PROCEEDINGS OF
7th INTERNATIONAL CONFERENCE
ON
COMPUTER AND INFORMATION TECHNOLOGY

26-28 DECEMBER 2004, DHAKA, BANGLADESH

ICCIT

Organized by

BRAC University
Dhaka, Bangladesh

Supported by

Ahsanullah University of Science and Technology, Bangladesh
Chittagong University of Engineering and Technology, Bangladesh
Daffodil University, Bangladesh
East West University, Bangladesh
Independent University of Bangladesh, Bangladesh
International Islamic University Chittagong, Bangladesh
Islamic University of Technology, Bangladesh
North South University, Bangladesh
Prime University, Bangladesh

7th INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY
ICCIT 2004, 26-28 DECEMBER 2004, DHAKA, BANGLADESH

Designing 14-Segment Display for English Alphabet

Sk. Alamgir Hossain
Computer Science and Engineering Discipline
Khulna University
Khulna-9208, Bangladesh
hira_ku2003@yahoo.com

Md. Iqbal Aziz Khan
Computer Science and Engineering Discipline
Khulna University
Khulna-9208, Bangladesh
iqbal_aziz_khan@yahoo.com

Abstract

One of the simplest and most prevalent methods for displaying numerical digits uses a 7-segment formation to produce the decimal characters 0 through 9. A method for 14-segment formation system to display English Alphabet characters has been proposed in this paper. The 14-segment is arranged in such a way that these segments can characterize the entire English Alphabet. There are at most 26 Alphabet patterns to be displayed in English language and 5 bits are used to represent each of them. For each pattern particular segments are activated. A number of switching circuits and their respective combinational logic have been designed in order to represent each pattern.

Keywords

14-segment display, combinational logic, switching circuit, English alphabet, non-overlap.

INTRODUCTION

Presents days English alphabet are represented by dot matrix system. For a single English character a large number of dots are required. So this process is required large storage and time. So in our propose 14-segments for displaying English alphabet is better than the present dot matrix system.

In this paper first of all a grid structure consisting of 14-segments has been discovered so that the capital letters of English language be displayed by this segments. As there are 26 characters in English language so we required 5-bit inputs to represent each alphabet. After calculating which segments are activated for a particular character, appropriate logic circuit has been derived in order to display each English character.

SEGMENTATION

The proposed model of 14-segments for English alphabet is depicted in Figure 1.1. 14 non-overlapping segments of this model are structured to represent individual capital letters (A-Z) explicitly.

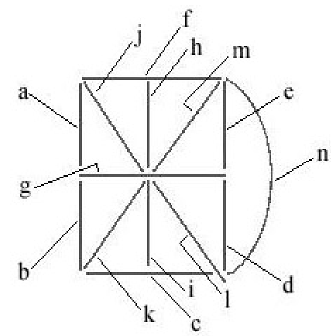


Figure 1.1. Model of 14-segments for English alphabet

To represent letter ‘A’, the segments a,b,d,e,f,g need to be activated. To represent ‘B’, the segments a,b,c,d,e,f and g need to be activated. Similarly for ‘C’ segments a,b,c and f. For ‘D’ segments a,b,c,f and,n. For ‘E’ segments a,b,c,f and g. For ‘F’ segments a,b,f and g. For ‘G’ segments a,b,c,d,f and g. For ‘H’ segments a,b,d,e and g. For ‘I’ segments d and e. For ‘J’ segments c, d and e. For ‘K’ segments a,b,k,l and m. For ‘L’ segments a,b and c. For ‘M’ segments a,b,d,e,j and m. For ‘N’ segments a,b,d,e,j and l. For ‘O’ segments a,b,c,d,f and g. For ‘P’ segments a,b,e,f and g. For ‘Q’ segments a,b,c,d,f and l. For ‘R’ segments a,b,e,f,g and l. For ‘S’ segments a,c,d,f and g. For ‘T’ segments f,h and i. For ‘U’ segments a,b,c,d and e. For ‘V’ segments j and m. For ‘W’ segments a,b,d,e,k and l. For ‘X’ segments j,k,l and m. For ‘Y’ segments i,j and m. For ‘Z’ segments c,f,k and m.

