CAM – ENCS3330 Project

Abdallah Alabed - 1190781, Osama Rihami - 1190560

Birzeit University, Ramallah, Palestine

Abstract—Large-capacity content-addressable memory (CAM) is beneficial in a variety of applications that require high-speed lookup table. It is used extensively in low power CPU design, network routers, and cache controllers. Content addressable memory system includes CAM cells that contains a compare circuit and a memory bit-cell that stores complementary bits. The compare circuit consists of complementary inputs to receive the complementary stored bits, and an input node to receive a single-ended reference bit. The main CAM design challenge is to reduce the power consumption associated with large amount of parallel switching circuitry, without sacrificing speed or density.

This paper presents a novel CAM circuit level implementation aiming at reducing the comparator power and the crowbar current. Consequently, the average current consumption during CAM operation is reduced. In addition, the proposed circuit topology eliminates the need to route the complementary data which saves routing resources. Simulation results using 22nm process technology shows that the elimination of the crowbar current during writing operation saves 40% of power for single bit-cell CAM, while sharing the compare circuitry among 8 bit cells CAM saves 14% of the power without any performance impact on area and delay.

Keywords—Low Power, CAM memory, caches, content addressable memory, memory architecture, SOC design, processors design, CAM cell, chip, design, XOR CAM.

I. INTRODUCTION

In Random Access Memory (RAM) system, a memory address and a read control signal are send to the RAM module to retrieve or read the contents stored at the target memory address. Content Addressable Memories (CAMs) is defined from the functionality point of view as a memory with large amount of stored data that performs comparison between the input searched data and the stored data. CAM cell can perform all functions of the SRAM cell, including read and write operations given address and data. In addition, it is capable of performing the matching/comparing operation. As a kind of comparison between CAM and RAM, one can say that the CAM is an inverse of RAM in terms of functionality. At the reading operation, RAM retrieves the stored data for a given address while CAM returns the address corresponding to a given data word as shown in Fig. 1 [1]. In addition, the search is carried out serially when searching for a data in a RAM block. Consequently, many cycles are needed to find a particular data word. But in CAM-based memory, all addresses are searched in parallel fashion to retrieve the corresponding address that stores the desired word.

CAMs are gaining increased importance due to their parallel pattern matching property. This property makes them useful in variety of applications such as cache controllers where a quick search is needed. CAMs can be defined as a hardware search based engines that are much faster than algorithmic approaches for search-intensive applications [2] like find matching contents in a database or table. For example, Fig. 2 [3] shows that the CAM system has been employed as a routing table where the match address output from the CAM is in fact a pointer used to retrieve associated data from the RAM. In this case the associated data is the router output port. Also, the CAM system may be viewed as a hardware embodiment of a software-based associative array. So, based on these domain of applications, the key/basic idea of the CAM is to compare the CAM data lines of the cell with contents of a certain bits, and then if the data is matched, the matched lines of the stored bits are raised and then the addresses at matched target data are returned. In addition to that, the CAM system may return the data word or other associated information [2].

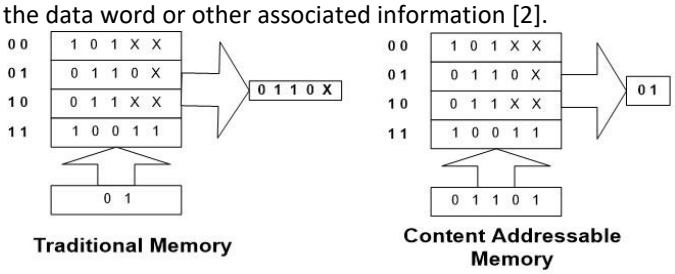


Fig. 1: RAM versus CAM in read operation [1]

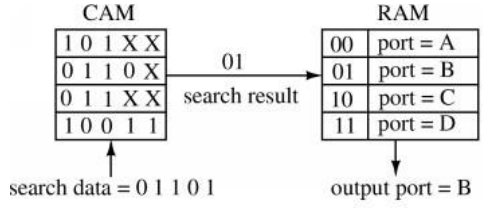


Fig. 2: Address lookup engine with CAM system [3]

The major use of CAM memory falls in the applications that require high search or lookup speed. However, the CAM search speediness comes at the cost of higher power consumption with large amount of parallel switching circuitry and larger silicon area [4], which are the two main design parameters especially for Mobile devices [5]–[6]. Thus, the main goal of the recent dedicated research in CAM Memory area concentrates on reducing power consumption without slowing down the CAMs search speed. The accomplished work has been done at two levels: circuit level and architecture level. In circuit level, the CAM structure is implemented as NAND-gate or NOR-gate types while the architectural level uses simple SRAM cell as a bit storage and the comparison function is equivalent to XOR i.e. XNOR logic operation [4]. Indeed, using XOR as pass gate is widely used in CAM-based memory arrays which are susceptible to currents contention due to the variation on arrival time and delays in the input signals. The design of the elementary CAM chip cell can be abstracted as a cross product of SRAM and XNOR circuits as presented in [7].Nonetheless, there have been many attempts to reduce the transistor count and resulting area for the CAM XOR block in order to decrease the power consumption. A comprehensive review of different varieties of CAM cells, were presented in [8]. Other design approaches were presented in [4], [8]–[9].

Bit cells of a CAM system include compare-circuits to compare contents of the bit cells with reference bit value provided to the compare circuits. Conventional CAM compare circuits are implemented with complementary or differential reference bit lines, which disadvantageously increase routing complexity and space requirements. Typically, the compare circuits include separate pass circuits. And thus the switching delays in the CAM cell can cause unwanted current contention between the separate pass circuits, which manifests itself as a crowbar current that wastes power and slows down CAM speed. The goal of this paper is to reduce both he active and the contention power in addition to reducing the required metal tracks to route signals. The proposed design is based on real product using CMOS technology. It reduces write energy compared to traditional real approaches. In addition, the proposed design saves on routing resources and interconnect power without any performance impact on chip area and delay. The rest of the paper is organized as follows: Section

II describes the functionality and the implementation of the CAM system. Section III illustrates the available techniques in reducing the power consumption for the CAM system. Section IV introduces our approach (for typical and custom CAM cell). In section V, the simulation results are studied and analyzed. Section VI concludes our proposed work.

II. CAM FUNCTIONALITY AND IMPLEMENTATION

The overall CAM functionality is to take a search word and return the matching memory location, but one can think in different perspective of this operation as a fully programmable arbitrary mapping of a large space of the input search word to a smaller space of the output match location. As another perspective, the operation of a CAM can be seen as tag portion of a fully associative cache where the input of tag portion of a cache is compared to all addresses stored in the tag memory. Then, when matching is occurred, a single match line goes high, indicating the location of a match. Although many circuits are common to both CAMs and caches, in our work we concentrate on CAMs rather than on fully associative caches since they target smaller capacity and higher speed CAMs.

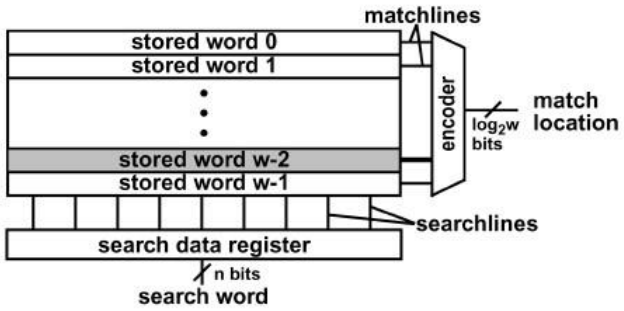


Fig. 3: Content-addressable memory (CAM) conceptual view searches over *W* words [4].

Content-addressable memory consists of *W* words as shown in Fig 3 [4]. The system input is a search word that will be broadcasted onto the search lines to be compared with a table of stored data. The CAM word size, in terms of bits, is often large, with an available and used implementations it ranging from 36 to 144 bits while the table size of the typical CAM is ranging between a few hundred to 32K entries, which correspond to an address space ranging from 7 bits to 15 bits. The match lines indicate whether a matching occurred between a stored word and the given search word. According to the match lines value (0 or 1), the location or address of word is determined by feeding the match lines to an encoder, assuming that the encoder is used in the systems where only a single match is expected. In addition to that there is often a hit signal, not shown in the Fig. 3, which is responsible to indicate the case when no matching location in the CAM is found.

The largest commercial implementations of single-chip CAMs are 18 Mbits size [10]; however, the largest CAMs stated in the literature are 9 Mbits size. Usually, as a rule of thumb, about half of the largest available SRAM chip size is the largest available CAM chip size where such rule of thumb refers to the fact states that the typical CAM cell contains two SRAM cells [11]. The capacity of CAM chips that have been published from 1985 to 2004 revealed an exponential growth rate of semiconductor memory circuits as reported in Fig. 4
[4].

From the CAM architecture side, an example of a small model is shown in Fig. 5 [4]. The CAM model in this example consists of 4 words where each word contains 4 bits arranged horizontally, which corresponding to 4 CAM cells.

Also, for each word there is a match-line which is 4-input AND gate, in addition to a differential search line-pair which corresponding to the search word bits. The CAM search process operates by loading first the search data word into the search-data registers. Then, all match lines are pre-charged, which put all of them in the match state. Afterwards, the search-word is broadcasted onto the differential search lines using the search-line drivers, and then a

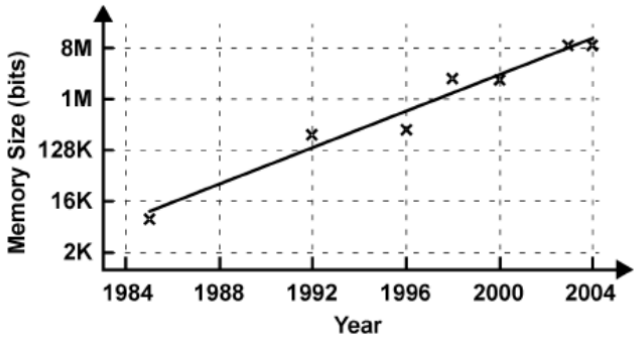


Fig. 4: CAM capacity (log scale) over year of publication [4].

comparison is performed between each stored bit of the CAM core cell and the corresponding search-lines bit. In the case of all bits are matched, the match-lines remain in the pre-charged high state and the match-lines that have miss bits are to ground. According to the match-lines state, the And gate will determine whether all bit matched or not. At last, the matched location address is extracted by using the encoder.

