CSE1003 Digital Logic and Design
Module 3
Combinational Circuits I
L3

DR. S.HEMAMALINI
PROFESSOR
SCHOOL OF ELECTRICAL ENGINEERING
VIT CHENNAI

Contents

4 hrs

Codes

- Codes are used extensively with computers to define alphanumeric characters and other information.

The commonly used binary codes are classified as:

1. Weighted and non-weighted codes
2. Numeric and alphanumeric codes
3. Error detecting and correcting codes
4. Self-complementary codes
5. Unit distance codes (Cyclic codes)
6. Sequential Codes
7. Reflective Codes

Binary-Coded-Decimal Code (8421 Code)

- If each digit of a decimal number is represented by its binary equivalent, the result is a code called binary-coded decimal.
- The 10 decimal digits 0 through 9 can be represented by their corresponding 4-bit binary numbers.
- It is a weighted code, with 8, 4, 2 and 1 representing the weights of different bits in the four-bit groups, starting from MSB and going towards LSB.
- BCD code is also known as 8421 code or natural binary code.
- It is also sequential. Therefore, it is useful for mathematical operations.

8 1000

12

0001 0010

Binary-Coded-Decimal Code (8421 Code)

- The four-bit binary numbers from 0000 through 1001 are used.
- The BCD code does not use the numbers 1010, 1011, 1100, 1101, 1110, and 1111.
- If any of the “forbidden” four-bit numbers ever occurs in a machine using the BCD code, it is usually an indication that an error has occurred.

Decimal Numbers		BCD Bit encoding
0		0000
1		0001
2	Valid	0010
3		0011
4	BCD code	0100
5		0101
6		0110
7		0111
8		1000
9		1001
10		1010
11		1011
12	Invalid	1100
13	BCD Code	1101
14		1110
15		1111

Binary-Coded-Decimal Code (8421 Code)

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

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

$874_{10} \rightarrow 100001110100_{BCD}$

23.15 (decimal)

0010 0011.0001 0101 (BCD)

Binary-Coded-Decimal Code (8421 Code)

Conversion of BCD code to decimal

- Start at the rightmost bit and break the code into groups of 4 bits.
- Write the decimal digit represented by each 4-bit group.

Convert the following BCD codes to decimal.

1000 0110	0011 0101 0001	1001 0100 0111 0000
$8\ 6_{10}$	351_{10}	9470_{10}

Binary-Coded-Decimal Code (8421 Code)

Convert 0110100000111001 (BCD) to its decimal equivalent.

$$\underbrace{0110}_{6}\ \underbrace{1000}_{8}\ \underbrace{0011}_{3}\ \underbrace{1001}_{9}$$

Convert the BCD number 011111000001 to its decimal equivalent.

$$\underbrace{0111}_{7}\ \underbrace{1100}_{\downarrow}\ \underbrace{0001}_{1}$$

The forbidden code group indicates an error in the BCD number.

BCD-to-Binary Conversion

- First, write the decimal equivalent of given BCD number and then convert it into binary equivalent.

Convert the following BCD code into its equivalent binary.

$$[00101001.01110101]_{BCD}$$

Given BCD number :	0010	1001	.	0111	0101
Decimal equivalent :	2	9	.	7	5

$$(29.75)_{10} = (11101.11)_2$$

Conversion of integer part:

	Quotient	Remainder	
$29 \div 2$	14	1	• LSB
$14 \div 2$	7	0	
$7 \div 2$	3	1	
$3 \div 2$	1	1	
$1 \div 2$	0	1	• MSB

Conversion of fractional part:

Multiplication	Integer part	
$0.75 \times 2 = 1.5$	1	• MSB
$0.5 \times 2 = 1.0$	1	• LSB

Drawbacks of BCD code

- BCD code is less efficient than pure binary-requires more bits.
- An N digit decimal number is represented by 4 x N bits in BCD code.
- The BCD code of 137 is 12 bits and the binary code of 137 is 8 bits; it shows that the BCD code is not efficient as compared to binary.
- The BCD code requires more space and time to transmit the information.
- Arithmetic operations are more complex than they are in pure binary form.

Comparison of BCD and binary

$$137_{10} = 10001001_2 \quad \text{(binary)}$$

8 bits

$$137_{10} = 0001\ 0011\ 0111 \quad \text{(BCD)}$$

12 bits

Excess-3 code

Decimal 0, 1, 2, 13, 14, 15

0 – 16

3 to 12. used

❖Excess-3 code is non-weighted and sequential code and it is useful for arithmetic operations.

