

Design of N-Segmented Display system for Bengali vowels

Saif Hannan¹, Ahasan Ullah Shohag²

^{1,2}Department of Electronic and Telecommunication Engineering
International Islamic University Chittagong, Bangladesh.

Corresponding e-mail: saifhannan@yahoo.com

Abstract

Different segmented display systems have been proposed for representing Bangla and English alphabets and numerals in past few years. In order to distinguish Bengali Vowels in precise form, they must be represented with their unavoidable curved shapes. Accurate recognition of Vowels is difficult due to their complex shapes. In this paper, we tried to represent a total of 11 Bangla vowels in simpler and accurate look. To improve their accuracy, we have proposed a 58-segmented display. And we derived a very simple, scientific logic to find segments out of matrix display. We have introduced a 4-bit input line to represent all the Bengali Vowels, analyzed all the 58-segments against each and every Vowel, and designed the suitable circuit for each of the 58 segments.

Keywords: Segments, Truth table, Expressions, 4-bit input line, Points.

I. INTRODUCTION

In order to represent Bangla vowels in an n segmented display, it is not possible to avoid both strokes and curves. For ingeniousness, we have to use least number of segments as possible, as well as obviate the co-existence of both curves and strokes. Curved segments will be expensive. So, envisaging accuracy, simplicity and efficiency, following purposes are followed: to represent all the Bengali vowels using least number of segments as possible, to improve their accuracy, to avoid segment overlapping and to alleviate the necessity of representing curved shapes using curved segments. For representing Bengali vowels, many types of segmented displays were developed. In order to obtain minimum segmentation using above mentioned points, we have to allow adequate space for drawing shapes for all vowels, place all the vowels on the same model and point out the similarities and dissimilarities.

Among different technologies to build the IC, we prefer CMOS technology for this work to implement, because CMOS technology has low power consumption and scalable high noise immunity [1] and CMOS circuits are much cheaper in comparison to other technologies [2].

II. DESIGN METHOD

As matrix display represents nice outlook, we have tried to find the segments over matrix display. At the beginning, we

drew the shapes for all Bengali vowels on a 30×30 matrix, keeping both similarities and dissimilarities among vowels; i.e., we have chosen the appropriate and minimum number of segments to represent different curves. Our 30×30 matrix whose cells are indexed from 1 to 900 and those are known as points. We drew the curves with distinct shapes without causing partition [3].

A. Segments

By the term 'Segment', we mean here a set or group of points that can be switched at a time [3]. To represent the shape of a particular vowel on the matrix, we need to split the shape of the vowel into several pieces and find out the points on the matrix for each piece. Then for all the vowels, we find all the pieces and superimpose them for similarities.

The common and similar pieces are separated from dissimilar pieces and then we have taken as minimum number of pieces as possible to identify them as segments.

B. Points to represent segments

Using *Find_Segment algorithm* we found that, a total of 58 segments were required to cover all the points that were possessed by all vowels drawn over the 30×30 matrix. The segments were named from S_0 to S_{57} . The points that were needed to represent the segments are summarized in Table I.

Table I: Points needed to represent Segments

Segments	Points Needed
S_0	212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 235, 236, 237, 238, 239
S_1	234
S_2	282, 311, 315, 340, 346, 377, 469, 529, 588, 646, 652, 681, 731, 732, 733
S_3	353, 383, 413, 443, 473, 503, 533, 563, 593, 623, 653, 683, 713, 743, 773, 803, 833
S_4	371
S_5	283
S_6	284
S_7	408
S_8	439