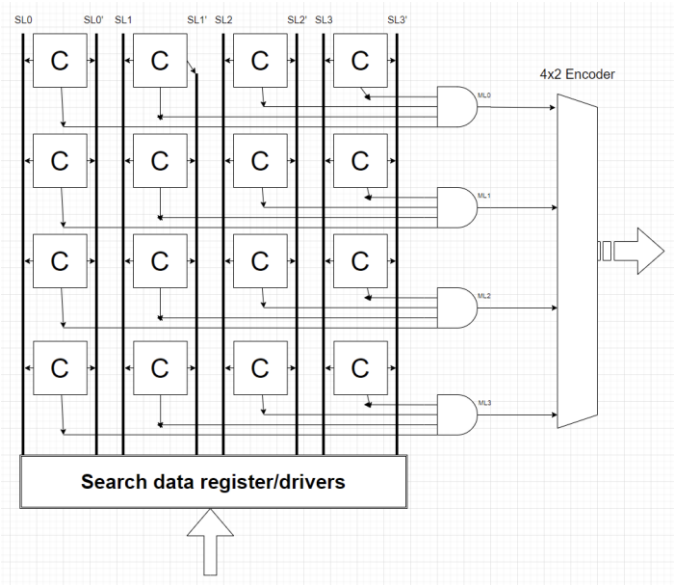


Fig. 5: Simple schematic CAM model for 4 words with 4 bits each, including individual core cells, differential search lines, and 'And' Gate [4].

The bit comparison operation is logically equivalent to applying an XOR gate between the search bit and the stored bit; however, from the implementation-wise, the NOR and NAND cells are often used. When using the NOR cells, the comparison is done between the complement of the stored bit D, and the complement of the search-data of the search line SL, by using four transistors with minimum size as possible to keep high cell density. The four transistors aim at implementing a pulldown of XNOR gate with SL and D inputs where each pair of them creates a pulldown path from the match-line ML by connecting ML to ground. In the case of matching between SL and D, both pulldown paths are disabled, resulted in disconnecting ML from ground. For NAND cell implementation, it is different from the NOR implementation through using three comparison transistors rather than four when comparing between the stored-bit D, and corresponding search-data on the corresponding search-lines SL. This cell implementation becomes clear in particular when connecting multiple NAND cells in serial fashion. In such a case, joining the and nodes will form a word, taking into account that the pulldown path of a CMOS NAND gate is created by a serial nMOS chain of all transistors. However, satisfying the match condition for the entire word is dependent on being every cell in a word in a match state condition.

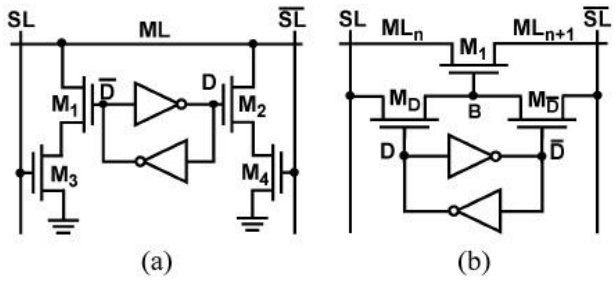


Fig. 6: CAM core cells for (a) 10-T NOR-type CAM and (b) 9-T NAND-type CAM [4].

The CAM core cells of the 10-T NOR-type and 9-T NAND type are shown in Fig. 6 [4] where SRAM is used as a data storage cell. For simplicity sake,the figure didn't show the SRAM access transistors and the associated bit lines, but it is clear that six of cell-transistors are accounted for both the SRAM storage and access transistors. The NOR cell has an important property concerning on providing a full rail voltage at all comparison transistors gates. On contrary, the NAND cell provides a reduced logic 1 voltage at node B.

Conventional CAM bit-cell CMOS implementation which is currently used by Intel is shown in Fig.7. The bit-cell (BIT) is written into by enabling the write word-line (WRWL) and driving the desired value through write bit-lines (WRBL/WRBLY). Then, the BIT value is compared against CAM-DATA and the MATCH is asserted when BIT and CAM-DATA

values are same. However, this particular implementation has a crowbar current issue since there is one gate delay between BIT and BITX which opens one pass gate before closing the second as shown in Fig.7. Therefore, the new propose circuit will reduce this crowbar current. For more information about the Intel CAM design you can see the patents in [12], [13].

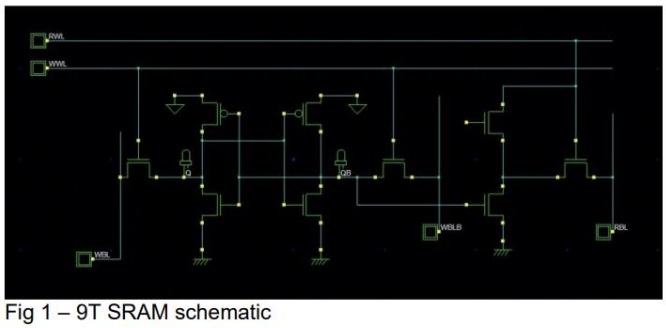


Fig 1 – 9T SRAM schematic

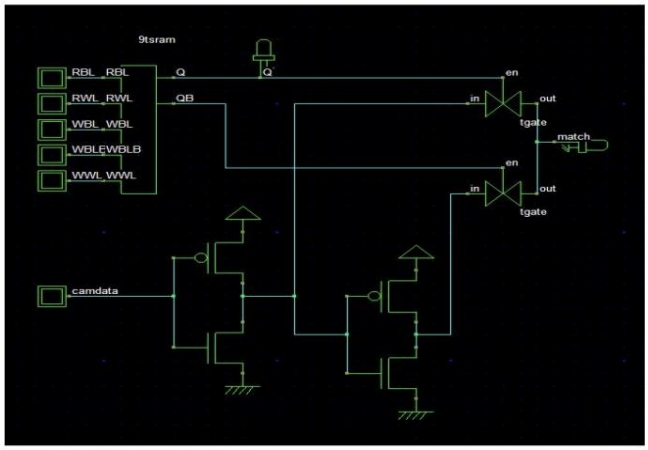
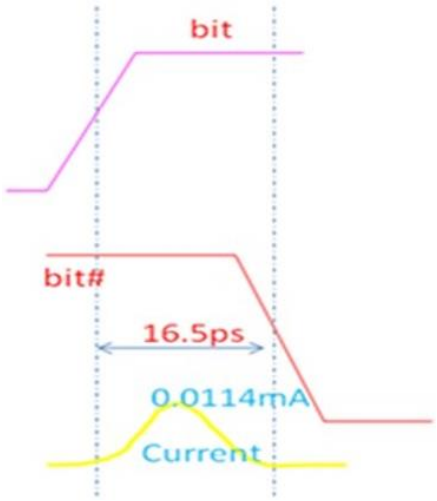


Fig 4 – CAM cell schematic

Fig.8 shows an example of CAM system in an exploded view



using 8-connected conventional Intel CAM cells. Each CAM cell includes a memory/storage bit-cell (SRAM) and a compare circuit (CMPR). Bit-cell stores a bit (BIT) and a complementary bit (BITX) at corresponding nodes of the bitcell. Bit cell may be static RAM (SRAM) or dynamic RAM (DRAM) [13], but the SRAM has been the preferred choice so far to build CAM-cells because of its robustness. Compare circuit CMPR includes complementary inputs (also referred to as differential inputs) to receive complementary stored bits

WBLB and WBLB, and a CAM data to two inverters as buffer. The two pass gates works as an XOR gate to chech if the cam data matched the bit stored in the 9T SRAM. Q' is used as enable to camData' pass gate, whereas Q is used as enable to camData input signal.

Current research on CAM system proposed Memristor technology [14], [15] to be used for CAM-based memory. The main advantages of this technology are: (1) it is non-volatility (2) and can be scale better than CMOS technology. However, the Memristor technology is still not well developed and add a cost to CMOS since a new material (metal oxide) is needed. In addition, there are many challenges with memristor technology such as big variation, state drift, read disturb of the state during match.

III. CAM POWER REDUCTION TECHNIQUES

Reducing the power dissipation by the CAM cells has been accomplished at two different levels where the first level is at the circuit design of the CAM cell, and the second level is at the architectural method used in connecting the CAM cells together. Most of the realized works have been focusing at the architecture level due to the difficulty of improving the design of the CAM cell at the circuit or transistor level. Although of that difficulty, the author of [7] describes a technique aims at reducing the power consumption and the area at the circuit level of the CAM. In addition , paper [16] discussed how to minimize power using serialization-widening with frequent value encoding for on chip signals like CAM signals. At the architecture level, most of the works have concentrated on pipelining (selective pre-charge) and pre-computation search approaches [4].

Pipelining or selective pre-charge, basically divide matchline into two segments and when performing a search on a data-bits, if the results of matching applied on the first segment fails there is no need to check the remaining bits. This will significantly reduce the power consumption. For example, when using uniform random data it only have to search (1/ 2) n of the rows. For n=3 this will save about 88 % of the match line power [17]. The approach can be generalized by dividing the match-line into many number of segments or stages and thus forming a pipeline [17]–[19]. If match in any stage is failed the next stages are shut-off resulting in power saving. Fig. 9 shows an example of pipelined match-line with four segments or stages compared to Non-pipelined match line [4]. In [18], the authors use a pipelined scheme by breaking the matching lines into several segments where the first segment consists of

8 bits and the rest segments of 34 bits each with 144 bits as a total word size. The work lies in evaluating the segments in the sequential fashion where the matching process in a segment begins based on the value of the matching line bit of the previous segment.