We know that for seven segment display , there is a seven segment display driver (IC 7447). Similarly we need a display driver for displaying English letters. For displaying the English letters the corresponding truth table of 5-bit inputs (A0,A1,A2,A3,A4) and 14 outputs (a,b,c,d,e,f,g,h,i,j,k,l,m,n) are shown in table 2.1.

Table 2.1. Truth table for 14 segments

Decimal Number	A5	A4	A3	A1	A0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	Letters Display
0	0	0	0	0	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	A
1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	B
2	0	0	0	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	C
3	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	1	D
4	0	0	1	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	E
5	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	F
6	0	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	0	0	0	G
7	0	0	1	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	H
8	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	I
9	0	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	J
10	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	K
11	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	L
12	0	1	1	0	0	1	1	0	1	1	0	0	0	0	1	0	0	1	0	M
13	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	N
14	0	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	O
15	0	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	P
16	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	0	0	Q
17	1	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	1	0	0	R
18	1	0	0	1	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	S
19	1	0	0	1	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	T
20	1	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	U
21	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	V
22	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	W
23	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	X
24	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	Y
25	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	1	0	Z
26	1	1	0	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
27	1	1	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
28	1	1	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
29	1	1	1	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
30	1	1	1	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
31	1	1	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

The combination from 26 to 31 is don't-care combination. Here 'x' represent "don't-care in Table 2.1.

DERIVATION OF SWITCHING FUNCTION

The following functions represent each segment. 'Σ' indicate, for these input combinations the value of that particular segment is 'on'/1 and 'ΣØ' indicate the output of that segment is "don't care" i.e. 'on'/off'.

a=Σ(0,1,2,3,4,5,6,7,10,11,12,13,14,15,16,17,18,20,22)+Σ ø (26,27,28,29,30,31)

b=Σ(0,1,2,3,4,5,6,7,10,11,12,13,14,15,16,17,20,22)+Σø(26,27,28,29,30,31)

c=Σ(1,2,3,4,6,9,11,14,16,18,20,25)+Σø(26,27,28,29,30,31)

d=Σ(0,1,6,7,8,9,12,13,14,16,18,20,22)+ Σø(26,27,28,29,30,31)

e=Σ(0,1,7,8,9,12,13,14,15,16,17,20,22) +Σø(26,27,28,29,30,31)

f=Σ(0,1,2,3,4,5,6,14,15,16,17,18,19,25) +Σø(26,27,28,29,30,31)

g=Σ(0,1,4,5,6,7,15,17,18) +Σø(26,27,28,29,30,31)

h=Σ (19) +Σø (26, 27, 28, 29, 30, 31)

i=Σ (19, 24) +Σø (26, 27, 28, 29, 30, 31)

j=Σ(12,13,21,23,24) +Σø(26,27,28,29,30,31)

k=Σ(10,22,23,25) +Σø(26,27,28,29,30,31)

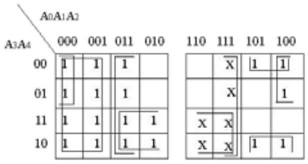
l=Σ(10,13,16,17,22,23) +Σø(26,27,28,29,30,31)

m=Σ(10,12,21,23,24,25) +Σø(26,27,28,29,30,31)

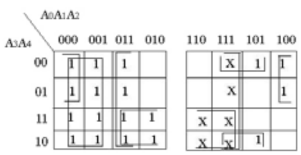
n=Σ (3) +Σø (26, 27, 28, 29, 30, 31)

MINIMIZATION OF SWITCHING FUNCTION

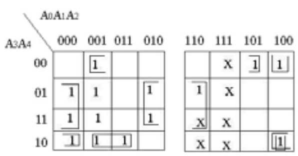
The minimization procedures of five variables using Karnaugh map method for each segments (a - n) are shown in Figure 1.2.



