



Data Compression Using Content Addressable Memories

Ashwin Santhosh^(✉) and Harish Kittur Malikarjun^(✉)

Department of Micro and Nano Electronics,
School of Electronics Engineering, VIT Vellore, Vellore, Tamil Nadu, India
ashwin.santhosh2016@vitstudent.ac.in,
kittur@vit.ac.in

Abstract. This paper presents a 16 bit CAM of 16 words for data compression. The input data is searched for match among the 16 stored data. The match produces compressed data output. The average compression time is also calculated for each search data and is compared with the existing CRAM compression design.

Keywords: Content-Addressable memory · NAND cell · XNOR CAM

1 Introduction

The access time for an item stored in memory is reduced if the memory is accessed by its content rather than by its address. Such type of memories are called Content Addressable Memories. The search data is compared with an array of stored data and its match location address is obtained within single clock cycle. High search speed makes content addressable memories a better choice for lookup operations. Speed of operation of the system is improved because of the parallel search operation. It is also possible to cascade CAM cells [1] so that the size of the look up table can be increased. They have a wide range of applications like ATM switches, network routers, data compression, IP filter, memory mapping etc. Even though it has these many applications it is best suitable for searching operations [5].

2 CAM Basics

The Fig. 1 shows an array of content addressable memory which contains four words each with four bits arranged in horizontal manner. Corresponding to each word there exist a matchline and search data is also given to each cell. Then all matchlines are precharged high, making them all temporarily in the match condition. Each cell compares its stored data against the data on its SL. Matchlines where all bits match discharge to ground and rest of the matchlines where there is a mismatch discharge to ground. Then the address of the match location is produced by the encoder [1].

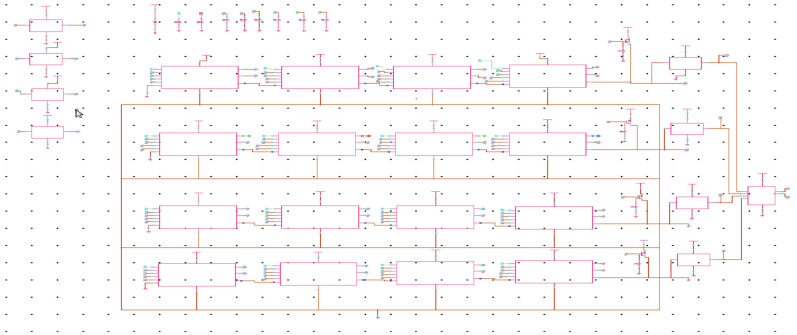


Fig. 1. CAM with 4 words having 4 bits each.

3 XNOR NAND Type Cell

The NAND cell does the comparison between the stored data and the search data using 3 comparison transistors shown in Fig. 2. The operation of the NAND cell is explained below with example. Consider $SL = 1$ and stored value = 1, the first transistor say MX is ON passing logic 1 to node D. Hence the middle transistor say MY is turned ON. When $SL = 0$ and stored value = 0 also MY is turned ON. Logic 1 is passed by transistor MZ in this case. Thus for both the match cases the node D is at logic 1 turning ON MY. The case when SL is not equal to stored value leads to the miss condition turning OFF MY. Thus the node D is XNOR function of SL and stored value [1]. When n cells are connected in series the match line ML1 to MLn are connected forming a word [1–4]. A match condition occurs for the entire word only if every cells in a word in the chain remain matched (Fig. 3).