Another works [17], [20], [21] have exploited the pipeline scheme called pre-computation by storing additional information about the stored words likes storing the number of ones in the data word as shown in Fig. 10 [4] . Another work stores the parity bit as an additional information for each stored words by adding additional CAM cell to the original word CAMs cells [7]. For an input search data word, the computed parity bit is compared with the stored parity bit of each stored words. When a parity bit of the input word matches a parity bit of a stored word, the output match-line of the CAM cell is used as an enable bit to start the comparing process between the

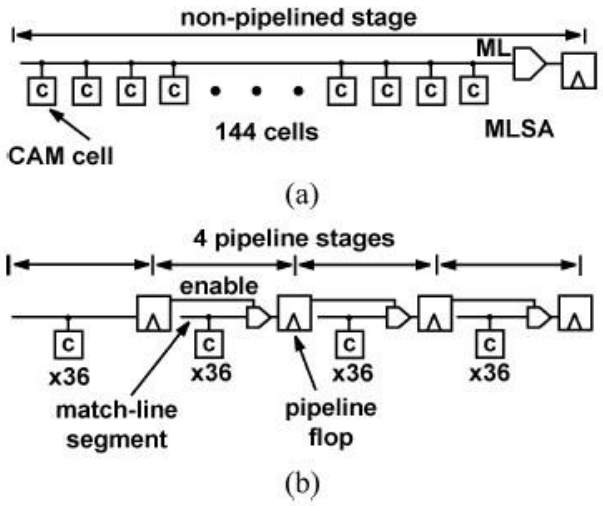


Fig. 9: Match-line schemes (a) Non-pipelined match-line (b) 4-Stage piplined match-line [4].

input word and the stored words. Also, in addition to parity bit pipelining, the matching operation between input word and stored words has been pipelined, but with segments of one bit only. Besides, In [23] a combination of the two power reduction techniques is used.

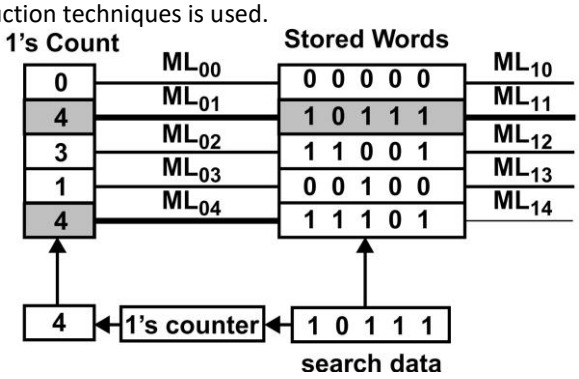


Fig. 10: Conceptual view of pre-computation-based CAM system [4].

It is evident that the proposed solutions at the architecture level improves the power consumption; however, they have several drawbacks which are summarized as following:

1) The latency is increased due to the subsequent matching and also in computing additional information as the case of parity bit.
2) Parity extractor circuit and its CAM cell may consume significant area.

With these drawbacks of the accomplished improvement at the architecture level, the circuit or transistor design still plays an important role since any improvement at this level directly will affect the architecture level. Thus, our work focuses on improving the power consumption of the CAM cell with minimum impact on latency and area.

IV. PROPOSED CAM-CELL CIRCUIT DESIGN

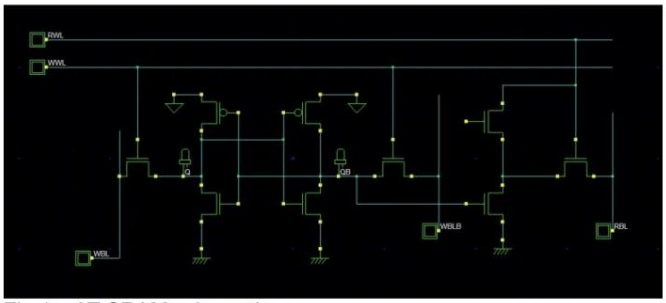


Fig 1 – 9T SRAM schematic

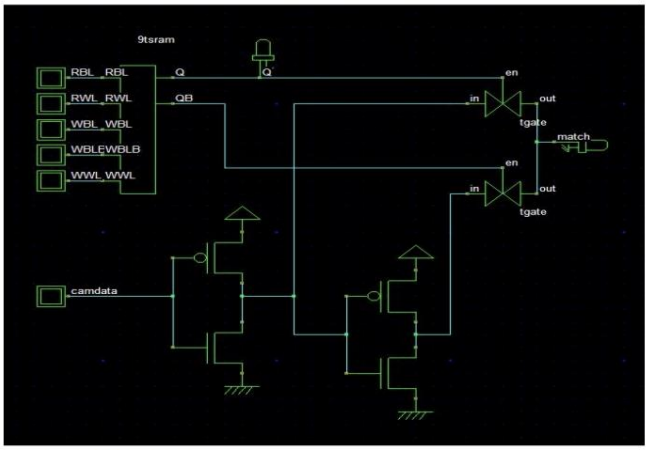


Fig 4 – CAM cell schematic

Fig. 11: Circuit diagram of 9T SRAM, proposed bit cell and compare circuit.

Fig.11 shows the circuit diagram of the proposed implementation of the CAM bit-cell along with its compare-circuit. The proposed compare-circuit includes circuitry which

is controlled by the logic values of single-ended inverted reference bit camData, camData' and the complementary stored bits WBL, WBLB, which forms the inputs of the desired compare-circuit. The output of the compare-circuit is a match-bit which indicates whether the inverted reference bit camData' matches the stored bit WBL. The compare-circuit output a match-bit with logic value 1 when the inverted reference bit camData' same as the stored bit WBL (matches the stored bit WBL) and logic value 0 when the inverted reference bit camData' matches the stored bit WBL (same as the stored complementary bit WBLB).

WBL = Q	WBLB = Q'	camData	camData'	Output/Match
1	0	1	0	1
1	0	0	1	0
0	1	1	0	0
0	1	0	1	1