S ₉	499
S ₁₀	559
S ₁₁	617
S ₁₂	704
S ₁₃	730
S ₁₄	590
S ₁₅	621
S ₁₆	266, 326, 356, 386, 446, 476, 506, 536, 566, 596, 626, 656, 686, 716, 746, 776, 806, 836
S ₁₇	296
S ₁₈	416
S ₁₉	41, 70, 100, 131, 132, 133, 134, 135, 136, 137, 168, 198
S ₂₀	368, 337, 308, 279, 280, 281, 312, 374, 405, 435, 494, 523, 552, 581, 608, 609
S ₂₁	343
S ₂₂	465
S ₂₃	610
S ₂₄	641
S ₂₅	672
S ₂₆	703
S ₂₇	734, 765, 796, 827, 828
S ₂₈	436
S ₂₉	407
S ₃₀	378, 350, 381, 410, 468, 497, 526, 556
S ₃₁	349
S ₃₂	351
S ₃₃	586
S ₃₄	678
S ₃₅	740
S ₃₆	771
S ₃₇	253, 373, 403, 434, 501, 531, 561, 619, 677, 735, 764, 763, 762
S ₃₈	313
S ₃₉	470
S ₄₀	706
S ₄₁	761
S ₄₂	483, 513, 544, 575, 606, 637, 668, 699
S ₄₃	572, 603, 634, 665, 696, 727, 758, 759, 760
S ₄₄	263, 293, 332
S ₄₅	624, 654, 685, 715
S ₄₆	322
S ₄₇	380
S ₄₈	254, 285, 288, 317, 409, 438, 467, 496, 525, 555,
S ₄₉	679
S ₅₀	772
S ₅₁	316
S ₅₂	259, 260, 291
S ₅₃	376, 345, 258, 709, 708, 707, 705, 456, 486, 517, 548, 579
S ₅₄	444, 445, 327, 357, 387, 265, 108, 139, 170, 171, 172, 201

S ₅₅	462, 461, 430, 400, 342, 255, 256, 318, 441, 471, 528, 527, 651, 681, 739, 768, 797, 826, 825, 824, 823, 792
S ₅₆	287
S ₅₇	442

C. Segments to Represent each Alphabet

We required a total of 58-Segments to take all the points that were possessed by any vowels over a 30x30 matrix. The Segments that needed to represent the Vowels are summarized in Table II.

Table II: Bengali Vowel versus 58-Segments

Bengali vowels	Segment needed to be activated
A	S ₀ , S ₁ , S ₂ , S ₃ , S ₄ , S ₅ , S ₆ , S ₇ , S ₈ , S ₉ , S ₁₀ , S ₁₁ , S ₁₂ , S ₁₃ , S ₁₄ , S ₁₅ , S ₄₂ , S ₄₄ = 18
Av	S ₀ , S ₁ , S ₂ , S ₃ , S ₄ , S ₅ , S ₆ , S ₇ , S ₈ , S ₉ , S ₁₀ , S ₁₁ , S ₁₂ , S ₁₃ , S ₁₄ , S ₁₅ , S ₁₆ , S ₁₇ , S ₁₈ , S ₄₂ , S ₄₄ = 21
B	S ₀ , S ₁ , S ₁₉ , S ₂₀ , S ₂₁ , S ₂₂ , S ₂₃ , S ₂₄ , S ₂₅ , S ₂₆ , S ₂₇ = 11
C	S ₀ , S ₁ , S ₈ , S ₁₁ , S ₁₉ , S ₂₀ , S ₂₁ , S ₂₂ , S ₂₃ , S ₂₈ , S ₂₉ , S ₃₀ , S ₃₁ , S ₃₂ , S ₃₃ , S ₃₄ , S ₃₅ , S ₃₆ , S ₄₉ = 19
D	S ₀ , S ₁ , S ₅ , S ₇ , S ₈ , S ₁₃ , S ₁₄ , S ₁₉ , S ₂₁ , S ₂₂ , S ₂₈ , S ₂₉ , S ₃₄ , S ₃₇ , S ₃₈ , S ₃₉ , S ₄₀ , S ₄₁ , S ₄₂ = 19
E	S ₀ , S ₁ , S ₅ , S ₇ , S ₈ , S ₁₃ , S ₁₄ , S ₁₉ , S ₂₁ , S ₂₂ , S ₂₈ , S ₂₉ , S ₃₄ , S ₃₇ , S ₃₈ , S ₃₉ , S ₄₀ , S ₄₁ , S ₄₂ , S ₄₃ = 20
F	S ₃ , S ₁₁ , S ₁₆ , S ₁₇ , S ₁₈ , S ₃₂ , S ₃₃ , S ₃₄ , S ₃₅ , S ₄₄ , S ₄₅ , S ₄₆ , S ₄₇ , S ₄₈ , S ₄₉ , S ₅₀ , S ₅₁ , S ₅₂ = 18
G	S ₃ , S ₁₂ , S ₂₃ , S ₂₄ , S ₂₅ , S ₂₆ , S ₃₅ , S ₃₆ , S ₄₀ , S ₄₆ , S ₅₀ , S ₅₁ , S ₅₂ , S ₅₃ , S ₅₆ = 15
H	S ₁ , S ₃ , S ₁₂ , S ₁₇ , S ₁₈ , S ₂₃ , S ₂₄ , S ₂₅ , S ₂₆ , S ₃₅ , S ₃₆ , S ₄₀ , S ₄₆ , S ₅₀ , S ₅₁ , S ₅₂ , S ₅₃ , S ₅₄ , S ₅₆ = 19
I	S ₄ , S ₆ , S ₉ , S ₁₀ , S ₁₃ , S ₁₄ , S ₁₅ , S ₃₁ , S ₃₅ , S ₃₈ , S ₃₉ , S ₄₁ , S ₄₂ , S ₄₇ , S ₅₅ , S ₅₆ = 16
J	S ₁ , S ₄ , S ₆ , S ₉ , S ₁₀ , S ₁₃ , S ₁₄ , S ₁₅ , S ₁₇ , S ₁₈ , S ₃₁ , S ₃₅ , S ₃₈ , S ₄₁ , S ₄₇ , S ₃₉ , S ₄₁ , S ₄₂ , S ₅₄ , S ₅₅ , S ₅₆ , S ₅₇ = 21

