Code Converters

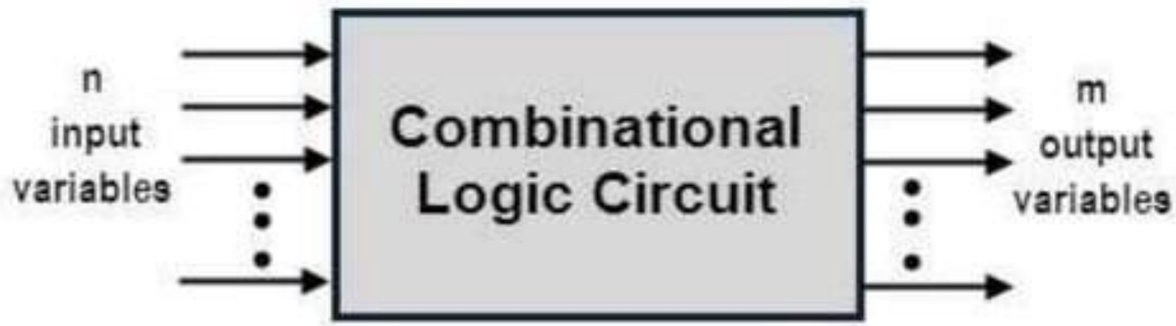
Combinational circuits

- A combinational logic circuit is one in which the present state of the combination of the logic inputs decides the output .
- The term combination logic means combining of two or more logic gates to form a required function where the output at a given time depends only on the input.

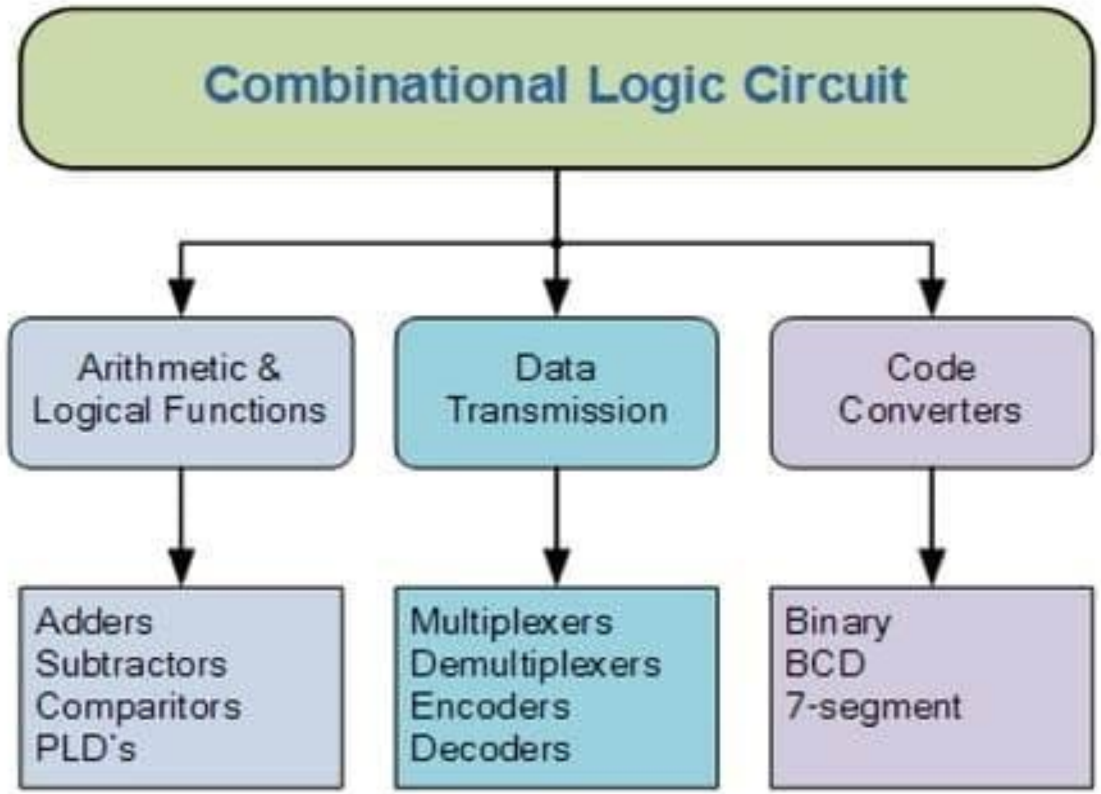
Characterstics:

- The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
- The combinational circuit do not use any memory. The previous state of input does not have any effect on the present state of the circuit.
- A combinational circuit can have an n number of inputs and m number of outputs.