❖The excess-3 code for a given decimal number is determined by adding ‘3’ to each decimal digit in the given number and then replacing each digit of the newly found decimal number by its four-bit binary equivalent.

❖The XS-3 code has six invalid states 0000, 0001, 0010, 1101, 1110 and 1111.

Decimal		Excess-3
0	+3 = 3	0011
1	+3 = 4	0100
2	+3 = 5	0101
3		0110
4		0111
5		1000
6		1001
7		1010
8		1011
9	+3 = 12	1100

Excess-3 code

0 1 0 1	0 1 1 0	• *BCD* code of 56
0 0 1 1	0 0 1 1	• Add 3 to each digit
1 1 1	1 1	• Carry
1 0 0 0	1 0 0 1	• Excess-3 code of (56)

- In excess-3 code, the N digit decimal is represented by 4 x N bits.
- For example, excess-3 code of 12 is 01000101 and there are 8 bits.
- The binary code of 12 is 1100 and there are 4 bits.
- This shows that the excess-3 code is not efficient as compared to binary.
- It requires more space and time to transmit the information.

Binary codes for decimal digits

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

Gray Codes

- Gray code belongs to a class of code known as minimum change code, in which a number changes by only one bit as it proceeds from one number to the next.
- Hence this code is not useful for arithmetic operations.
- This code finds extensive use for shaft encoders, in some types of analog-to-digital converters, etc.
- Gray code is reflected code.
- The Gray code is not a weighted code.

Decimal numbers	*Binary code*	*Gray code*
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Conversion of a Binary Number into Gray Code

(i) the MSB of the Gray code is the same as the MSB of the binary number;

(ii) the second bit next to the MSB of the Gray code equals the Ex-OR of the MSB and second bit of the binary number; it will be 0 if there are same binary bits or it will be 1 for different binary bits;

(iii) the third bit for Gray code equals the exclusive-OR of the second and third bits of the binary number, and similarly all the next lower order bits follow the same mechanism.

$$(74)_{10} = (1001010)_2$$

$$\begin{array}{lccccccccccccc}
\text{Binary} & 1 & \rightarrow\oplus\rightarrow & 0 & \rightarrow\oplus\rightarrow & 0 & \rightarrow\oplus\rightarrow & 1 & \rightarrow\oplus\rightarrow & 0 & \rightarrow\oplus\rightarrow & 1 & \rightarrow\oplus\rightarrow & 0 \\
 & \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow \\
\text{Gray} & 1 & & 1 & & 0 & & 1 & & 1 & & 1 & & 1
\end{array}$$

$$\text{Gray code of } 74 = 1101111$$

Gray-to-Binary Code Conversion

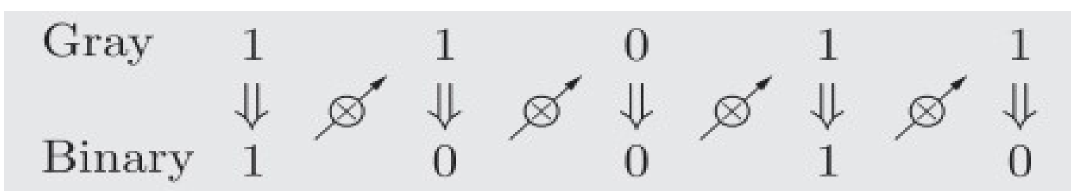
1. The most significant bit (left most bit) of the equivalent binary code is the same as the MSB of the given Gray code.

2. Add the MSB of the binary to the next significant bit of the Gray code, note the sum and ignore the carry.

3. Add the 2nd bit of the binary to the 3rd bit of the Gray; the 3rd bit of the binary to the 4th bit of the Gray code, and so on, each time note the sum and ignore the carry.

4. Continue above step till all Gray bits are used. This sequence of bits is the binary equivalent of the Gray code number.

For example the conversion of Gray code 11011 is shown as below.



Alphanumeric Codes

- A computer must be capable of handling nonnumeric information like numbers, letters, and special characters. These codes are classified as alphanumeric or character codes.
- An alphanumeric code is a binary code of a group of elements consisting of ten decimal digits, the 26 letters of the alphabet (both in uppercase and lowercase), and a certain number of special symbols such as #, /, &, %, etc.

ASCII code

- American Standard Codes for Information Interchanging (ASCII) is the most widely used alphanumeric code.
- This is basically a 7-bit code and so, it has $2^7 = 128$ possible code groups.
- The ASCII code can be used to encode both the lowercase and uppercase characters of the alphabet (52 symbols) and some special symbols as well, in addition to the 10 decimal digits.
- This code is used to exchange the information between input/output device and computers, and stored into the memory.

