Portland State University

PDXScholar

Dissertations and Theses

Dissertations and Theses

2-12-1993

Smart Memory: An Inexact Content-Addressable Memory

Jack Lee

Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds

Part of the [Electrical and Computer Engineering Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Lee, Jack, "Smart Memory: An Inexact Content-Addressable Memory" (1993). *Dissertations and Theses*. Paper 4605.

<https://doi.org/10.15760/etd.6489>

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

AN ABSTRACT OF THE THESIS OF Jack Lee for the Master of Science in
Electrical and Computer Engineering presented February 12, 1993.

Title: Smart Memory: An Inexact Content-Addressable Memory

APPROVED BY THE MEMBERS OF THE THESIS COMMITTEE:

W. Robert Daasch, Chair

Michael A. Driscoll

Pah I. Chen

The function of a Content-Addressable Memory (CAM) is to efficiently search the information stored in the memory, by using hardware rather than software with a corresponding improvement in searching speed. This hardware allows a parallel search by matching the data stored in memory to a search key rather than sequentially searching address by address as is done in a Random Access Memory. Although existing CAMs are more efficient in finding relevant information than RAM, there are additional improvements that can be made to further improve its efficiency. For example,

previous CAMs use a word parallel searching scheme that can only identify exact matches. To find the best (closest) match, previous CAMs had to use bit serial approaches. Although still more efficient than RAM searching, these CAMs were limited by the word size (bit width) of the memory. Responding to this inefficiency, the CAM described in this thesis improves best-fit searching by using analog design in combination with digital design. This design retains a mismatch line to collect the result of the comparison of each bit of a word which is decoded by a simple flash A/D. This means that after a single operation the best-fit plus all words with zero to three bits of mismatch, are determined. This word/bit parallel searching makes this CAM more efficient than existing CAMs. The best-fit function of this CAM is good for database retrieval, communications and error correction circuitry. By using the high speed searching and the inexact match feature, this CAM also provides efficient sorting and set operations. The accumulated searching time is shortened when compared to regular CAM and RAM. The inexact CAM in this thesis is designed using mixed analog/digital design in a 2μ CMOS technology.

SMART MEMORY : AN INEXACT CONTENT-ADDRESSABLE MEMORY

by

JACK LEE

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL AND COMPUTER ENGINEERING

Portland State University
1993

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the thesis of Jack Lee presented February 12, 1993.

W. Robert Daasch, Chair

Michael A. Driscoll

Pah I. Chen

APPROVED:

Rolf Schaumann, Chair, Department of Electrical Engineering

Roy W. Koch, Vice Provost for Graduate Studies and Research

ACKNOWLEDGEMENTS

I am very thankful to my parents. They give me strongly support and encouragement during my graduate study. Their love creates my energy to work and study hard.

I would like to thank Dr. Robert Daasch, my advisor and friend. His brilliant guidance and patient encouragement helps me to succeed in my study.

I would also like to thank my roommate Wen-Tsung Yen and his brother Wen-Tzu Yen for their enthusiastic help of my experiment. By their help to connect the leads of my experiment integrated circuit chip, it becomes possible to finish my entire test.

Thanks to my friends and the faculty and staff of the Electrical Engineering Department for their friendly support and encouragement in many ways.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
 CHAPTER	 PAGE
I INTRODUCTION	1
Purpose	1
Communication	2
Set Operations	4
Thesis Overview	5
II DESIGN	7
Basic Ideas of the Design	7
CAM Array	9
Mismatch Detector	10
Priority Resolver	10
Summary	13
III CIRCUITS SCHEMATIC AND NETLISTS	16
CAM Array	16
Mismatch Detector	23
Priority Resolver	29
Whole Chip	33
Summary	37
IV EXPERIMENT RESULTS	40

Test Circuit	40
Match Line Voltage	43
Invert Voltage of the Mismatch Detector	43
Function of the Priority Resolver	45
Speed	47
Current	53
Summary	53
V CONCLUSION AND FUTURE WORK	55
REFERENCES	58
APPENDIX	59

LIST OF TABLES

TABLE		PAGE
I	An Example of the Set Operation	5
II	Simulation Results of the Voltage Level on the Match Line	19
III	Three Operating Regions for a Generic MOS Transistor	20
IV	Binary Code for the Number of Mismatched Bits	24
V	Match Line Voltage Simulated at Four Operation Frequencies	29
VI	I/O Port Arrangement of the IAM Chip	38
VII	Experimental Results of the Match Line Voltage for 16 Search Keys	45
VIII	Experimental Results of the Invert Voltage of the Mismatch Detector	46
IX	Experimental Results of the Noise Margin of the Flash A/D	46
X	Experimental Results of the Hit Finder Circuit	47
XI	Experimental Results of the Pass Control Circuit	49