- The required output data is obtained from this process by transforming the binary information given at the input.



Classification of Combinational Logic



Code Converters:

- Code is a symbolic representation of discrete information.
- A converter that changes coded information to a different code system is called code converter.
- Numbers are usually coded in one form or another so as to represent or use it as required.

Let’s discuss the conversion of various codes from one form to other.

- **Binary to gray code conversion**
- **Gray code to binary conversion**
- **DECIMAL TO BCD CODE CONVERTER**
- **BCD TO EXCESS-3 CODE CONVERTER**

Binary To Gray Code conversion:

Binary to gray code conversion is a very simple process. There are several steps to do this types of conversions. Steps given below elaborate on the idea on this type of conversion.

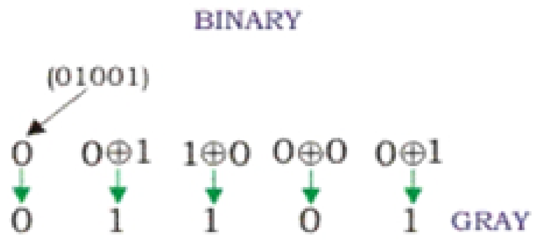
(1) The M.S.B. of the gray code will be exactly equal to the first bit of the given binary number.

(2) Now the second bit of the code will be exclusive-or of the first and second bit of the given binary number, i.e if both the bits are same the result will be 0 and if they are different the result will be 1.

(3)The third bit of gray code will be equal to the exclusive-or of the second and third bit of the given binary number. Thus the **Binary to gray code conversion** goes on. One example given below can make your idea clear on this type of conversion.

Explanation:

Now concentrate on the example where the M.S.B. of the binary is 0 so for it will be 0 for the most significant gray bit. Next, the XOR of the first and the second bit is done. The bits are different so the resultant gray bit will be 1. Again move to the next step, XOR of second and third bit is again 1 as they are different. Next, XOR of third and fourth bit is 0 as both the bits are same. Lastly the XOR of fourth and fifth bit is 1 as they are different. That is how the result of binary to gray code conversion of 01001 is done whose equivalent gray code is 01101.



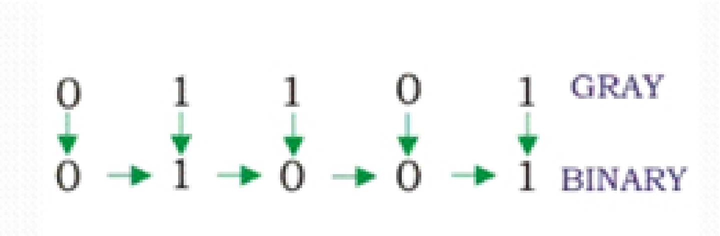
Gray code to binary conversion :

Gray code to binary conversion is again very simple and easy process. Following steps can make your idea clear on this type of conversions.

(1) The M.S.B of the binary number will be equal to the M.S.B of the given gray code.

(2) Now if the second gray bit is 0 the second binary bit will be same as the previous or the first bit. If the gray bit is 1 the second binary bit will alter. If it was 1 it will be 0 and if it was 0 it will be 1.

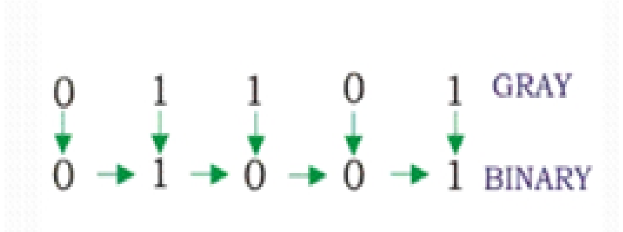
(3) This step is continued for all the bits to do **Gray code to binary conversion**. One example given below will make your idea clear.



Explanation:

The M.S.B of the binary will be 0 as the M.S.B of gray is 0.

Now move to the next gray bit. As it is 1 the previous binary bit will alter i.e. it will be 1, thus the second binary bit will be 1. Next look at the third bit of the gray code. It is again 1 thus the previous bit i.e. the second binary bit will again alter and the third bit of the binary number will be 0. Now, 4th bit of the given gray is 0 so the previous binary bit will be unchanged, i.e. 4th binary bit will be 0. Now again the 5th grey bit is 1 thus the previous binary bit will alter, it will be 1 from 0. Therefore the equivalent Binary number in case of gray code to binary conversion will be (01001)



DECIMAL TO BCD CODE CONVERTER

BCD TO EXCESS-3 CODE CONVERTER

DECIMAL TO BCD CODE CONVERTER

The BCD Code

Binary Coded Decimal (BCD) code is used to represent decimal digits in binary.