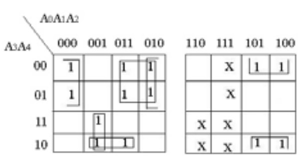
a= $A_0'A_1' + A_1A_2 + A_1A_3 + A_1'A_2'A_3' + A_0A_1'A_4'$



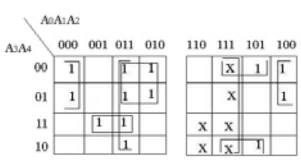
b= $A_0'A_1' + A_1A_2 + A_1A_3 + A_1'A_2'A_3' + A_0A_2A_4'$



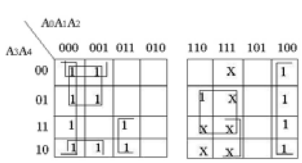
c= $A_1A_2'A_4 + A_0'A_2'A_4 + A_1'A_2A_3'A_4' + A_1'A_2'A_3A_4' + A_0A_1'A_2'A_4' + A_0'A_2A_3A_4'$



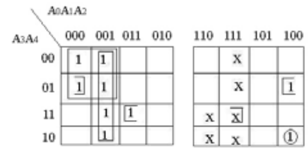
d=$A_0A_1'A_4' + A_0'A_1A_3' + A_0'A_2'A_3' + A_0'A_1'A_2A_3 + A_0'A_2A_3A_4'$



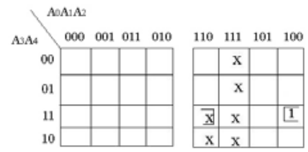
e= $A_1A_2 + A_0'A_1A_3' + A_0'A_2A_3A_4 + A_1'A_2'A_3' + A_0A_2A_4'$



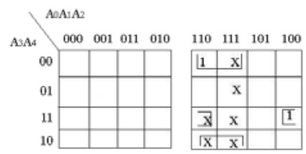
f=$A_1'A_2' + A_0A_1A_4 + A_1A_2A_3 + A_0'A_1'A_3' + A_0'A_1'A_4'$



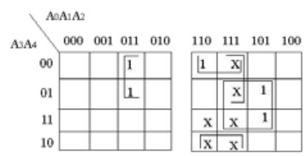
g= $A_0'A_1'A_3' + A_0'A_1'A_2 + A_1'A_2'A_3'A_4 + A_1A_2A_3A_4 + A_0A_1'A_2'A_3A_4'$



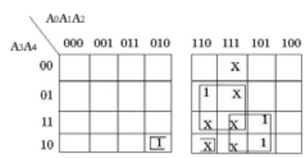
h= $A_0A_2'A_3A_4$



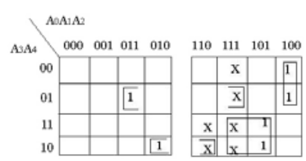
i=$A_0A_1A_4' + A_0A_2'A_3A_4$



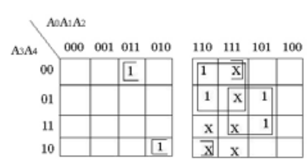
j= $A_0A_1A_4' + A_1A_2A_3' + A_0A_2A_4$



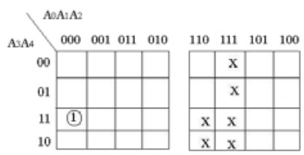
k= $A_0A_1A_4 + A_0A_2A_3 + A_1A_2'A_3A_4'$



l= $A_1A_2'A_3A_4' + A_1A_2A_3'A_4 + A_0A_2A_3 + A_0A_1'A_2'A_3'$



m= $A_0A_2A_4 + A_0A_1A_3' + A_1A_2'A_3A_4' + A_1A_2A_3'A_4'$



n= $A_0'A_1'A_2'A_3A_4$

Figure 1.2. Minimization of switching functions

DESIGN WITH LOGIC GATES

The circuit diagram for each segment is shown in Figure 1.3. In the circuit only AND, OR, and NOT gates are used. "Electronics Workbench" version 5.2 is used to simulate the following combinational circuits.