III. ARCHITRCTURE PROPOSAL

We got the desired architecture by using Find_Segments algorithm, which is imprinted in Figure.1. For design simplicity, we have used bar type segments most. Matrix points that make up a curved shape, are considered as detached points connected together to form a segment. A pictorial view of all characters is shown in figure 2.

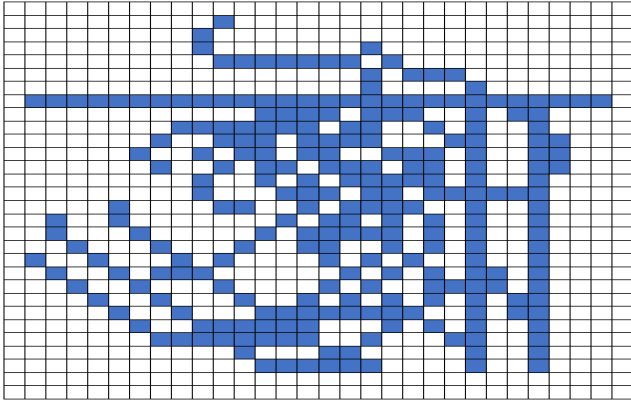
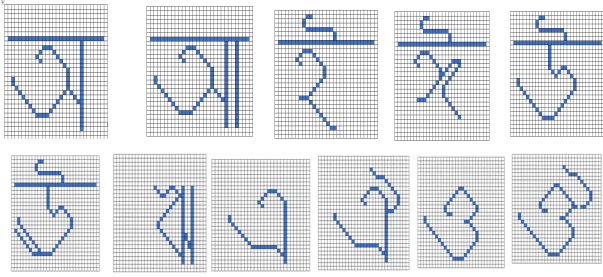
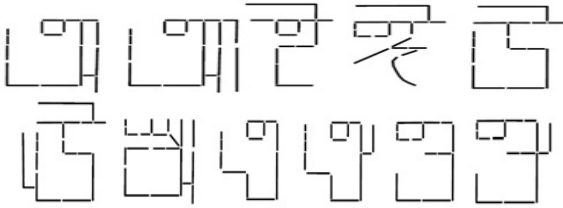


Figure 1: Proposed Model of 58-Segmented display for Bengali Vowels.