BCD code is a 4-bit binary code the first 10 combinations represent the decimal digits 0 to 9.

Decimal Numerals	Binary Numerals
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

DECIMAL TO BCD CODE CONVERTER

The remaining six 4 bit (Figure 1)combinations 1010, 1011, 1100, 1101, 1110 and 1111 are considered to be invalid and do not exist.

Decimal	BCD				
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	
-	1	0	1	0	
-	1	0	1	1	
-	1	1	0	0	Unused
-	1	1	0	1	
-	1	1	1	0	
-	1	1	1	1	

BCD TO EXCESS-3 CODE CONVERTER

The Excess-3 BCD system is formed by adding 0011 (3) to each BCD value as in Table . For example, the decimal number 7, which is coded as 0111 in BCD, is coded as 0111+0011=1010 in Excess-3 BCD.

Binary Numerals	Excess-3
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

BCD Excess-3

Decimal	BCD 8 4 2 1	Excess-3 BCD + 0011
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

Part one Pages 53,54,55

5.1.2 8421-BCD/Three-Excess Code Converter

The following example shows the conversion of the 8421 code into three-excess code.

The following OR normal forms apply for the outputs e ... h:

$$e = 0 \vee 2 \vee 4 \vee 6 \vee 8$$

$$f = 0 \vee 3 \vee 4 \vee 7 \vee 8$$

$$g = 1 \vee 2 \vee 3 \vee 4 \vee 9$$

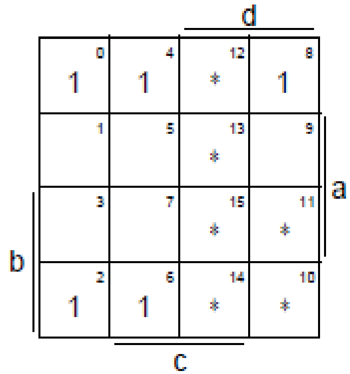
$$h = 5 \vee 6 \vee 7 \vee 8 \vee 9$$

These OR normal forms are simplified with the aid of KV diagrams (figs. 5.1.2.1 ... 5.1.2.4, page 54).

According to the simplified equations, the circuit can be set up as shown in fig. 5.1.2.5 (page 54).

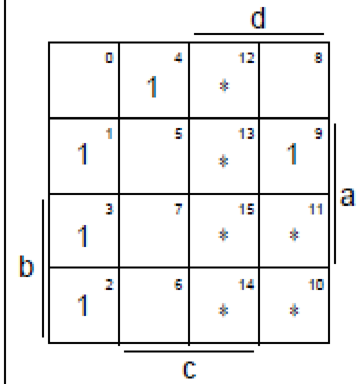
Weighting	Origin code 8421-BCD				Destination code 3-excess			
	d	c	b	a	h	g	f	e
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0
10	1	0	1	0	Don't care fields			
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

Table 5.1.2.1



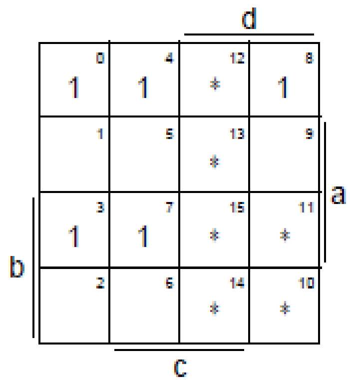
$e = \overline{a}$

Fig. 5.1.2.1 KV diagram for e



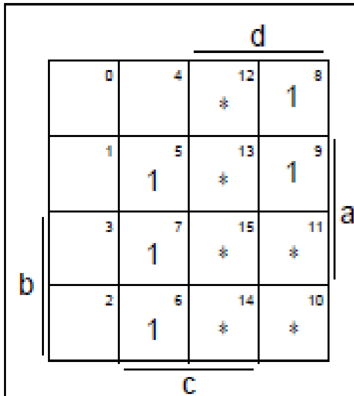
$g = b\,\overline{c} \vee a\,\overline{c} \vee \overline{a}\,\overline{b}\,c$