LIST OF FIGURES

FIGURE		PAGE
1	A general communication system.	2
2	A communication example with noise in channel.	3
3	Venn diagram of the Hamming sets 0,1,2,3,>3.	4
4	Block diagram of major subsystem of IAM.	8
5	I/O ports of the three major subsystems of IAM.	8
6	Schematic of a static CAM cell.	9
7	Block diagram of the subcells of the priority resolver.	11
8	Comparison of ripple and tree structures for a priority resolver.	12
9	I/O map for the <i>Hit finder</i> cell.	13
10	I/O map for the <i>Pass control</i> cell.	14
11	The 16 inputs tree structure of the <i>Pass control</i> circuit.	14
12	Floor plan of the priority resolver for IAM chip.	15
13	Building block for a typical CAM cell.	17
14	The multivalued logic search circuit for the IAM.	18
15	Transistor size for a multivalued logic search circuit.	19
16	Layout of a CAM cell.	20
17	The saturation region of the transistors in the search circuit.	21
18	Schematic of a flash A/D converter.	24
19	Karnaugh maps of LSB and MSB of the mismatch detector.	25
20	The final circuit for the A/D converter.	25
21	The operating voltage of the CAM array and the mismatch detector.	26
22	Simulation results of the A/D converter.	26

23	Predict multivalue logic binary output levels of match line.	27
24	The match line voltage operation at 1 MHz.	27
25	The match line voltage operation at 6 MHz.	28
26	The match line voltage operation at 7 MHz.	28
27	The match line voltage operation at 8 MHz.	29
28	Circuit schematic for the <i>Hit finder</i> subcell.	30
29	Layout of a <i>Hit finder</i> cell.	31
30	Circuit schematic for a four word <i>Best finder</i> subcell.	32
31	Circuit schematic for 1 bit of the <i>Pass control</i> subcell.	33
32	Circuit schematic for 4 bits of the <i>Pass control</i> cell.	34
33	Layout of the <i>Pass control</i> cell.	35
34	Floor plan of the IAM chip.	36
35	Photomicrograph.	39
36	Block diagram of the IAM test circuit.	41
37	Photograph of the IAM test circuit.	42
38	Experimental result of the match line and invert voltage.	44
39	Measured Voltage level on the match line.	50
40	Measured Voltage level of MSB(Top) and LSB(Bottom).	50
41	Predict multivalue logic binary output levels of match line.	51
42	Measured voltage on a match line operating at 2 MHz.	51
43	Measured voltage of MSB(Top) and LSB(Bottom) operating at 2 MHz.	52
44	Measured voltage on a match line, MSB and LSB (Top-Down) at 3 MHz. ..	52
45	The match line voltage simulated at 3 MHz and connecting 7pF.	53

CHAPTER I

INTRODUCTION

PURPOSE

A memory system, which can rapidly choose the stored item that most closely matches a given input, is useful in communications, signal processing and artificial intelligence [1]. An earlier approach that performs this function, the Pipelined "Best Match" Content-Addressable Memory (CAM) [2], used a pipelined CAM memory circuit to do this function. Another approach is the Versatile Data String-Search VLSI [3] which used a finite-state machine logic.

For the pipelined match approach, the actual time from input to output is a linear function of the pipeline length because the input has to be propagated to the output through all of its words. The finite-state machine logic approach takes more than one operation to get the match result. If we design a CAM system which can find the match result in one operation and if we can reduce the ratio of the propagation time over the number of words, then we can use the CAM to do a faster search.

The Inexact CAM (IAM) is the memory system designed to achieve this goal. This IAM has been designed, fabricated and tested. The design and test are described in later Chapters. In this Chapter, I will introduce the IAM by using two examples. One example uses the IAM to do the error detection and correcting for communication. The other example is using the IAM to do set operations since the CAM is the ideal medium to implement sets in the same way RAMs are ideal to implement arrays [4].

COMMUNICATION

Data transmission plays an important role in communications. One can easily imagine the consequences of errors on telemetry, or remote control signals for maintenance a satellite in space, or the inconvenience due to errors in compressed picture transmission. However the transmitted data are often corrupted by various sources of noise (errors). The noise may be human error, lightning, thermal noise, imperfections in equipment, etc., and may result in error so that the data received is different from that sent. Error-correcting codes are designed to encode the data, by adding a certain amount of redundancy to the message, so that the original message can be recovered if errors have occurred. Error-correcting codes are used in several domains such as satellite channel transmissions, magnetic recordings and compact disks. High speed transmissions and high density recording need a strong protection against errors. A general communication system is shown in Figure 1. The message is encoded by the encoder which adds some number of bits to the message to help the error detection. When the decoder receives this message, the error will be detected and corrected by the decoder if it is correctable. Thus, the system transfers messages in a reliable way. There are many ways to compose a reliable message. We can transfer the same message three times and compare these three received results to find out the correct one by statistics. The high quality equipment can reduce the chance of error but it may be very expensive. To transfer the same message many times increases the transfer time for each message. By using an error correction system, a message will be transferred only once.