V. SIMULATION AND RESULTS

Spice Design and Simulation

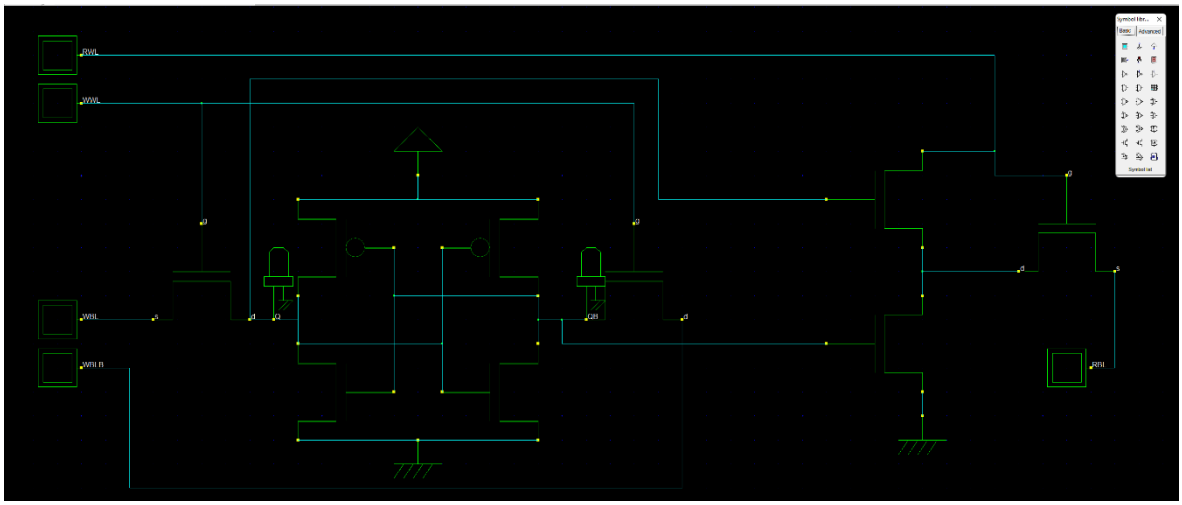


Figure V-1 9t Sram

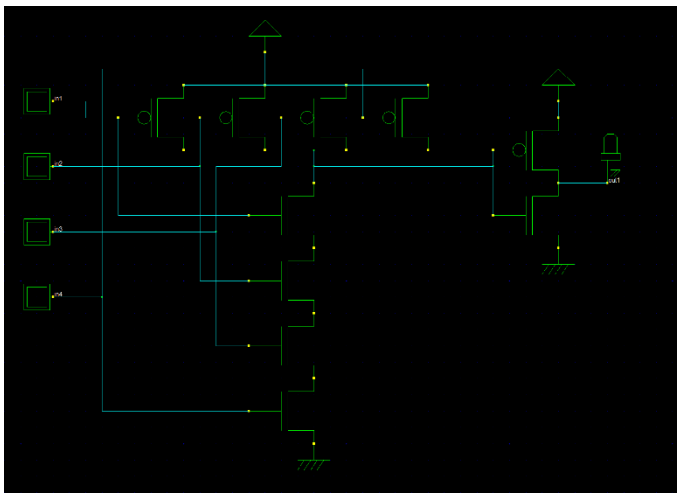


Figure V-3 And Gate

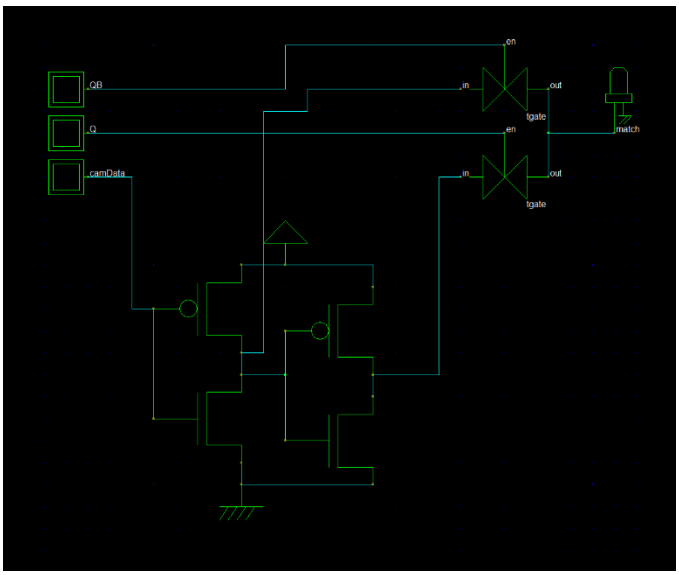


Figure V-2 Match check

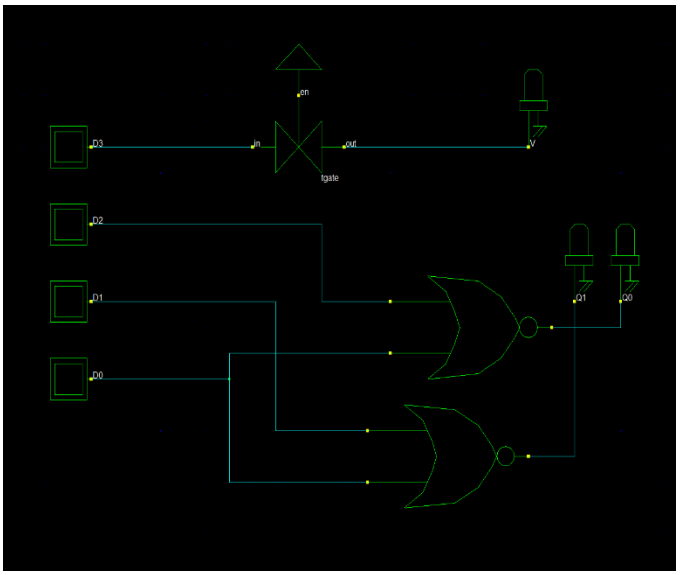


Figure V-4 4x2 Encoder

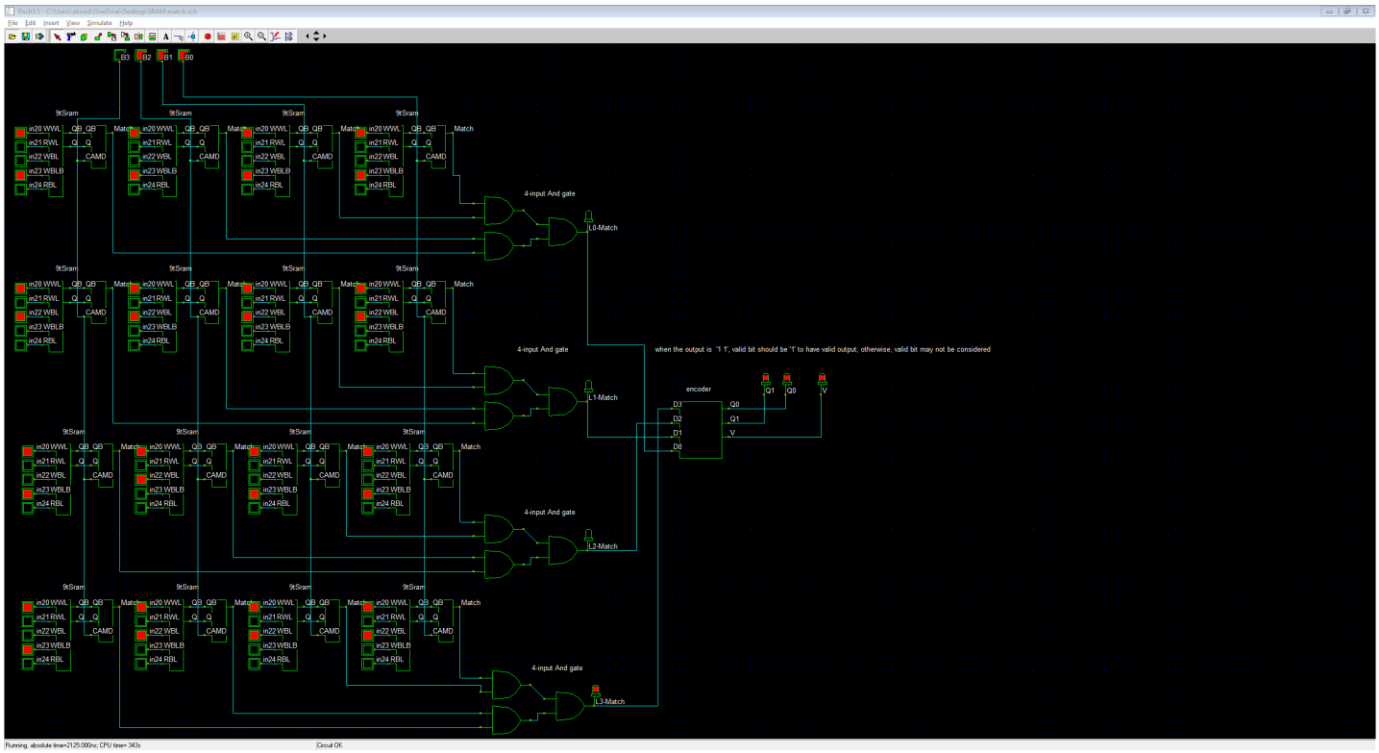


Figure 1 CAM Simulation - 1

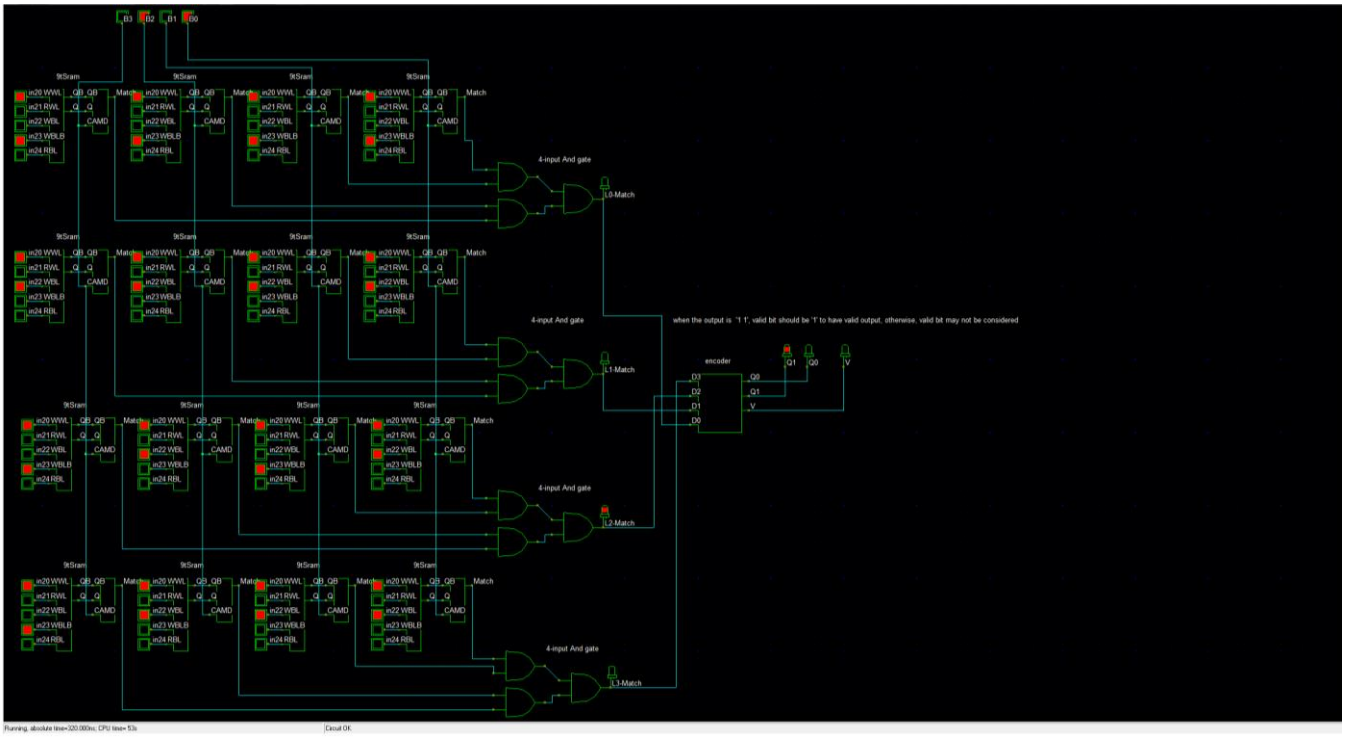


Figure 2 CAM Simulation - 2

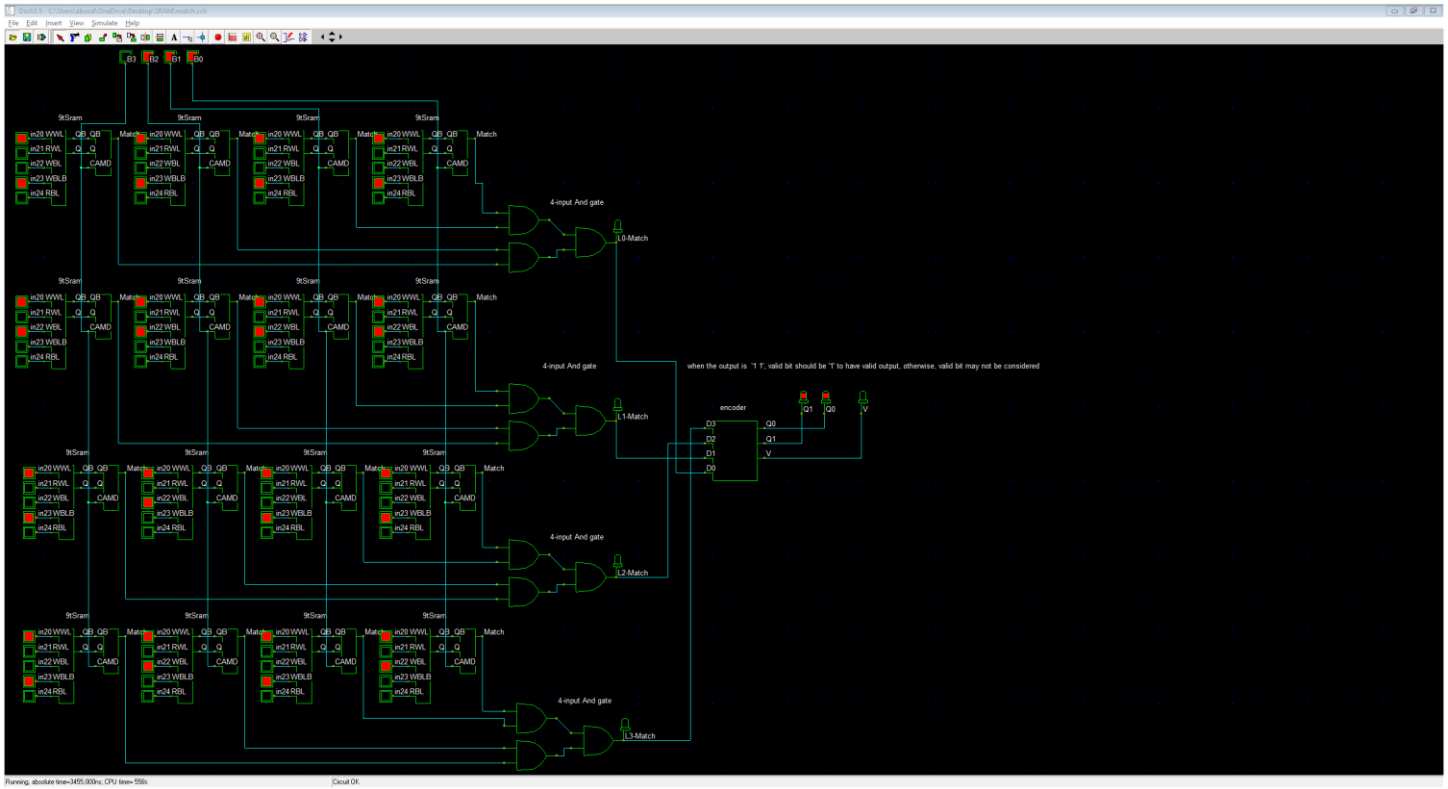


Figure 3 CAM Simulation - 3

In figure 1, the content in the memory as follows:

Line #	*Value*				*Address*
Line 0	0	0	0	0	00
Line 1	1	1	1	1	01
Line 2	0	1	0	1	10
Line 3	0	1	1	1	11

And the cam data is '0101' so the output should be '10'.

In figure 2, the content in the memory as follows:

Line #	*Value*				*Address*
Line 0	0	0	0	0	00
Line 1	1	1	1	1	01
Line 2	0	1	0	0	10
Line 3	0	1	1	1	11

And the cam data is '0111' so the output should be '11'.

Only in this particular case ('11' output), we should look at the encoder valid bit, if it is 1 then our output is correct, otherwise the output is not correct. As in figure 3, there is no match whereas the output is '11'.