(a)



(b)

Figure 2: Overview of all vowels (a) proposed 58-segmented system and (b) 49-segment display [4]

IV. PERFORMANCE

From Fig.2 (b), the difference between the proposed and previous architecture [4] becomes apparent. For all vowels, the newly proposed design is better. Moreover, they [4] are incapable to depict the real curved shapes of Bangla vowels. So, our proposed design wins the extent in all aspects, at the cost of a few more segments.

V. CIRCUIT DESIGN

A. Truth table

The truth table was derived as shown in Table III. Simplified expression for each segment was found from this truth table by K-map. We have derived appropriate logic circuits to display each and every Bengali Vowels. We also have assumed that 'A' has a 4-bits code 0000; 'Av' has a 4-bits code 0001, and so on [5].

Table III: Truth table

A	0	0	0	0	0	0	0	1	1	1	
B	0	0	0	0	1	1	1	1	0	0	0
C	0	0	1	1	0	0	1	1	0	0	1
D	0	1	0	1	0	1	0	1	0	1	0
S ₀	1	1	1	1	1	1	0	0	0	0	0
S ₁	1	1	1	1	1	1	0	0	1	0	1
S ₂	1	1	0	0	0	0	0	0	0	0	0
S ₃	1	1	0	0	0	0	1	1	1	0	0
S ₄	1	1	0	0	0	0	0	0	0	1	1
S ₅	1	1	0	0	1	1	0	0	0	0	0
S ₆	1	1	0	0	0	0	0	0	0	1	1
S ₇	1	1	0	0	1	1	0	0	0	0	0
S ₈	1	1	0	1	1	1	0	0	0	0	0
S ₉	1	1	0	0	0	0	0	0	0	1	1
S ₁₀	1	1	0	0	0	0	0	0	0	1	1
S ₁₁	1	1	0	1	0	0	1	0	0	0	0
S ₁₂	1	1	0	0	0	0	0	1	1	0	0
S ₁₃	1	1	0	0	1	1	0	0	0	1	1
S ₁₄	1	1	0	0	1	1	0	0	0	1	1
S ₁₅	1	1	0	0	0	0	0	0	0	1	1
S ₁₆	0	1	0	0	0	0	1	0	0	0	0
S ₁₇	0	1	0	0	0	0	1	0	1	0	1
S ₁₈	0	1	0	0	0	0	1	0	1	0	1
S ₁₉	0	0	1	1	1	1	0	0	0	0	0
S ₂₀	0	0	1	1	0	0	0	0	0	0	0
S ₂₁	0	0	1	1	1	1	0	0	0	0	0
S ₂₂	0	0	1	1	1	1	0	0	0	0	0
S ₂₃	0	0	1	1	0	0	0	1	1	0	0
S ₂₄	0	0	1	0	0	0	0	1	1	0	0
S ₂₅	0	0	1	0	0	0	0	1	1	0	0
S ₂₆	0	0	1	0	0	0	0	1	1	0	0
S ₂₇	0	0	1	0	0	0	0	0	0	0	0
S ₂₈	0	0	0	1	1	1	0	0	0	0	0
S ₂₉	0	0	0	1	1	1	0	0	0	0	0
S ₃₀	0	0	0	1	0	0	0	0	0	0	0
S ₃₁	0	0	0	1	0	0	0	0	0	1	1
S ₃₂	0	0	0	1	0	0	1	0	0	0	0
S ₃₃	0	0	0	1	0	0	1	0	0	0	0
S ₃₄	0	0	0	1	1	1	1	0	0	0	0
S ₃₅	0	0	0	1	0	0	1	1	1	1	1
S ₃₆	0	0	0	1	0	0	0	1	1	0	0
S ₃₇	0	0	0	0	1	1	0	0	0	0	0
S ₃₈	0	0	0	0	1	1	0	0	0	1	1
S ₃₉	0	0	0	0	1	1	0	0	0	1	1
S ₄₀	0	0	0	0	1	1	0	1	1	0	0
S ₄₁	0	0	0	0	1	1	0	0	0	1	1
S ₄₂	1	1	0	0	1	1	0	0	0	1	1
S ₄₃	0	0	0	0	0	1	0	0	0	0	0
S ₄₄	1	1	0	0	0	0	1	0	0	0	0
S ₄₅	0	0	0	0	0	0	1	0	0	0	0
S ₄₆	0	0	0	0	0	0	1	1	1	0	0
S ₄₇	0	0	0	0	0	0	1	0	0	1	1
S ₄₈	0	0	0	0	0	0	1	0	0	0	0
S ₄₉	0	0	0	1	0	0	1	0	0	0	0
S ₅₀	0	0	0	0	0	0	1	1	1	0	0
S ₅₁	0	0	0	0	0	0	1	1	1	0	0
S ₅₂	0	0	0	0	0	0	1	1	1	0	0
S ₅₃	0	0	0	0	0	0	0	1	1	0	0