ASCII code

$b_7 b_6 b_5$ $b_4 b_3 b_2 b_1$

A 100 0001

b 110 0010

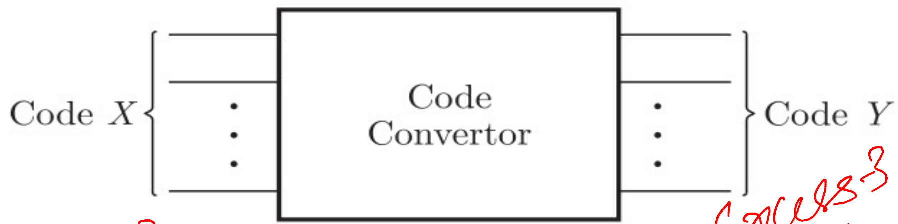
American Standard Code for Information Interchange (ASCII)

	$b_7 b_6 b_5$							
$b_4 b_3 b_2 b_1$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	¦
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

CODE CONVERTERS

Combinational logic ckt.

- A code converter is a combinational logic circuit which accepts the input information in one binary code, converts it and produces an output into another binary code.
- To convert from one binary code A to binary code B, the input lines must provide the bit combination of elements as specified by A and the output lines must generate the corresponding bit combinations of code B.
- A combinational circuit consisting of logic gates performs this transformation operation.



BCD

Excess3 code

Block diagram of a code convertor

Common conversions

- ❑ BCD to 7-segment
- ❑ BCD to binary
- ❑ Binary to BCD
- ❑ Binary to Gray code
- ❑ Gray code to binary

BCD-to-Excess-3 Code converter

A, B, C, D
Input variables

W, X, Y, Z
o/p variables

- Excess-3 code is a modified BCD code. It is obtained by adding 3 to each BCD code.
- Note that the input combinations 1010, 1011, 1100, 1101, 1110, and 1111 are invalid in BCD. So they are treated as don't cares.

10 – 15

Decimal	*BCD code*				*Excess-3 code*			
Equivalent	*A*	*B*	*C*	*D*	*W*	*X*	*Y*	*Z*
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

