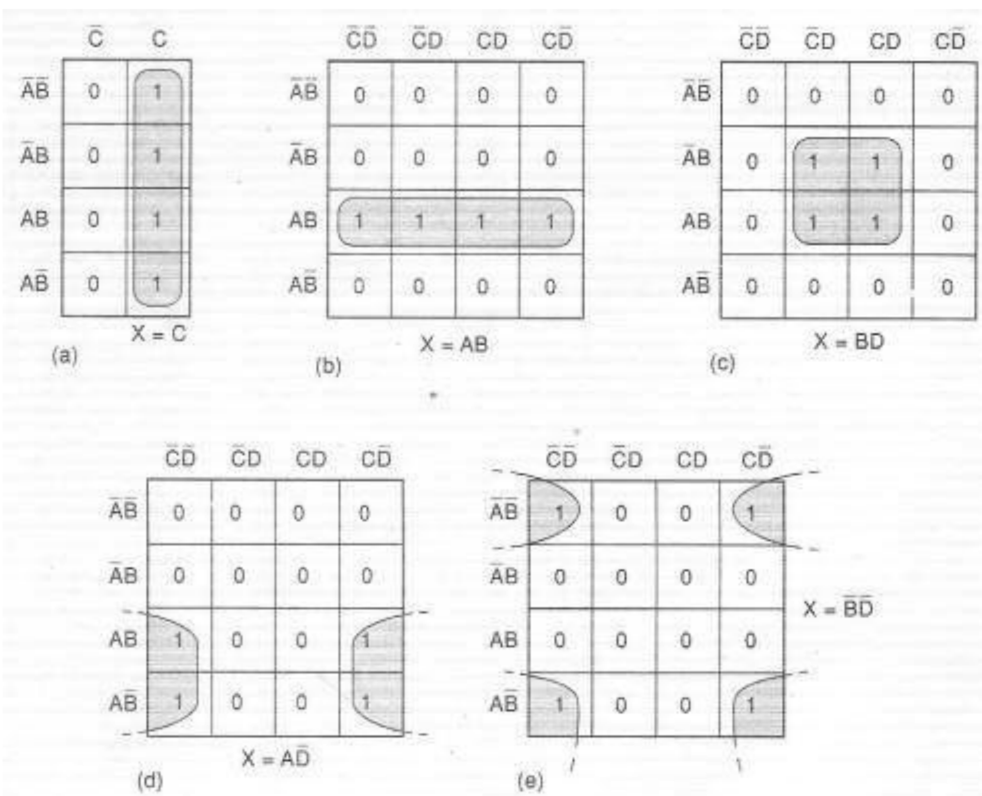
(c)

Doubles



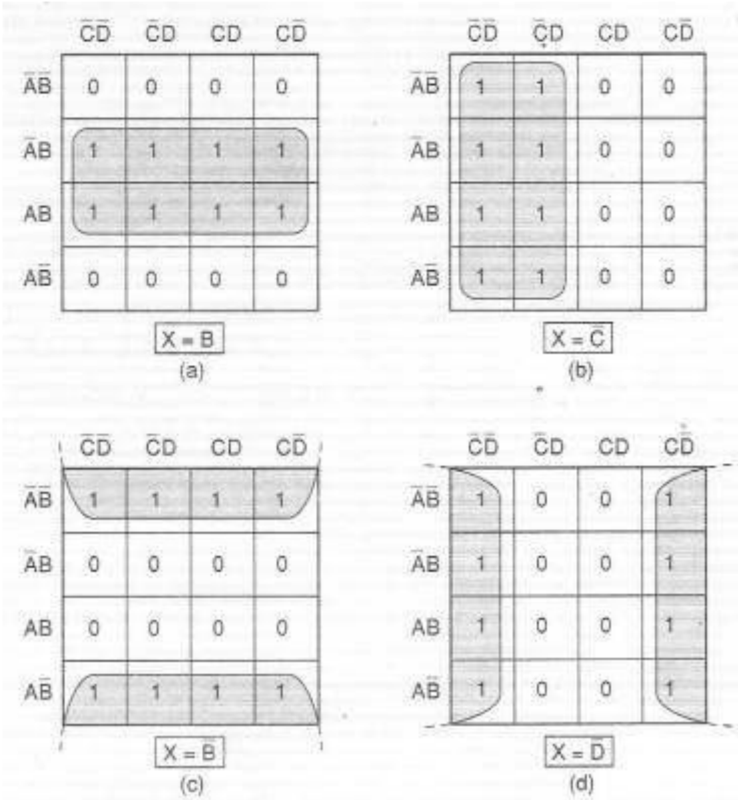
$X = \overline{A}B\overline{C} + AB\overline{C}$
$= B\overline{C}$
(a)
$X = \overline{A}B\overline{C} + \overline{A}BC$
$= \overline{A}B$
(b)
$X = \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C} = \overline{B}\overline{C}$
(c)
$\overline{A}\overline{B}C$
$X = \overline{A}\overline{B}CD + \overline{A}\overline{B}C\overline{D}$
$+ A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D}$
$= \overline{A}\overline{B}C \quad + A\overline{B}\overline{D}$
$A\overline{B}\overline{D}$
(d)

Quads



$X = C$
(a)
$X = AB$
(b)
$X = BD$
(c)
$X = A\overline{D}$
(d)
$X = \overline{B}\overline{D}$
(e)

Octets



Steps for K Map Reduction

1. Place 1s according to the Truth Table or logical expression and fill 0s at all the other positions
2. Locate the isolated 1s (That can not be combined) and circle them.
3. Locate 1s which form a double and encircle them in a group
4. Locate 1s which form a Quad and encircle them in a group
5. Locate 1s which form a Octet and encircle them in a group
6. After identifying, check if any 1s is yet to be encircled, if yes encircle them with each other or with the already encircled 1s by means of overlapping.
7. Note that the number of groups should be minimum
8. That any 1 can be included any number of times without affecting the expression
9. The redundant group has to be eliminated as it increases the number of gates required.

K Map Examples