Fig. 5.1.2.3 KV diagram for g



$f = \overline{a}\,\overline{b} \vee a\,b$

Fig. 5.1.2.2 KV diagram for f



$h = d \vee a\,c \vee b\,c$

Fig. 5.1.2.4 KV diagram for h

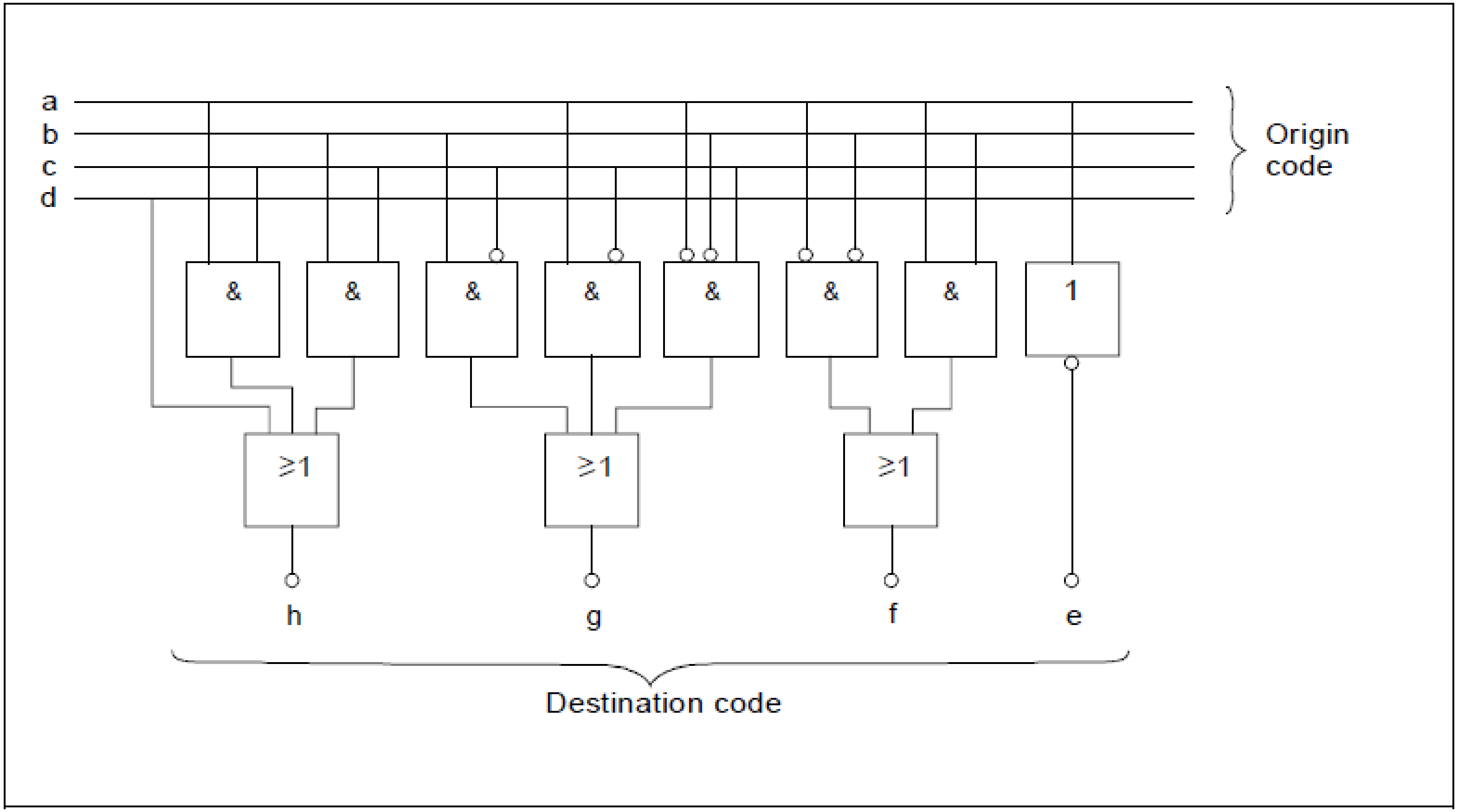


Fig. 5.1.2.5 Circuit for an 8421-BCD / three-excess code converter

If only NAND elements are available for the circuit setup, they must be converted accordingly:

$$e = \overline{a}$$

$$f = \overline{\overline{\overline{a}\,\overline{b}} \wedge \overline{a\,b}}$$

$$g = \overline{\overline{b\,\overline{c}} \wedge \overline{a\,\overline{c}} \wedge \overline{a\,\overline{b}\,\overline{c}}}$$

$$h = \overline{\overline{d} \wedge \overline{a\,c} \wedge \overline{b\,c}}$$

Code converters for any conversion task can be calculated by this method.