Figure 1. A general communication system.

An example is shown in Figure 2. A message 'YES' is to be sent to the user. The encoder codes a 'YES' by '00000' and a 'NO' by '11111'. If there are two errors during the transfer, a possible received message would be '01001'. The decoder can correct the error by using this '01001' as a search key to search the codes '00000' and '11111' to find out the code with the best match. The best match vector will be '00000'. The user then gets the corrected message 'YES'. The best match feature of IAM makes it useful in this correction system.

Figure 2. A communication example with noise in channel.

In an error correcting communication system, a binary code is transferred. Each of these binary codes has two parts, information code and redundancy code. The information code is the original information to be transferred. The redundancy code is the code to be added in the communication process to create the error correcting ability for the communication. Technically, the redundancy code is added to enlarge the distance between these information codes. The distance between two binary codes is the total number of bits that differ between them. This additional distance provides increased error tolerance for each code than what it had before. It also increases the number of the redundancy codes and takes additional time to transfer it. The increased code distance also increases the error correcting ability. The balance between the code transfer time and the error correcting ability depends on the way it is encoded. There are many ways

to encode a message. The Hamming code [5] and the Golay code [6] are two examples.

The IAM can correct single, double or triple errors for an error correcting communication system in one operation by its ability to find one, two and three bits mismatch. This characteristic makes the IAM attractive application in communication.

SET OPERATIONS

Using the idea of 'distances' from the previous section, a picture of a set can be imagined as in Figure 3. Each specific distance represents a subset. A subset can be defined as the distance between search key and elements. A distance of '0' means this element exactly matches the search key.

Figure 3. Venn diagram of the Hamming sets 0,1,2,3,>3.

The distance can be defined by a tag of each element or by the element itself. For example, the error correcting is one use that operated the distance of the element. Here is an example that operates the distance on the tag of each element. Define the universe 'U' as 'living things' and '00000000' is its code. Its subset could be defined related to each tag as in Table I. If we search the living things with the distance equal to one, we can get horse, cow, dog and cat. The spider and ant could be retrieved by a search of living

things with the distance equal to three.

A design objective for the IAM is word/bit parallelism which means all bits and all words in the memory are processed in one operation. For example, in a RAM of 1K by 16, a search for the words with two bits of mismatch takes 16000 operations for the worst case because it is a word/bit serial search. It takes 225 operations for a regular CAM if it has no accumulator. If a regular CAM has accumulator, it takes 16 operations because a regular CAM doing a word parallel/bit serial search. In the IAM, this search only take one operation because this IAM processed word/bit parallel. All bits and all words are processed in one operation.

TABLE I
AN EXAMPLE OF THE SET OPERATION

Tag	Elements	Distance	Subset
00000010	horse	1	1
00010000	cow	1	1
00101000	tulip	2	2
01110000	spider	3	3
01000110	ant	3	3
01100000	violet	2	2
00001100	grass	2	2
00100000	dog	1	1
00000100	cat	1	1
00110110	bacteria	>3	4
01010011	snake	>3	4

THESIS OVERVIEW

The IAM described in this thesis provides a single operation inexact match function and includes the exact match function of a regular CAM. The IAM is designed to find one, two or three bits of mismatch to the search key. A mixed analog/digital circuit design is used to minimize the circuit size. In the next Chapter, the design of the Inexact

CAM is introduced by using function blocks. The circuit of each block is described in Chapter III. Test results of this Inexact CAM is in Chapter VI.

CHAPTER II

DESIGN

This Chapter introduces the design of an inexact CAM. The ability to decode levels of mismatch during a search operation is the unique feature of this CAM. To approach this feature, the IAM will be designed to have the following functions:

- It can search the words with an exact match to the search key.
- It can search the words with one, two and three or more bits' mismatch to the search key.
- It can search the word which is the nearest match to the search key.
- It can do all of the above in one operation.
- It can report the search results sequentially or as a parallel words.

The circuit will be constructed from three subsystems that are introduced in the following sections. An overview of this IAM design describes the relationship among these parts.

BASIC IDEAS OF THE DESIGN

A CAM Array is needed to store words. After a search key is applied to this CAM, we need a Mismatch Detector Circuit to identify the number of mismatched bits for each word of the CAM Array. We also need a Priority Resolver Circuit to do the inexact match functions after we get the number of mismatched bits of each word from the Mismatch Detector. The last circuit we need is a circuit which can find the best matched word. This circuit is included in the Priority Resolver Circuit. All these three subsystems are the main points of the inexact CAM design as shown in Figure 4.