Layout and Simulation

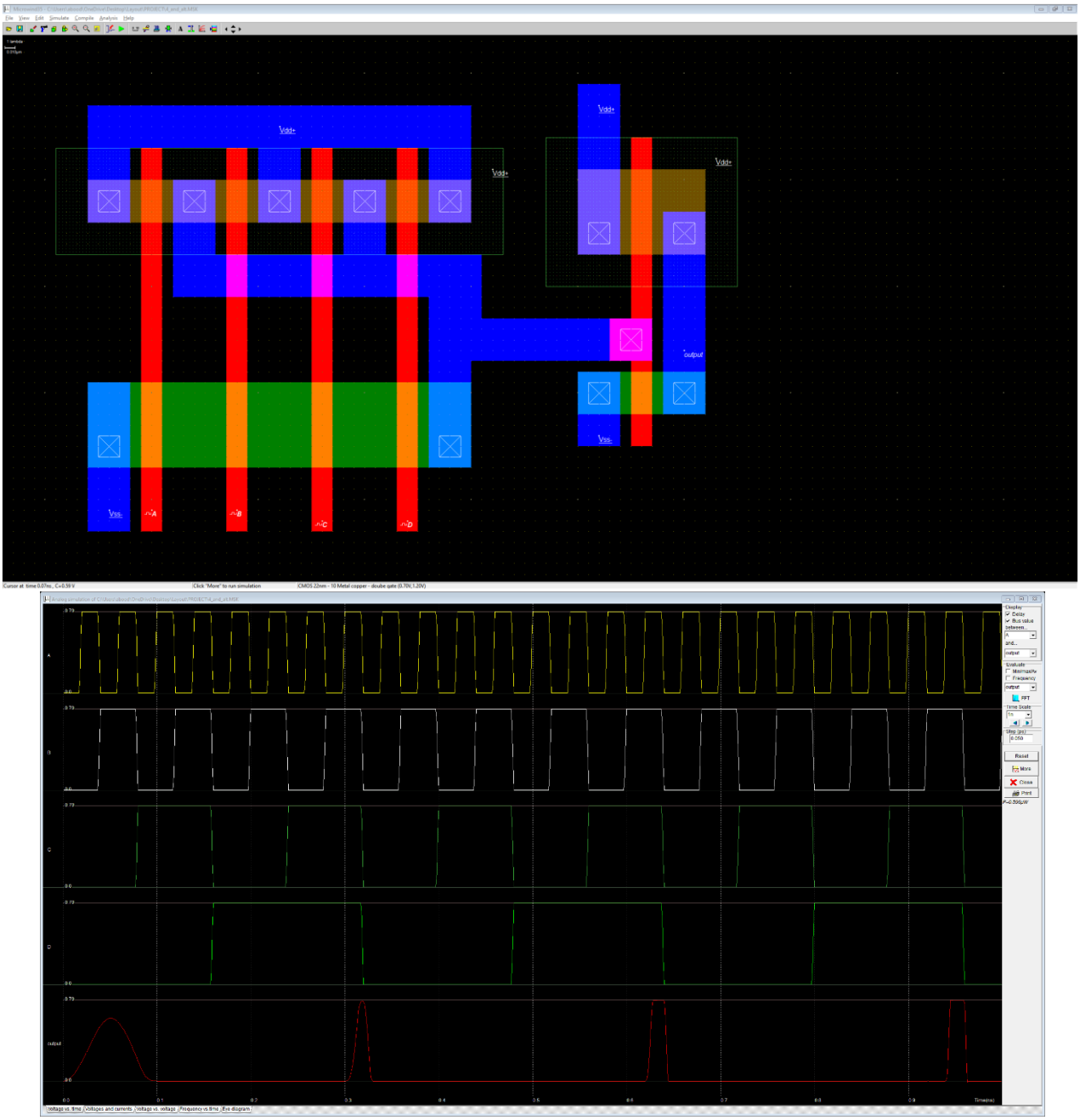


Figure V-5 And Gate Layout And Simulation

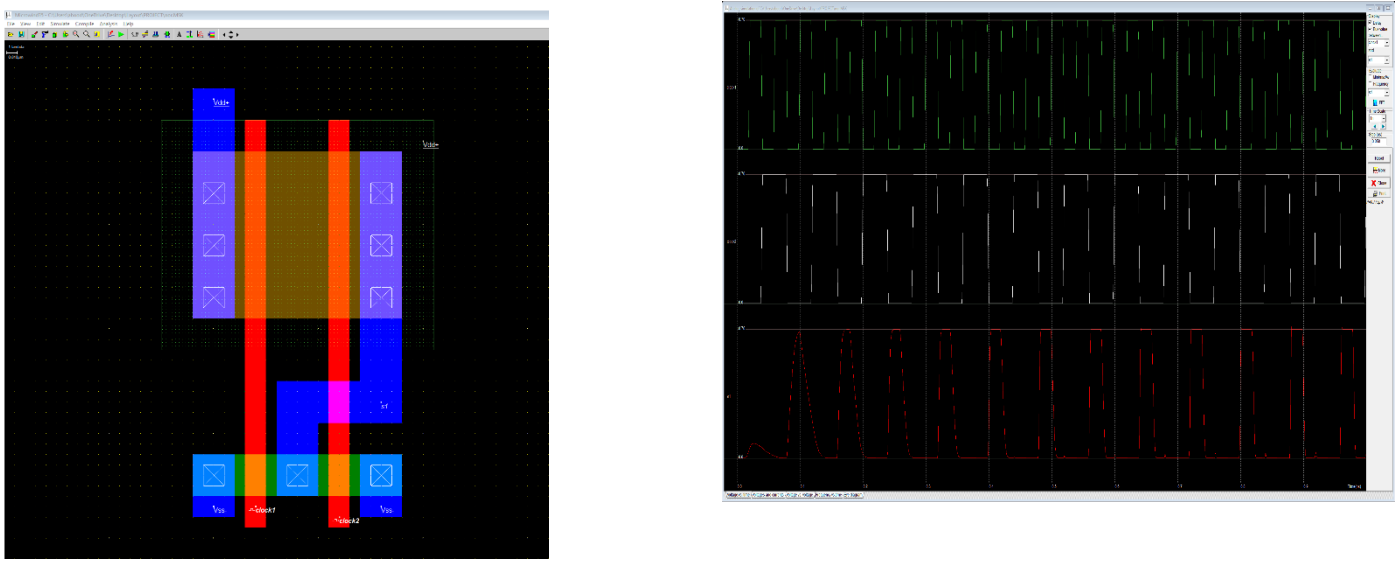


Figure V-6 nor Gate Layout And Simulation

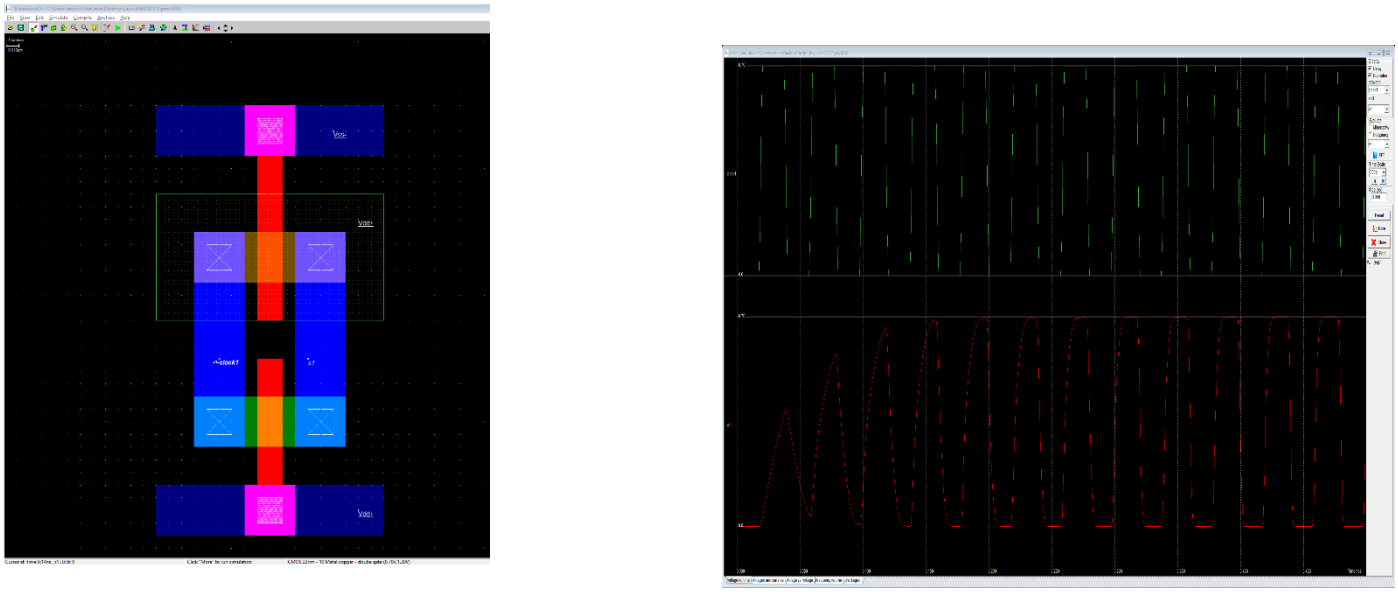


Figure V-7 nor Gate Layout And Simulation

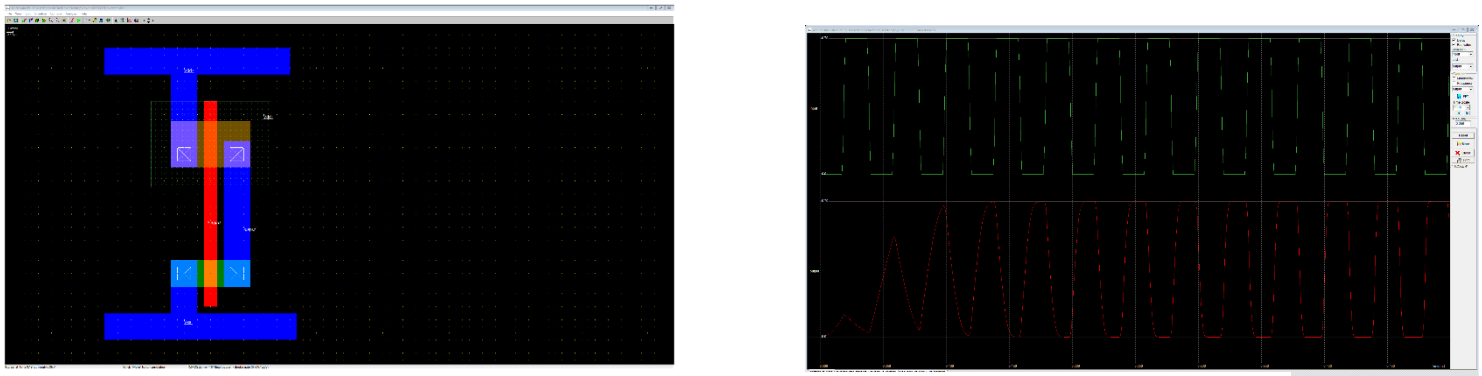
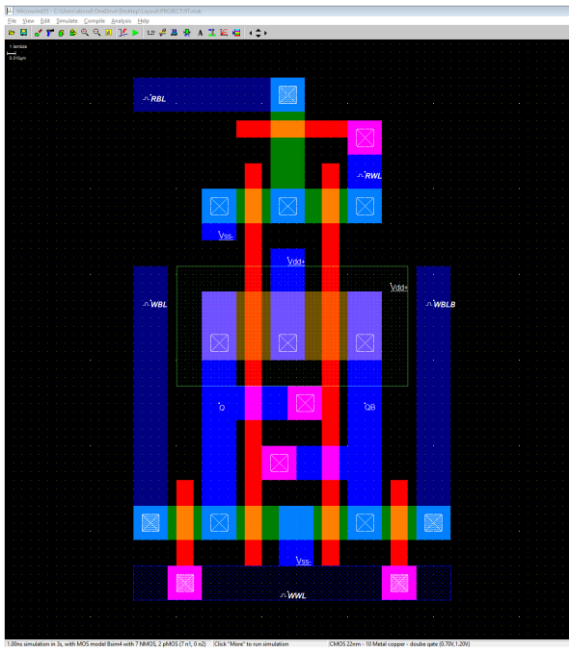


Figure V-8 inverter Layout And Simulation



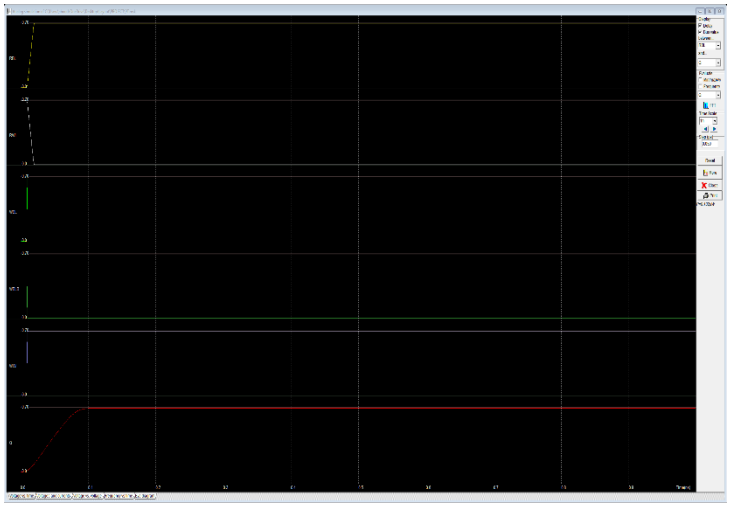
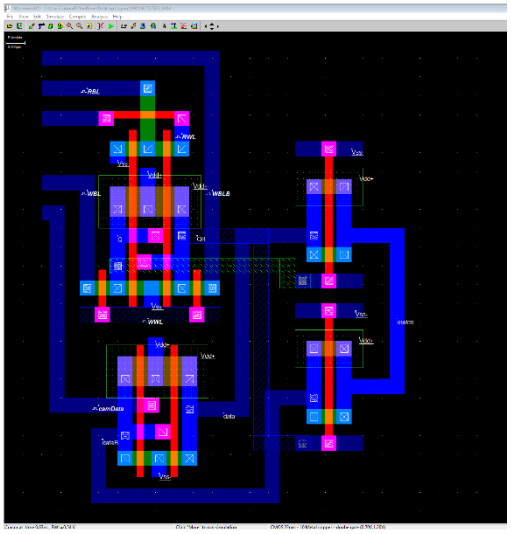