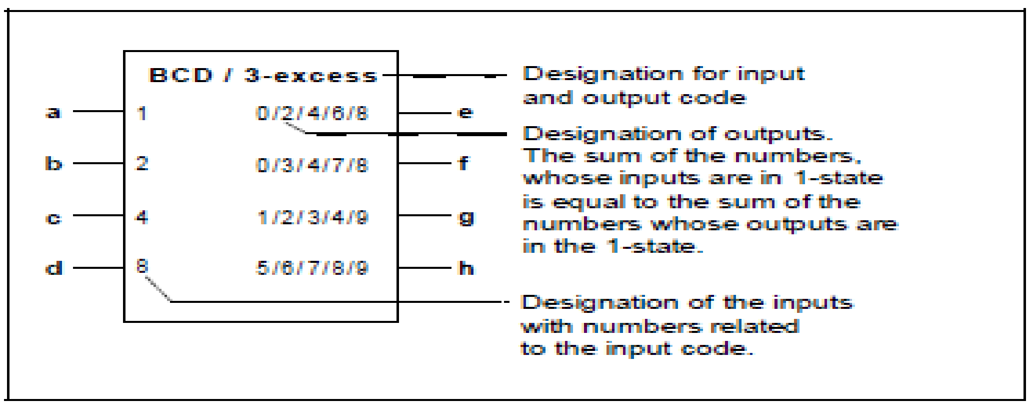


Fig. 5.1.2.6 Circuit symbol

Part two pages 56,57

5.2.1 8421-BCD / Decimal Code Converter

❒ **Experiment 1:**

Dec. value	Inputs: BCD code				Outputs: Decimal code									
	2^3	2^2	2^1	2^0										
	d	c	b	a	e	f	g	h	i	j	k	l	m	n
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5	0	1	0	1	0	0	0	0	0	1	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	1	0	0	0
7	0	1	1	1	0	0	0	0	0	0	0	1	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	1	0
9	1	0	0	1	0	0	0	0	0	0	0	0	0	1

Table 5.2.1.1

The following simplifications are possible under consideration of the pseudo-tetrades:

$$e = \bar{a}\,\bar{b}\,\bar{c}\,\bar{d} \qquad f = a\,\bar{b}\,\bar{c}\,\bar{d}$$