K-Maps for excess-3 codes

Y

AB \ CD	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

$W = A + BD + BC$

$Y = C'D' + CD$

AB \ CD	00	01	11	10
00	1		1	
01	1		1	
11	X	X	X	X
10	1		X	X

AB \ CD	00	01	11	10
00		1	1	1
01	1			
11	X	X	X	X
10		1	X	X

$X = BC'D' + B'D + B'C$

$Z = D'$

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

BCD-to-Excess-3 Code converter

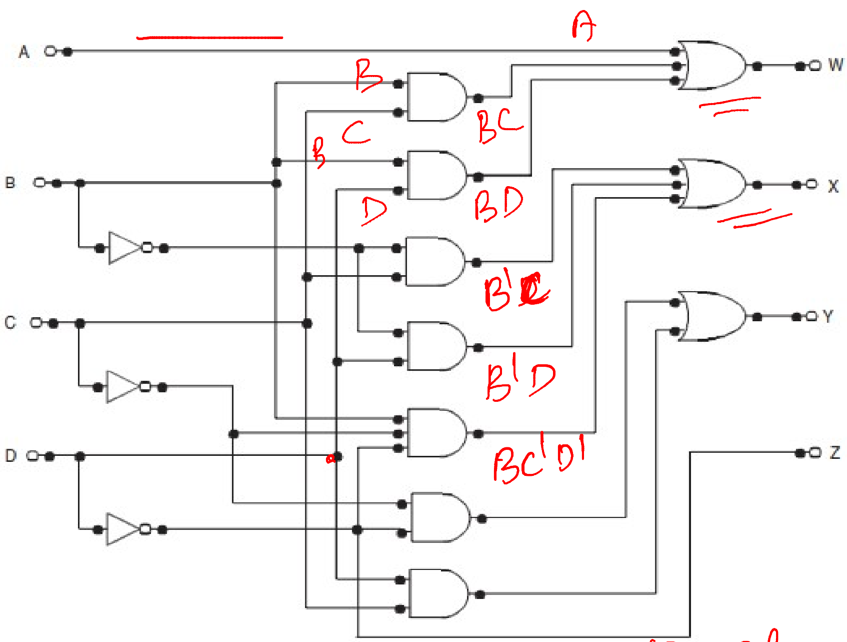
$W = A + BD + BC$

$= A + B(C+D)$

$X = BC'D' + B'D + B'C$

$= B'(C+D) + BC'D'$

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

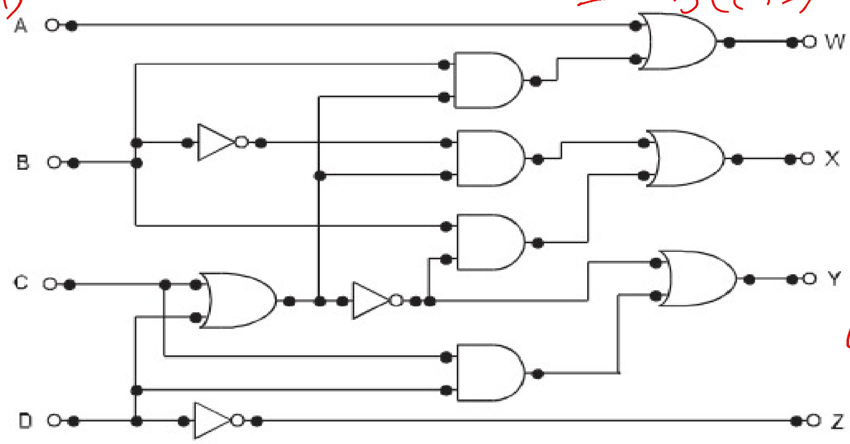


$A+BD+BC$

$= BC'D' + B'D + B'C$

$= D'$

7-AND 3-OR



4-AND

4-OR

$Y = C'D' + CD = CD + (C+D)'$

Excess-3-to-BCD Code Converter

Decimal Equivalent	Excess-3 code				BCD code			
	W	X	Y	Z	A	B	C	D
0	0	0	1	1	0	0	0	0
1	0	1	0	0	0	0	0	1
2	0	1	0	1	0	0	1	0
3	0	1	1	0	0	0	1	1
4	0	1	1	1	0	1	0	0
5	1	0	0	0	0	1	0	1
6	1	0	0	1	0	1	1	0
7	1	0	1	0	0	1	1	1
8	1	0	1	1	1	0	0	0
9	1	1	0	0	1	0	0	1

K-Maps for Excess-3-to-BCD Code Converter

WX \ YZ	00	01	11	10
00	X	X		X
01				
11	1	X	X	X
10			1	

$A = WX + WYZ$

WX \ YZ	00	01	11	10
00	X	X		X
01			1	
11		X	X	X
10	1	1		1

$B = XYZ + X'Y' + X'Z'$

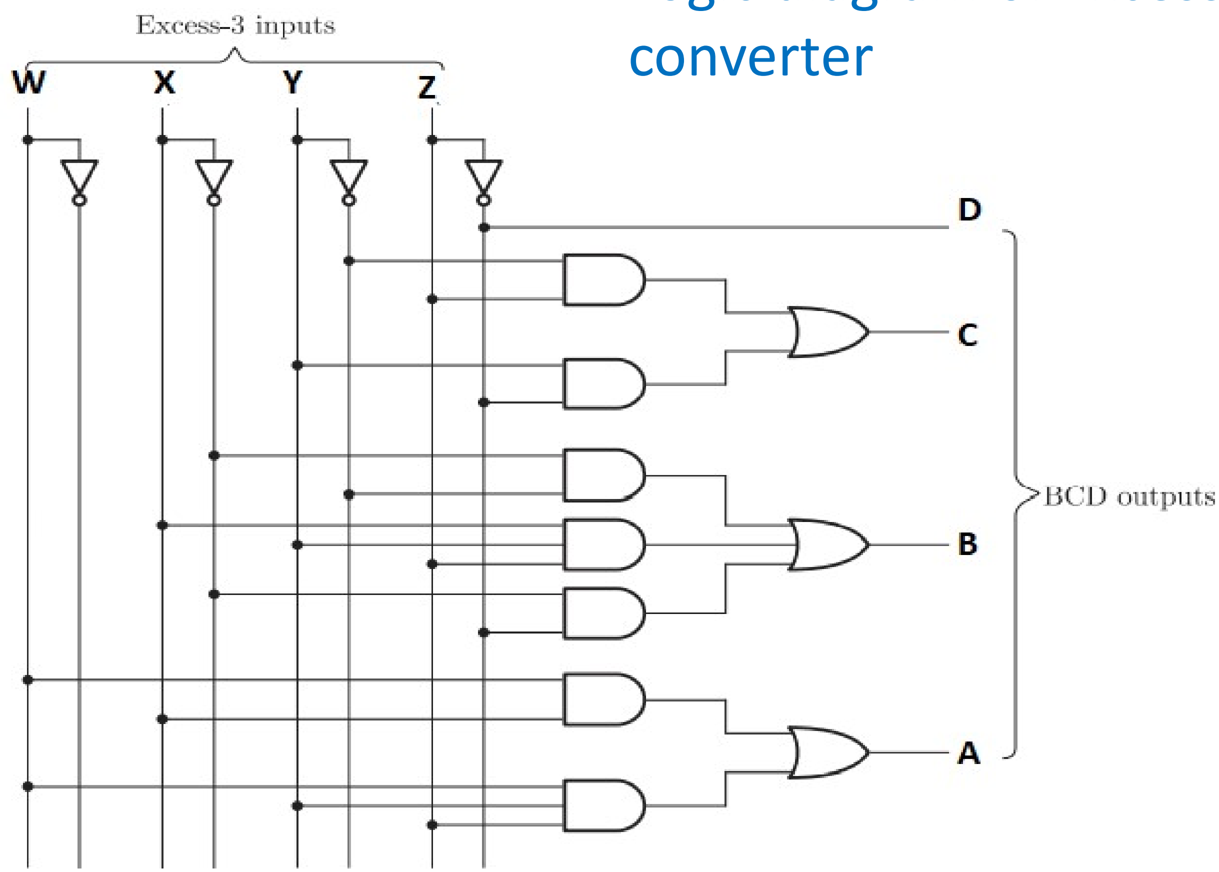
WX \ YZ	00	01	11	10
00	X	X		X
01		1		1
11		X		X
10		1		1