S ₅₄	0	0	0	0	0	0	0	0	0	1	0	1
S ₅₅	0	0	0	0	0	0	0	0	0	0	1	1
S ₅₆	0	0	0	0	0	0	0	1	1	1	1	1
S ₅₇	0	0	0	0	0	0	0	0	0	0	0	1

B. Simplified Expression

We first constructed the combination vector for each Vowel. It shows which segments are needed to represent each Vowel. Then from the truth table the expression for each segment is derived.

$$\begin{aligned}
S_0 &= A'B' + A'BC' & S_1 &= A'B' + ABC' + AB'D' & S_2 &= A'B'C' \\
S_3 &= A'B'C' + A'BC + AB'C'D' \\
S_4 &= A'B'C' + AB'C'D + AB'CD' & S_5 &= A'C' \\
S_6 &= A'B'C' + AB'C'D + AB'CD' & S_7 &= A'C' \\
S_8 &= A'C' + A'B'CD & S_9 &= A'B'C' + AB'C'D + AB'CD' \\
S_{10} &= A'B'C' + AB'C'D + AB'CD' \\
S_{11} &= A'B'C' + A'B'CD + A'BCD' \\
S_{12} &= A'B'C' + A'BCD + AB'C'D' \\
S_{13} &= A'C' + AB'C'D + AB'CD' & S_{14} &= AC' + AB'C'D + AB'CD' \\
S_{15} &= A'B'C' + AB'C'D + AB'CD' & S_{16} &= A'B'C'D + A'BCD' \\
S_{17} &= A'B'C'D + A'BCD' + AB'D' \\
S_{18} &= A'B'C'D + A'BCD' + AB'D' & S_{19} &= A'B'C + A'BC' \\
S_{20} &= A'B'C & S_{21} &= A'B'C + A'BC' & S_{22} &= A'B'C + A'BC' \\
S_{23} &= A'B'C + A'BCD + AB'C'D' \\
S_{24} &= A'B'CD' + A'BCD + AB'C'D' \\
S_{25} &= A'B'CD' + A'BCD + AB'C'D' \\
S_{26} &= A'B'CD' + A'BCD + AB'C'D' \\
S_{27} &= A'B'CD' & S_{28} &= A'B'CD + A'BC' & S_{29} &= A'B'CD + A'BC' \\
S_{30} &= A'B'CD & S_{31} &= A'B'CD + AB'C'D + AB'CD' \\
S_{32} &= A'B'CD + A'BCD' & S_{33} &= A'B'CD + A'BCD' \\
S_{34} &= A'B'CD + A'BC' + A'BCD' \\
S_{35} &= A'B'CD + A'BC + AB'C' + AB'CD' \\
S_{36} &= A'CD + AB'C'D' & S_{37} &= A'BC' \\
S_{38} &= A'BC' + AB'C'D + AB'CD' \\
S_{39} &= A'BC' + AB'C'D + AB'CD' \\
S_{40} &= A'BC' + A'BCD + AB'C'D' \\
S_{41} &= A'BC' + AB'C'D + AB'CD' \\
S_{42} &= A'C' + AB'C'D + AB'CD' & S_{43} &= A'BC'D \\
S_{44} &= A'B'C' + A'BCD' & S_{45} &= A'BCD' \\
S_{46} &= A'BC + AB'C'D' & S_{47} &= A'BCD' + AB'C'D + AB'CD' \\
S_{48} &= A'BCD' & S_{49} &= A'B'CD + A'BCD' \\
S_{50} &= A'BC + AB'C'D' & S_{51} &= A'BC + AB'C'D' \\
S_{52} &= A'BC + AB'C'D' & S_{53} &= A'BCD + AB'C'D' \\
S_{54} &= AB'D' & S_{55} &= AB'C'D + AB'CD' \\
S_{56} &= A'BCD + AB'C' + AB'CD' & S_{57} &= AB'CD'
\end{aligned}$$