Figure V-9 9T Sram Layout And Simulation



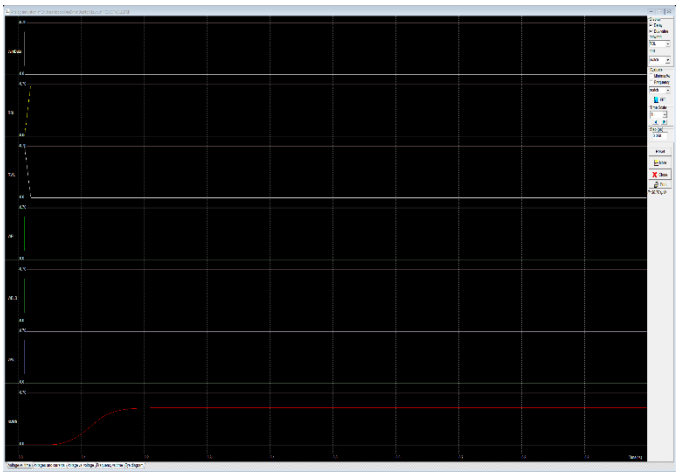


Figure V-10 CAM Cell Layout And Simulation

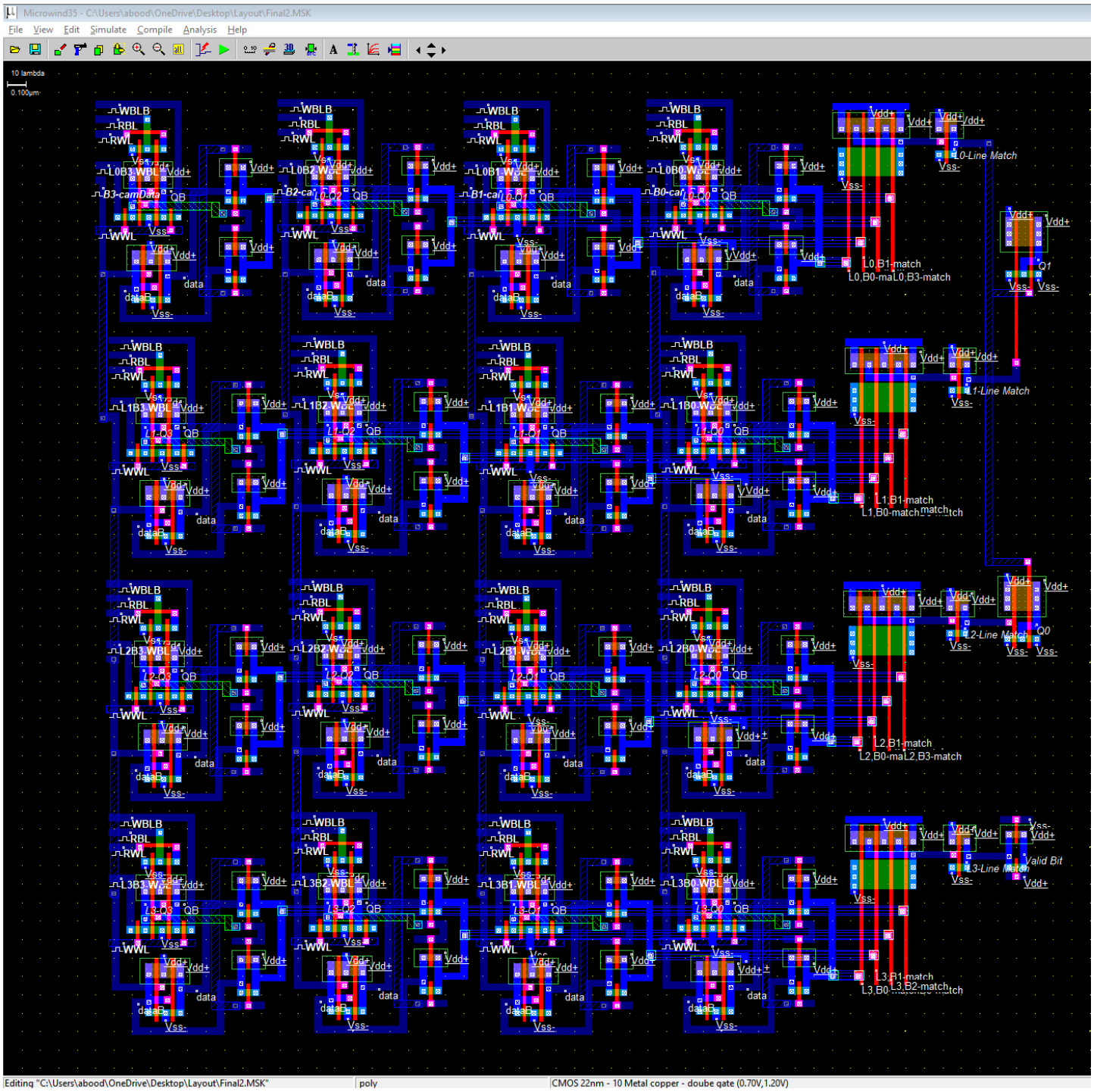


Figure V-11 4x4 CAM Layout And Simulation

In the Simulation at figure 1, the content of the Memory and Cam Data:

Line #	*Value*				*Address*
Line 0	0	1	1	1	00
Line 1	0	0	1	0	01
Line 2	0	1	1	0	10
Line 3	1	1	1	0	11

And the cam data is '1110' so the output should be '11' and valid bit should be '1'.

The delay for both simulation from the start to get the output was 215ps which is good for our CAM.

In the Simulation at figure 2, the content of the Memory and Cam Data:

Line #	*Value*				*Address*
Line 0	0	1	1	1	00
Line 1	0	0	1	0	01
Line 2	0	1	1	0	10
Line 3	1	1	1	0	11

And the cam data is '1110' so the output should be '11' and valid bit should be '1'.

Figure 1 CAM Simulation

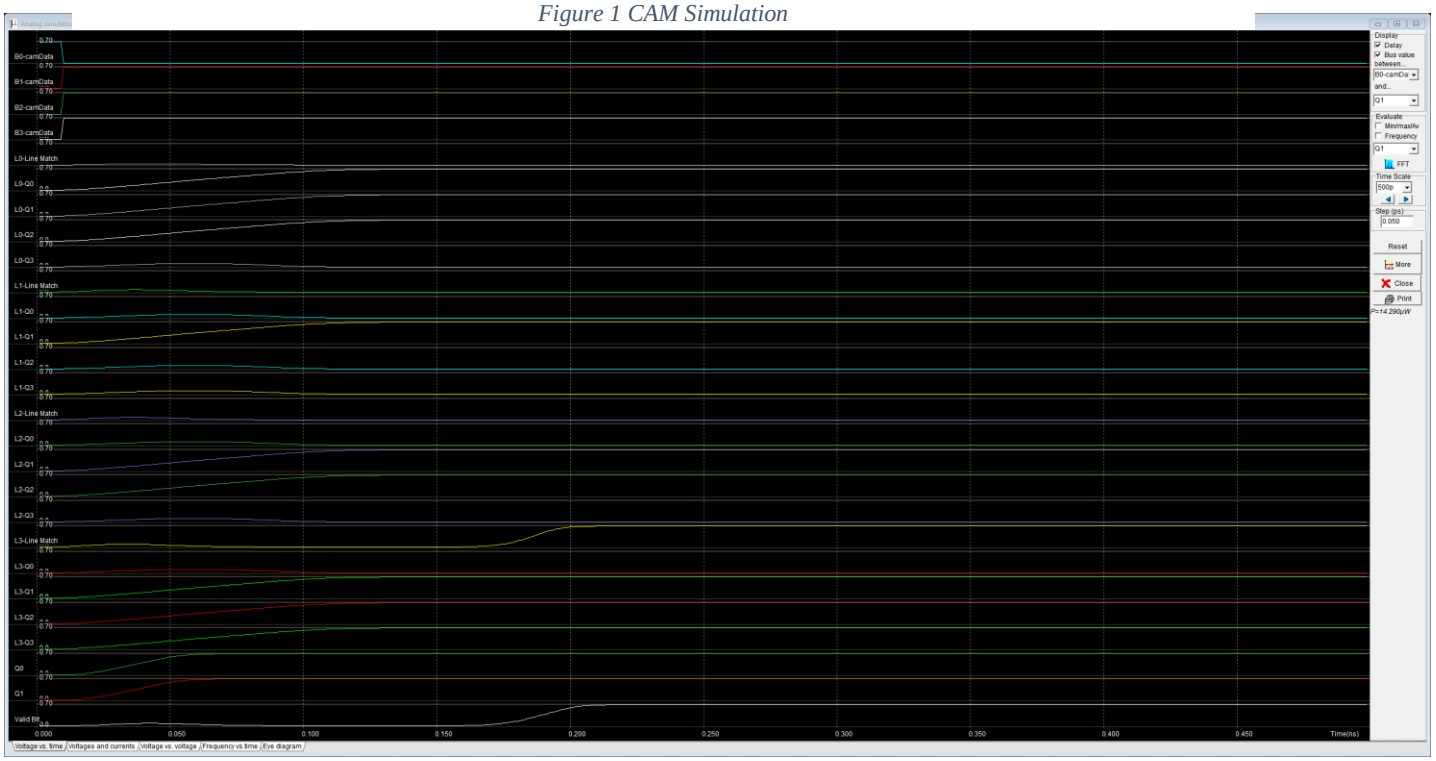
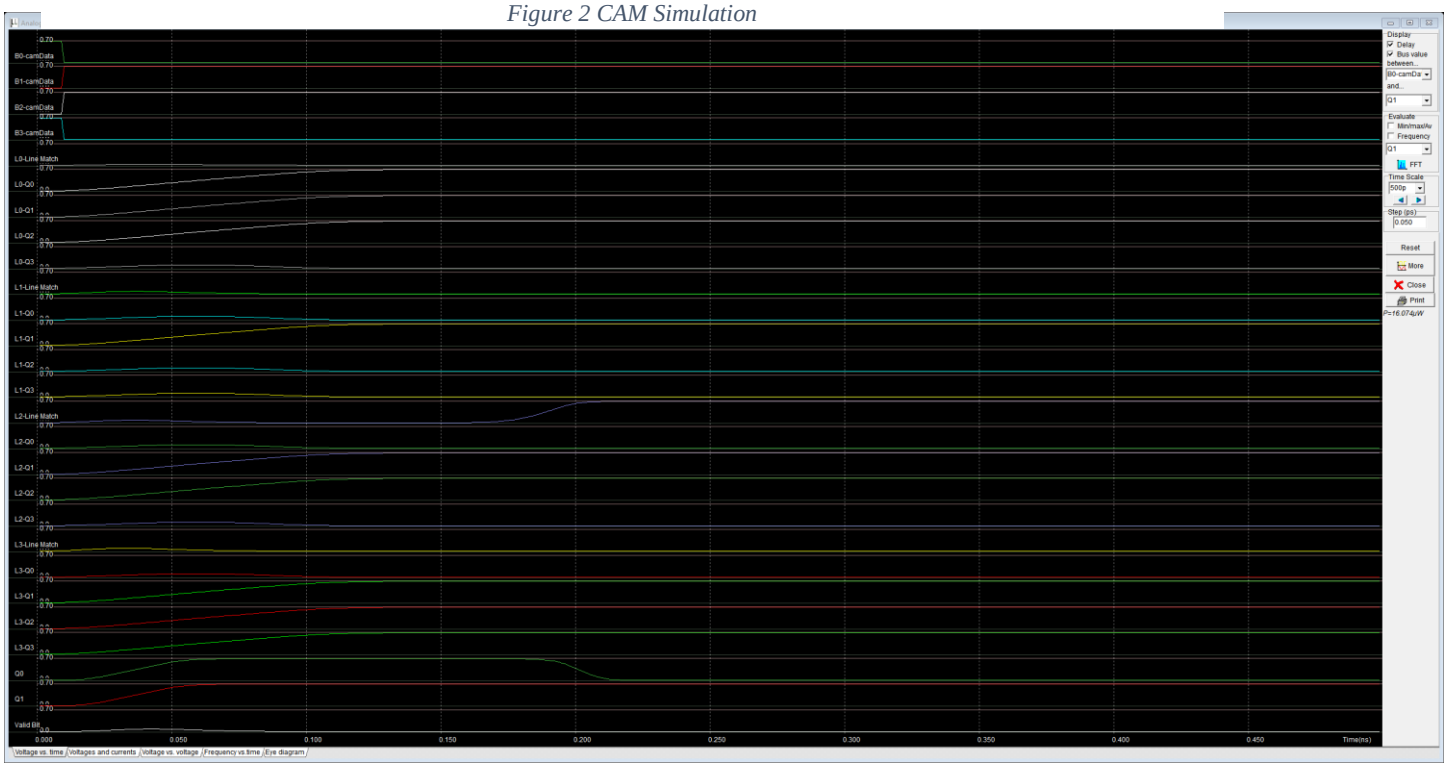