$$g = \bar{a}\,b\,\bar{c} \qquad h = a\,b\,\bar{c}$$

$$i = \bar{a}\,\bar{b}\,c \qquad j = a\,\bar{b}\,c$$

$$k = \bar{a}\,b\,c \qquad l = a\,b\,c$$

$$m = \bar{a}\,d \qquad n = a\,d$$

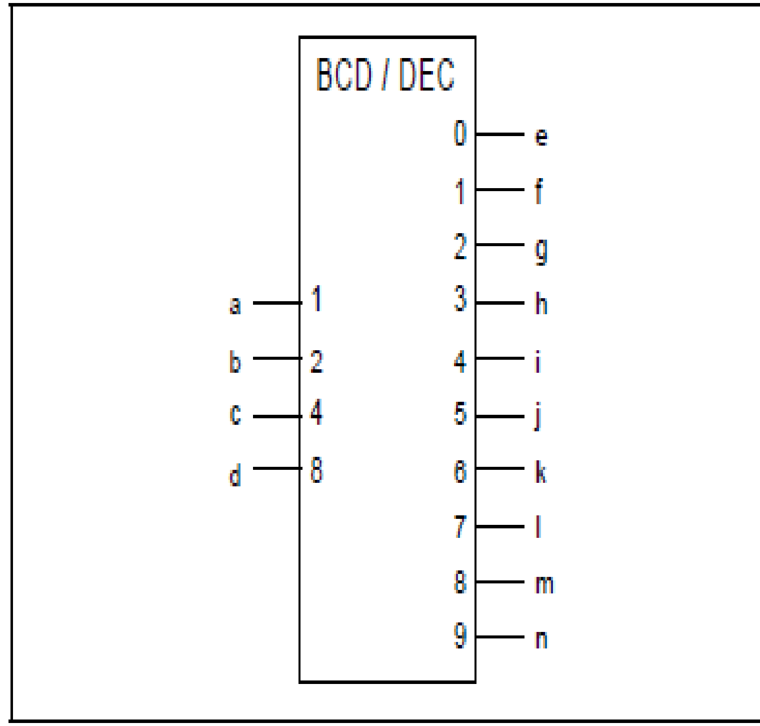


Fig. 5.2.1.2 Circuit symbol

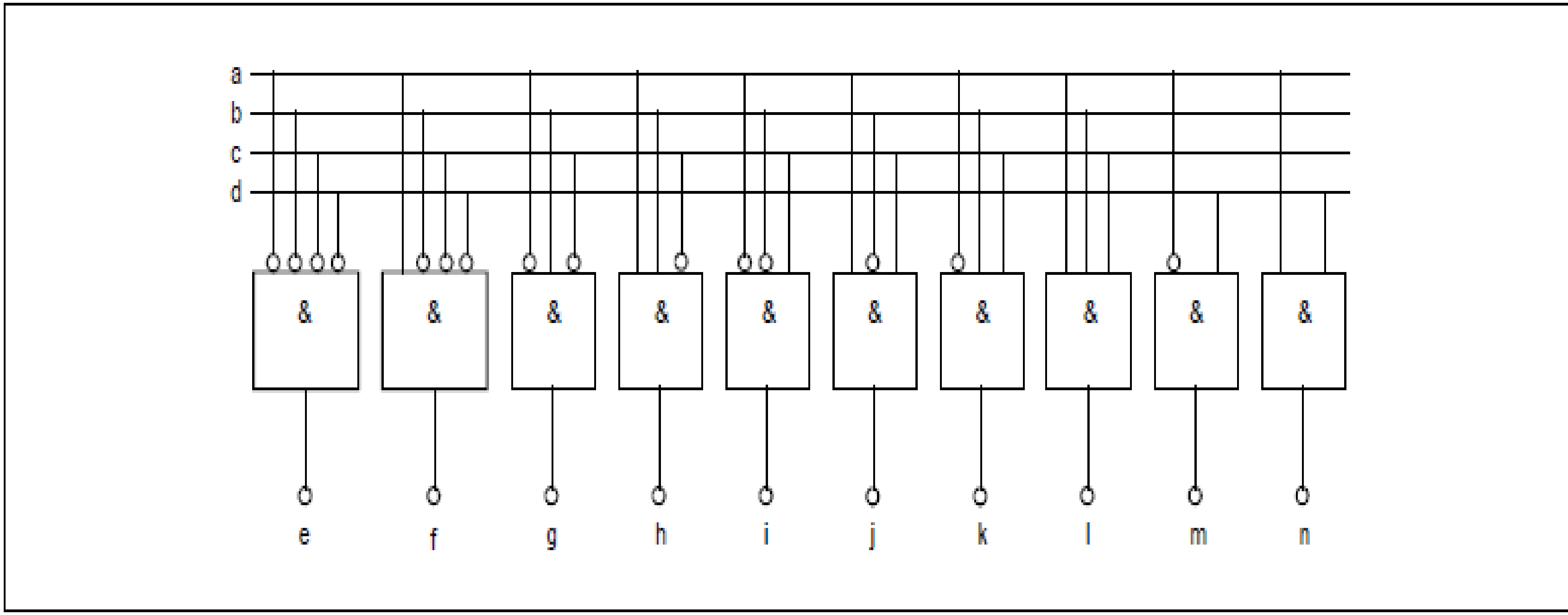