$C = Y'Z + YZ'$

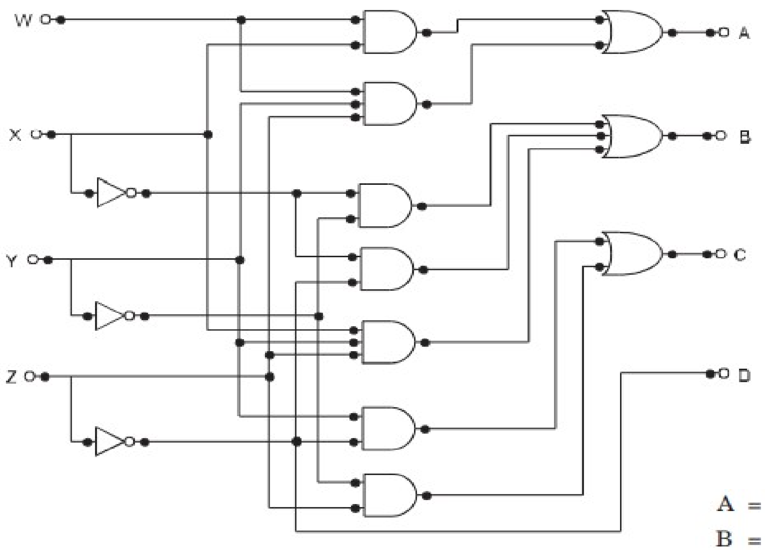
WX \ YZ	00	01	11	10
00	X	X		X
01	1			1
11	1	X	X	X
10	1			1

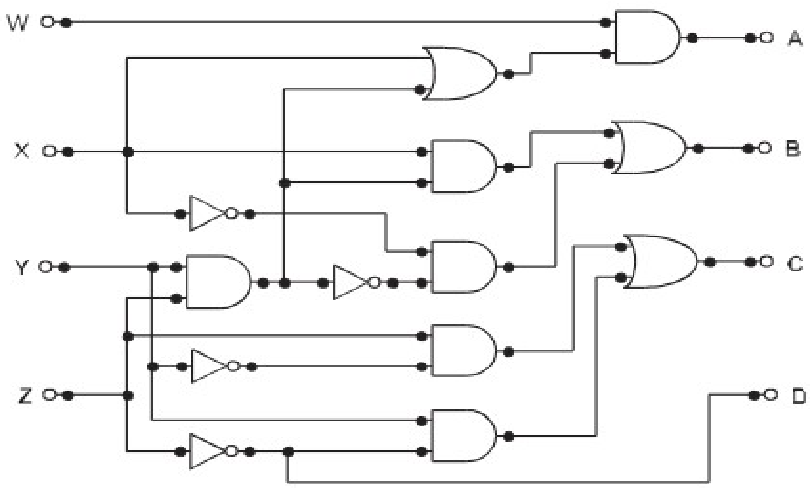
$D = Z'$

Logic diagram of Excess-3 to-BCD code converter



Logic diagram of an Excess-3-to-BCD converter





$$A = WX + WYZ = W(X + YZ)$$

$$B = X'Y' + X'Z' + XYZ = X'(Y' + Z') + XYZ = X'(YZ)' + XYZ$$

$$C = Y'Z + YZ'$$

$$D = Z'$$