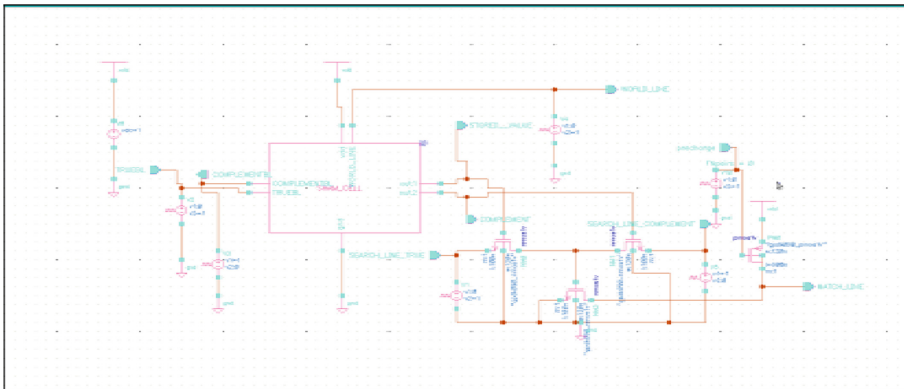


Fig. 2. NAND type CAM cell

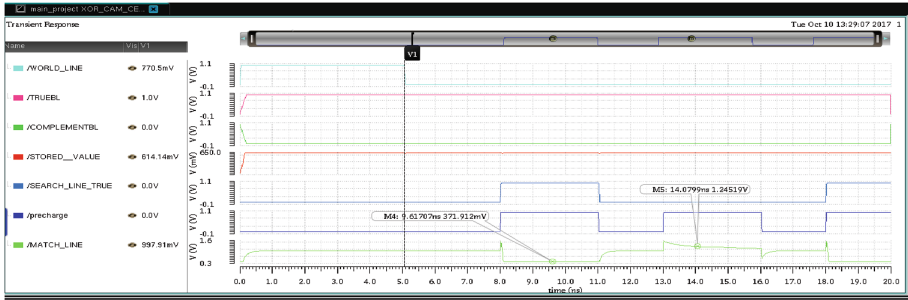


Fig. 3. Output waveforms showing the functionality of a single XNOR NAND type cell

4 Data Compression

Data compression eliminates the unwanted data that is present in an information, producing a shorter but same information. Content Addressable Memories [1] are the best choice for Data Compression because the movement of packets through LAN or WAN needs address translation. Majority of the compression algorithm time is taken for maintaining and searching these data structures. The algorithm throughput can be increased if they are replaced by some hardware search engine. In a Data Compression application, Content Addressable Memory lookup is performed after each word of the actual data is given. A CAM will generate a result in a one clock cycle regardless of the size of the table or length of search list. This makes Content Addressable Memory the best choice for Data Compression applications that make use of complex tables in their algorithm. Data compression helps to reduce the processing time of the system and also helps in resource optimization [5]. In the proposed design, the data bits are stored in the CAM. Then the search data is compared with the bits stored in each of the rows of CAM cells. For the match case, the lowest match location address (token) replaces the search data and hence the number of bits in the final output is reduced [2]. In the case of missing match, the next search data is searched in the array [2]. The existing CRAM design [3] consist of RAM, encoder/decoder in addition to Content Addressable Memories. To a single silicon array the compression and decompression engines (CAM and RAM) are added [3, 4]. They are fast but simultaneous compression and decompression cannot be achieved on separate data.

5 The 16 * 16 CAM Array

It consist of cascaded connection of 16 CAM cells in 16 rows to form the 16 match lines. At each of the matchline output the precharge transistor [4] puts the initial voltage of the match lines to VDD. For match case all the nmos transistors M1 to M16 in the corresponding row are on thus creating a path to ground. For mismatch case one of the series transistors M1 to M16 is off putting the corresponding match line in high state. All the matchline outputs are inverted and are given as input to the 16 * 4 encoder. At the encoder output, the compressed 4 bits are obtained (Fig. 4 and Table 1).

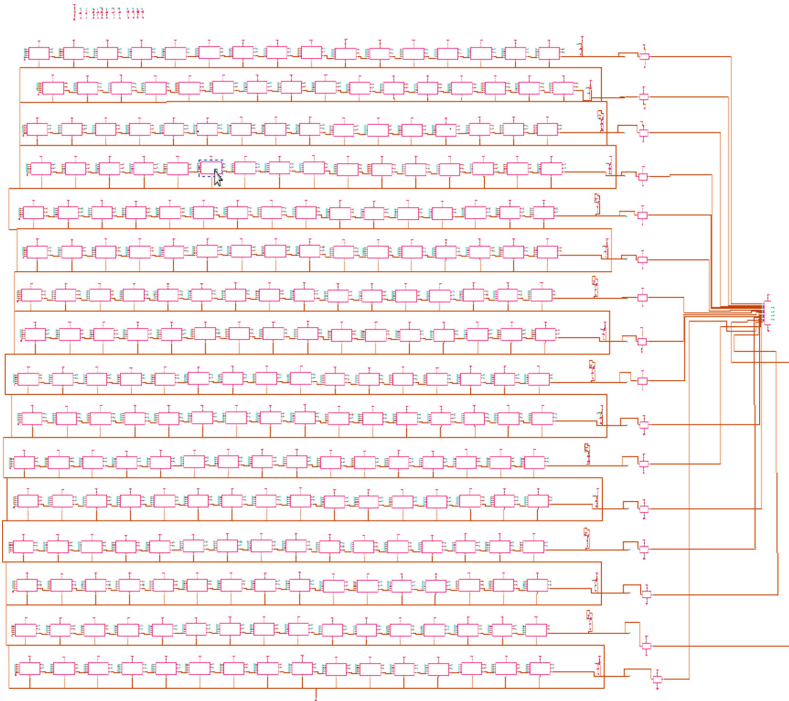


Fig. 4. The 16 * 16 CAM array

Table 1. The 16 bit stored words in each of the 16 rows.