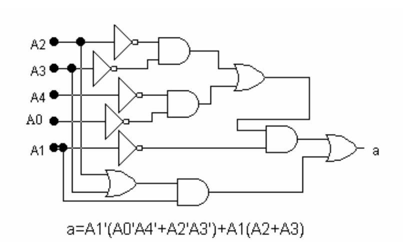


Figure 1.3.1. Logic circuit for segment 'a'

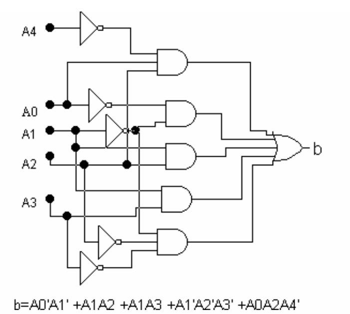


Figure 1.3.2. Logic circuit for segment 'b'

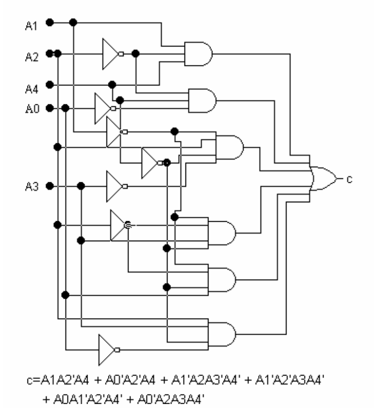


Figure 1.3.3. Logic circuit for segment 'c'

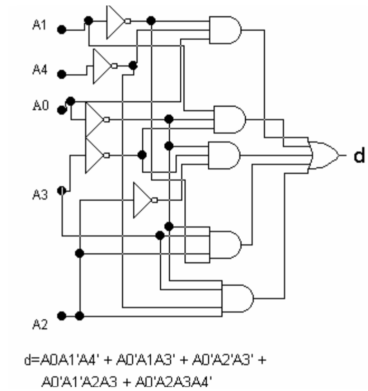


Figure 1.3.4. Logic circuit for segment 'd'

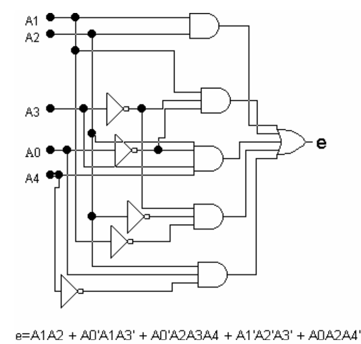


Figure 1.3.5. Logic circuit for segment 'e'

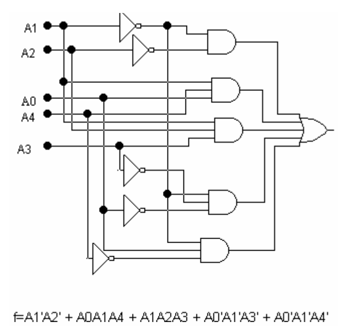


Figure 1.3.6. Logic circuit for segment 'f'

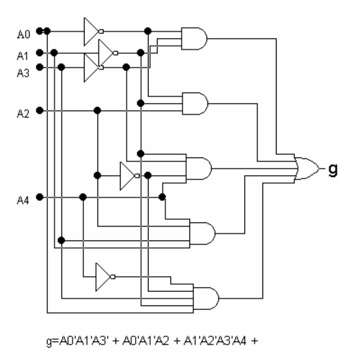


Figure 1.3.7. Logic circuit for segment 'g'

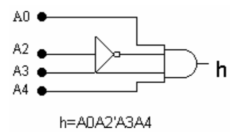


Figure 1.3.8. Logic circuit for segment ‘h’

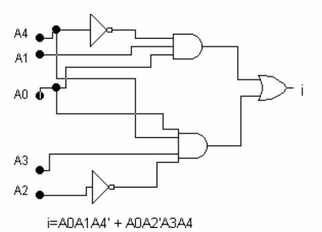


Figure 1.3.9. Logic circuit for segment ‘i’