C. Implementation

There are 11 Alphabets in Bangla vowels. For this we needed 11 output signals. To get 11 signals, it is required to have 4 signals as input lines. By using these lines, we can specify our desired output signal as binary value. The logic circuit for the 1st segment of display is shown below. All the 58 segments would be designed accordingly. Here, only AND, OR and NOT gates are used. In this design process we have taken the assistance from **DSCH** and **Micro wind** software. The cost was not estimated for implementing the system, as the work was just a proposal and hence no experiment was done.

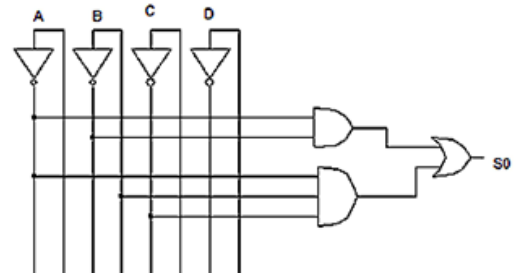


Figure 3: Sample Circuit Implementation.

VI. DISCUSSION AND CONCLUSION

The distinction between the proposed and previous architecture [4, 5, 6, and 7] becomes clear. For all vowels, the newly proposed design is better. Since our designed circuit is little bit complex, we can reduce the complexity of the circuit through k-map analysis. If, we can reduce the number of required segment, it will respond faster. In future work, we can make it capable to implement the joint Bangla Alphabets, try to increase the number of characters in same display and can reduce the complexity.

Our designed architecture helps to find segments for characters and symbols used in Bangla language by matrix display. The output is quite nice, though the characters are in segmented form. It may be interfaced with any Computer or may be used on **calculators, ATM, Ticket Counter, and Toll Plaza** where users are friendly with Bangla language.

REFERENCES

- [1] S.H.Teen, L.L.Lim and J.H. Lim; "IC Layout Design of Decoder Using Electric VLSI Design System", International Journal of Electronics and Electrical Engineering Vol. 3, No. 1, February, 2015.
- [2] R.Behzad, "Design of analog CMOS integrated circuits", McGraw-Hill Higher Education, New York, 2001
- [3] Tanzin Rahman, Tanvir Khan, Sarder Saadat Ahmed, Chandan Kumar Karmakar 'N-Segmented Display of Bangla Numerals' in proceedings of 8th international conference on computer and information Technology (ICIT), Dhaka, Bangladesh, 2005.
- [4] Salahuddin Mohammad Masum, Sarwar Morshedul Haque, Swapon Chandra Dash, Kazi Faisal Kabir 'Segmented Display System for Bengali Alphabets'. Faculty of Science & Information Technology, Daffodil International University, Dhaka 1207, Bangladesh.
- [5] Islam, A.K.M.N., Mahmud, S.M.M., Shahrier, N., & Sattar, M.A. (2003); "Designing 17-segment Display for Bangla Vowels", 6th ICCIT 2003, Jahangirnagar University, PP. 283-286, December 19-21.
- [6] MASUM, S.M. & AL-MAMUN, M.A. (2005); "Designing 14-Segment Display for Bengali Vowels", Asian Journal of Information Technology, Grace Publications Network, Vol. 4 No. 2, PP. 178-184.
- [7] Mohammad Osiur Rahman, Mohammad Aktaruzzaman Khan, "Display Unit for Bangla Characters" ISSN 1813-7733 Vol. - 4, December 2007 Published in April 2008 (p 71-86).