Fig. 5.2.1.1 Circuit for an 8421-BCD / decimal code converter

Part three pages 61,62

Coding circuits

Circuit 1

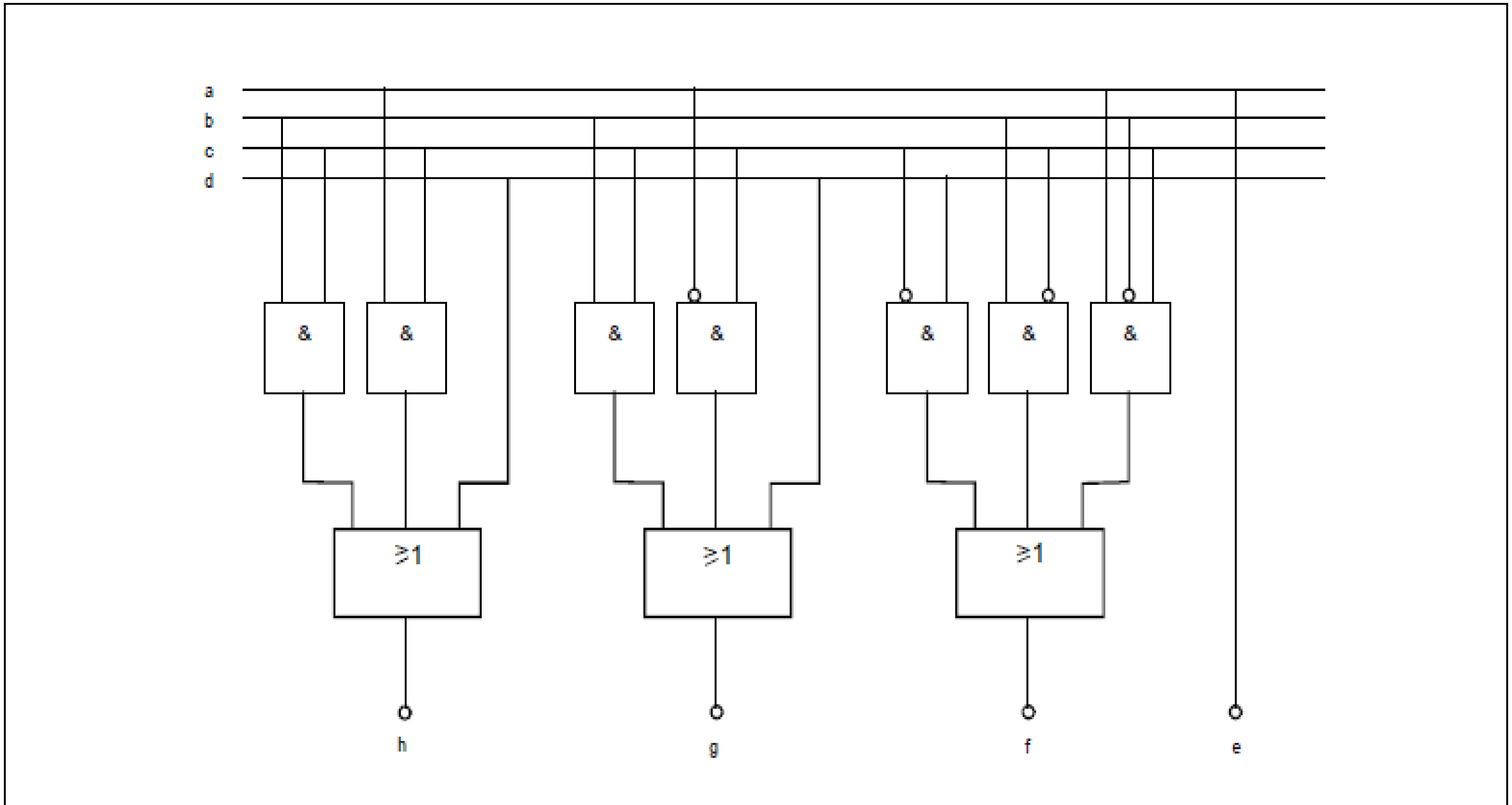


Fig. 5.2.3.1 Circuit 1

Circuit 2

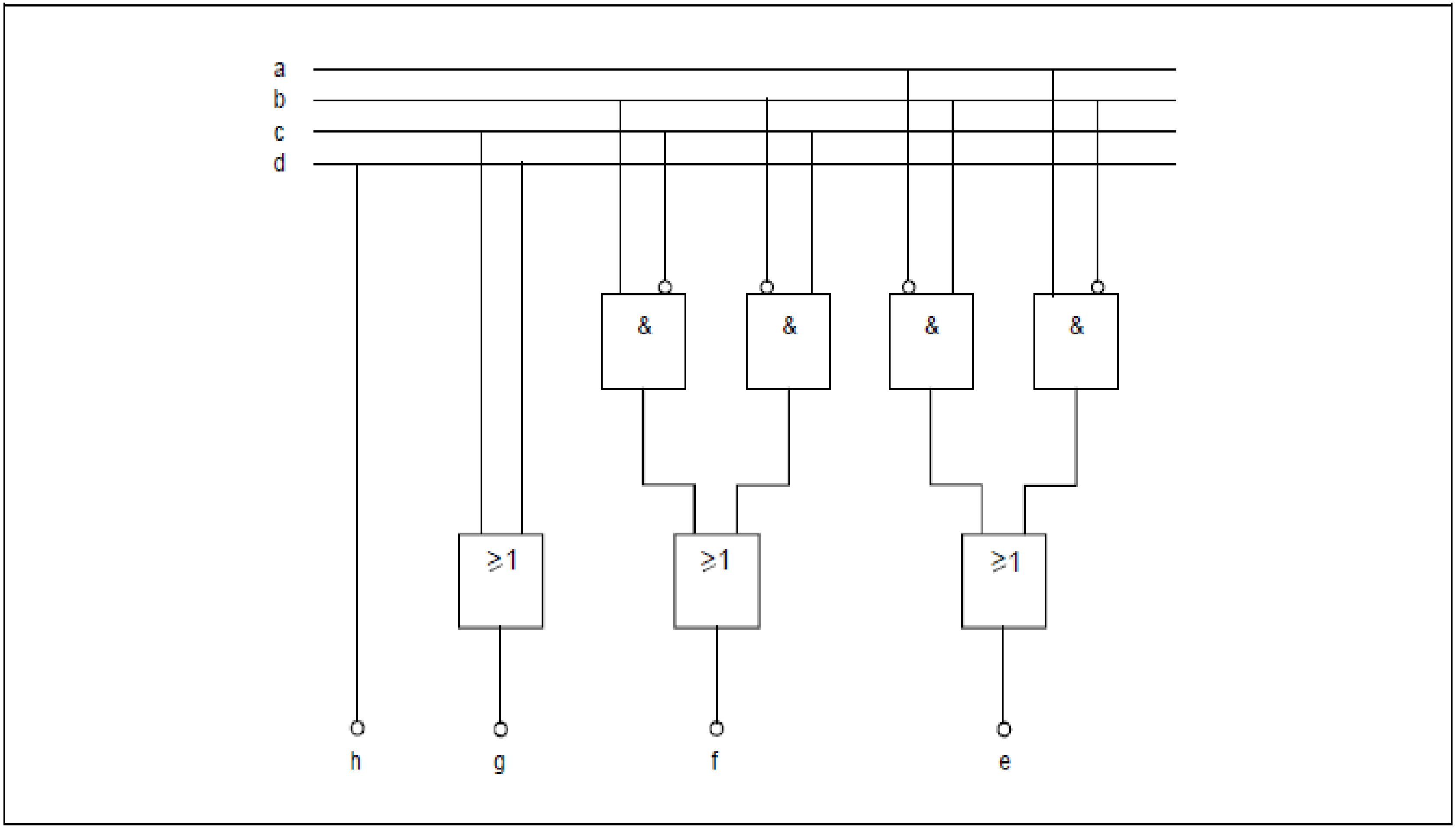


Fig. 5.2.3.2 Circuit 2