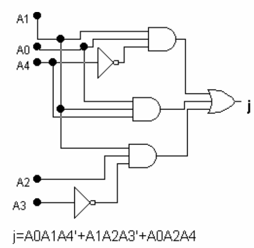


Figure 1.3.10. Logic circuit for segment ‘j’

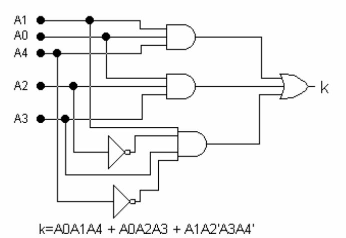


Figure 1.3.11. Logic circuit for segment ‘k’

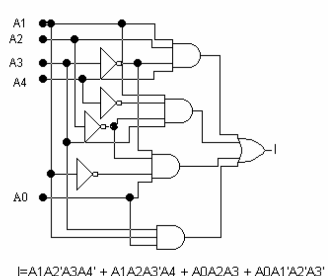


Figure 1.3.12. Logic circuit for segment ‘l’

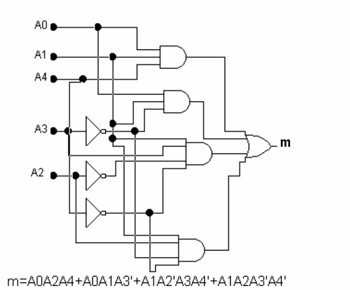


Figure 1.3.13. Logic circuit for segment ‘m’

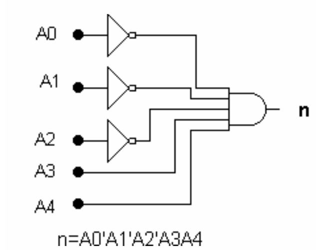
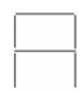


Figure 1.3.14. Logic circuit for segment ‘n’

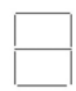
Figure 1.3. Design with AND, OR, NOT gates

EXPERIMENTAL RESULTS

Each representation of English capital letters using the corresponding segments is shown in Figure 1.4. The segments we want to activate for each English capital letters which are non-overlap are also shown with every letters below.



A= (a, b, d, e, f, g)



B= (a, b, c, d, e, f, g)

C= (a, b, c, f)

D= (a, b, c, f, n)

E= (a, b, c, f, g)

F= (a, b, f, g)

G= (a, b, c, d, f, g)

H= (a, b, d, e, g)

I= (d, e)

J= (c, d, e)

K= (a, b, k, l, m)

L= (a, b, c)

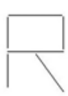
M= (a, b, d, e, j, m)

N= (a, b, d, e, j, l)

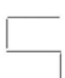
O= (a, b, c, d, e, f)

P= (a, b, e, f, g)

Q= (a, b, c, d, e, f, l)



R= (a, b, e, f, g, l)



S= (a, c, d, f, g)

T= (f, h, i)

U= (a, b, c, d, e)

V= (j, m)

W= (a, b, d, e, k, l)

X= (j, k, l, m)

Y= (i, j, m)

Z=(c, f, k, m)

Figure 1.4. Experimental results of segments

CONCLUSIONS

To the best of our knowledge, it is the first proposed 14-segment display system for English alphabet. Next effort should be giver to improve the quality of visualization for letters 'G' and 'V'.

REFERENCES

[1] ZVI KOHAVI, *switching and finite automata theory,* pp.85, 86, 90, 95,131

[2] M. Morris Mano, *Digital Logic and Computer Design,* pp.151, 100-109

[3] BPB Publications, *Microprocessor Data Hand Book,* pp 5,8,20,183

[4] Ahmed Yousuf Saber, Mamun Al Murshed Chowdhury and Suman Ahmed, *Designing 11-Segment Display for Bangla Digits,* 5th ICCIT 2002, pp 237-240