Figure 2 CAM Simulation



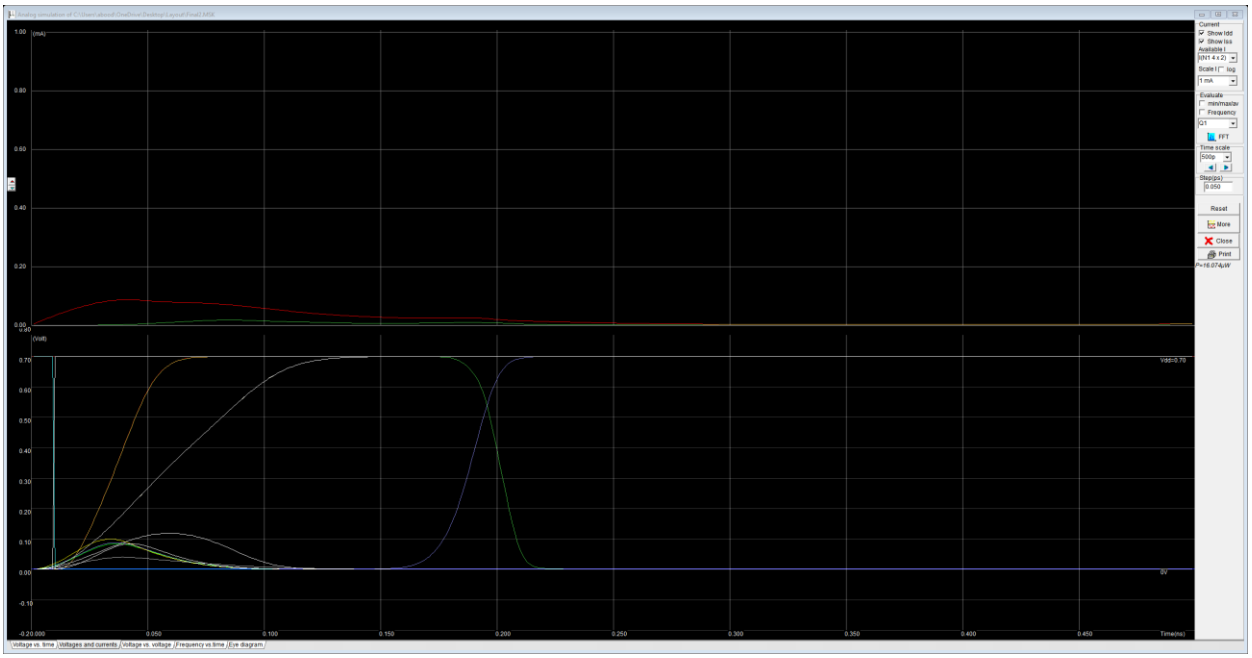


Figure V-14 CAM Voltage vs Current diagram

- **The Over All Area has been taken in consideration while designing all parts of the CAM, and the Over All Area was:**

Area = 5.1µ*5.2µ = 2.652 * 10^ (-11)

- **We have used 22nm tech with voltage = 1.2V and 0.7 considered as logic 1, and 0V as logic 0.**

- **The power consumption has been taken in consideration while designing all parts of the CAM, the max power (From Voltage vs Current figure above) Pmax = V*I = 0.7 V* 0.022mA = 15.4 µW**

VI. CONCLUSION

A Content Addressable Memory (CAM) is a memory unit that performs content matching instead of address decoding. CAM's are used in applications that required high speed lookup operations like network routers and cache controllers. However, the CAM search speediness comes at the cost of higher power consumption and larger silicon area. CAM system includes CAM cells, each having a compare-circuit and a memory bit-cell that stores complementary bits. Conventional CAM bit-cell implementation produces crowbar current during bit-cell write operation since there is one gate delay between WBL and WBLB which opens one pass gate before closing the second. In this paper, a novel CMOS CAM bit-cell implementation is presented to handle this problem and also to eliminate the need for routing camData and camData' to every CAM cell through using single-ended reference data (i.e., the inverted reference bit camData') instead of the complementary reference data. In addition, sharing resources across all cells reduces the total device width. The new implementation outperforms the conventional CAM cell implementation in terms of current saving without any performance impact in terms of area and delay.

VII. REFERENCES

[1] V. Tabatabaee, "Content addressable memories." http://www.isr.umd.edu/~vahidt/CAM.ppt.

[2] M. A. Scott Beamer, "Design of a low power content addressable memory (cam)," May 2009.

[3] K. Pagiamtzis, "Introduction to content-addressable memory (cam)." https://www.pagiamtzis.com/cam/camintro/.

[4] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (cam) circuits and architectures: a tutorial and survey," *Solid-State Circuits, IEEE Journal of*, vol. 41, pp. 712–727, March 2006.

[5] B. Mohammad, P. Bassett, J. Abraham, and A. Aziz, "Cache organization for embeded processors: cam-vs-sram," in *2006 IEEE International SOC Conference*, pp. 299–302, IEEE, 2006.

[6] B. Mohammad, M. T. Rab, K. Mohammad, and M. A. Suleman, "Dynamic cache resizing architecture for high yield soc," in *2009 IEEE International Conference on IC Design and Technology*, pp. 211–214, IEEE, 2009.

[7] D. GEORGIE, "Low power concept for content addressable memory (cam) chip design," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2013.

[8] A. Patwary, H. Greub, Z. Wang, and B. Geuskens, "Bit-line organization in register files for low-power and high-performance applications," in *Electrical and Computer Engineering, 2006. ICECE '06. International Conference on*, pp. 505–508, Dec 2006.

[9] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed lowpower cmos fully parallel content-addressable memory macros," *SolidState Circuits, IEEE Journal of*, vol. 36, pp. 956–968, Jun 2001.

[10] M. S. M. T. Sudalaimani1, "Design of high speed low power content addressable memory (cam) using parity bit and gated power matchline sensing," *International Journal for Research in Applied Science and Engineering Technology*, vol. 3, March 2015.

[11] S.-J. Ruan, C.-Y. Wu, and J.-Y. Hsieh, "Low power design of precomputation-based content-addressable memory," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, pp. 331– 335, March 2008.

[12] I. Corporation, "Low capacitance content-addressable memory cell," 01 1996.

[13] I. Corporation, "Content addressable memory constructed from random access memory," 10 2005.

[14] O. K. S.-K. K. D. A. Kamran Eshraghian, Kyoung-Rok Cho and S.M. S. Kang, "Memristor mos content addressable memory (mcam): Hybrid architecture for future high performance search engines," *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, vol. 19, August 2011.

[15] R. S. C. M. O. Yuanfan Yang, Jimson Mathew and D. K. Pradhan, "Low cost memristor associative memory design for full and partial matching applications," *IEEE TRANSACTIONS ON NANOTECHNOLOGY*, vol. 15, May 2016.

[16] K. M. A. Kabeer and T. Taha, "On-chip power minimization using serialization-widening with frequent value encoding," *VLSI Design*, 2014.

[17] D. Georgiev, "Low power concept for content addressable memory (cam) chip design," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, July 2013.

[18] K. Pagiamtzis and A. Sheikholeslami, "A low-power contentaddressable memory (CAM) using pipelined hierarchical search scheme," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 1512–1519, September 2004.

[19] C.-S. L. . J.-C. C. . B.-D. Liu, "A low-power content-addressable memory (cam) using pipelined hierarchical search scheme," *IEEE J. Solid-State Circuits*, vol. 39, pp. 1512–1519, September 2004.

[20] J.-C. C. C.-S. Lin and B.-D. Liu, "Design for low-power, low-cost, and high-reliability precomputation-based content-addressable memory," in *Proc. Asia-Pacific Conf. Circuits Syst.*, p. 319324, Circuits and Systems, 2002.