8421-BCD code				Circuit 1				Circuit 2			
d	c	b	a	h	g	f	e	h	g	f	e

Table 5.2.3.1

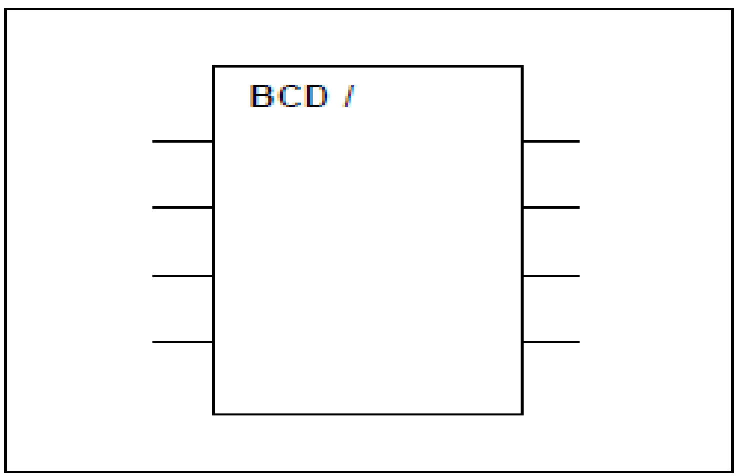


Fig. 5.2.3.3 Circuit symbol for circuit 1

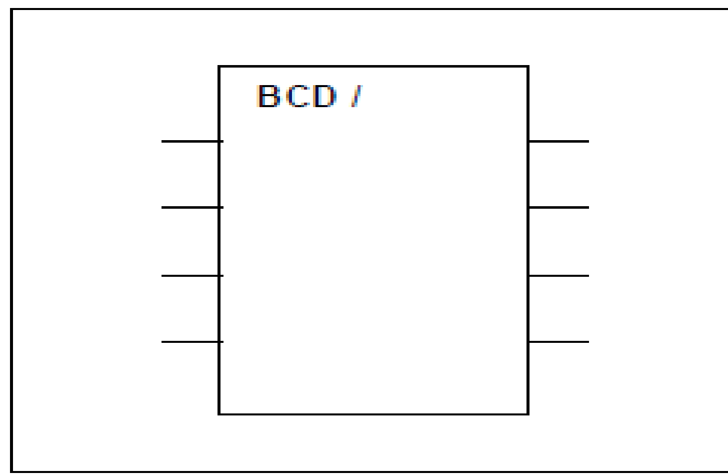


Fig. 5.2.3.4 Circuit symbol for circuit 2

MANY THANKS FOR YOUR ATTENTION