Match line	Token	Stored data
1	0000	00000000000000000000000000000000
2	0001	00010001000100010001000100010001
3	0010	00100010001000100010001000100010
4	0011	00110011001100110011001100110011
5	0100	01000100010001000100010001000100
6	0101	01010101010101010101010101010101
7	0110	01100110011001100110011001100110
8	0111	10001000100010001000100010001000
9	1000	10011001100110011001100110011001
10	1001	10101010101010101010101010101010
11	1010	10111011101110111011101110111011
12	1011	11001100110011001100110011001100
13	1100	11011101110111011101110111011101
14	1101	11101110111011101110111011101110
15	1110	11111111111111111111111111111111
16	1111	01110111011101110111011101110111

6 Simulation Results

The 16 * 16 CAM array is designed in 90 nm CMOS technology and the following simulation results are obtained in cadence virtuoso for four different search data (Figs. 5, 6, 7, 8 and Tables 2, 3).

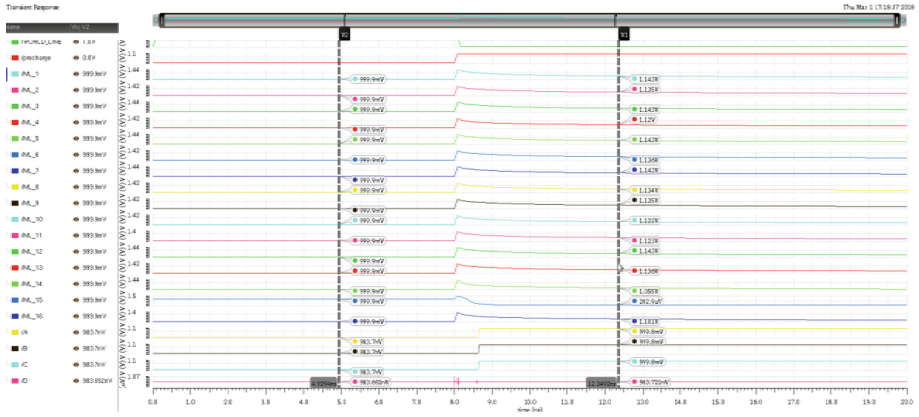


Fig. 5. Data compression for search data 11111111111111111111111111111111

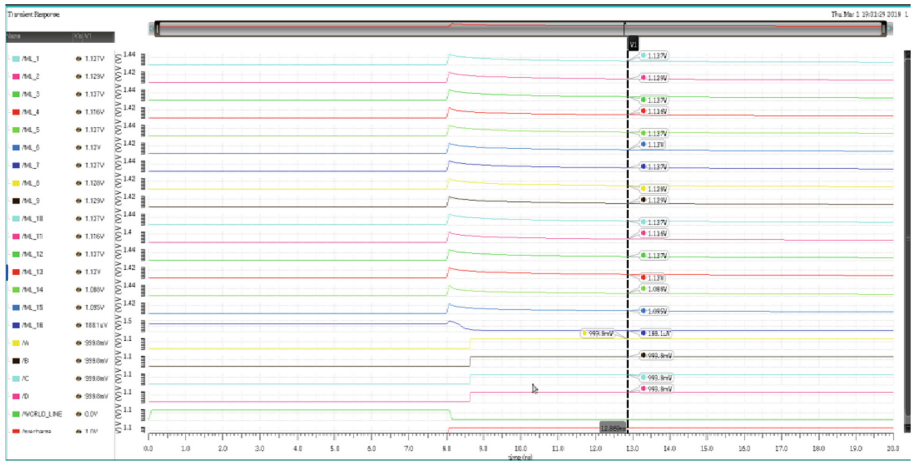


Fig. 6. Data compression for search data 01110111011101110111011101110111

Table 3. Comparison of compression time.

Design name	Compression time
CRAM compression design	100 MB/s
Proposed design	16 bits/0.5 ns

7 Conclusion

The 16 bit CAM of 16 words is designed for data compression. Data compression is successfully demonstrated. The compression time for the precharged match line is 0.5 ns. All the four compressions are obtained under 1 ns. Faster compression time is achieved as compared to CRAM design for data compression.

References

1. Pagiamtzis, K., Sheikholesami, A.: Content addressable memory (CAM) circuits and architectures: a tutorial and survey. *IEEE J. Solid State Circ.* **41**, 712–727 (2006)
2. Content Addressable Memory (CAM) Applications for ispXPLD devices
3. Craft, D.J.: A fast hardware data compression algorithm and some algorithmic extensions
4. McAuley, A.J., Francis, P.: Fast routing table lookup using CAMs. In: *Proceedings of IEEE Infocom*, vol. 3, pp. 1282–1391 (1993)
5. Peng, M., Azgomi, S.: Content-Addressable memory (CAM) and its network applications. Altera International Ltd