Figure 4. Block diagram of major subsystem of IAM.

In Figure 5., information is written into the CAM Array by the control *Word select* and *Data I/O* ports. When a search key is applied, the CAM Array passes the mismatch information of each word to the Mismatch Detector. The command *Search start* enables all of the *Match lines*. The Mismatch Detector decodes the mismatch information of each word from *Match line* into a binary code. These binary codes represent the numbers of mismatched bits of each word and are passed to the Priority Resolver.

Figure 5. I/O ports of the three major subsystems of IAM.

There are three subcircuits in the Priority Resolver. The first subcircuit, 'Mark', finds the words that meet the specific requested mismatch number from the *Search type command*. We give this circuit the name *Hit finder*. The second subcircuit of the Priority Resolver gets the result from *Hit finder* and pass one or all marks to

its output according to the *Search type command* request. We call this circuit a *Pass control* circuit. The last subcircuit of the Priority Resolver finds the smallest number of bits mismatched to the search key from all these mismatched number. This circuit is called *Best finder*. We will continue the discussion of these three blocks of Figure 5 in the following section.

CAM ARRAY

The CAM array is a two dimensional grid of static memory CAM cells. Figure 6. illustrates a basic CAM cell with static memory. Because the IAM circuit design is focused on the special function, inexact match, the function of a basic CAM cell will be described briefly.

Figure 6. Schematic of a static CAM cell.

A bit of information is stored in a static memory cell that is selected by *Word Select* (WS_j). The comparison is implemented with a pass-transistor XOR gate

and $match_j$ is evaluated by a NOR of the $match_{ji}$ of each of the CAM cells of word j . A pull up for the NOR is not included in Figure 6. After we put the CAM cell together in a two dimensional array, we get a CAM array. In this test model, the CAM array will be eight words of four bits each.

If we select the suitable size (W/L) pull up for the *match line* and let the $match_{ji}$ bit pull down only part of its voltage, we can get the number of miss bits by final voltage divided level on the *match line*. By this mixed analog and digital design, we store additional information on one wire. This means that the number of wires can be reduced. For most integrated circuit designs, we prefer additional transistors instead of additional wires.

MISMATCH DETECTOR

The second part is the Mismatch Detector which converts the *match line* voltage level into binary code that indicates the number of mismatched bits. In this experiment, Mismatch Detector can detect zero, one, two or three bits of mismatch for each word. So the converted binary code of each word needs two bits to represent the number of mismatched bits.

PRIORITY RESOLVER

There are three subcircuits in Priority Resolver. They are *Best finder*, *Hit finder* and *Pass control*. The first subcircuit, *Best finder*, inputs the binary codes from Mismatch Detector and selects the best match to the search key from all words in the memory. The second subcircuit, *Hit finder*, compares the binary codes from Mismatch Detector to the user's instruction code that selects the hits for this comparison. Every hit will assert high on the output, Hit_j line, of the *Hit finder*. Whenever the user wants to find the best matched word, the result of the first circuit, *Best finder* is used

instead. The third subcircuit, *Pass control*, selects the highest priority or all of the hits to the output port of the Priority Resolver.

Figure 7 shows a simple example of how these three subcircuits work together. The binary codes from Mismatch Detector are passed through the *Best finder* and the *Hit finder*. In this example, the output of the Mismatch Detector is shown on the left. The *Best finder* selects the one bit mismatch '01' and forwards this to the *Hit finder*. The user instruction selects two bit mismatches from the input which is requested to identify the highest priority for a final output of the Priority Resolver '1000'.

Figure 7. Block diagram of the subcells of the priority resolver.

Now we want to choose a circuit structure for the Priority Resolver. Any circuit that has to compare results between words includes a propagation time. The Priority Resolver has such a compare function and we have to choose a circuit structure that limits propagation time. Considering the possibilities of a ripple or a tree structure as Figure 8, we have chosen the tree structure because this structure has a shorter critical path from input to output.

The second cell we want to consider is *Hit finder* cell. A *Hit finder* cell is shown in Figure 9. This cell compares the binary codes of four words to the instruction code and asserts the *Hit* flag.

Pass control circuit is the last part of Priority Resolver. *Pass control* can be instructed to pass all highest priority information from its input ports to its output ports. It also has a *Enable* port as its input port, see Figure 10, and a *Hit exist* port as an output port for the operation of a tree structure.

Figure 8. Comparison of ripple and tree structures for a priority resolver.

Figure 11 shows the tree of a *Pass control* circuit. If the instruction is *Pass all*, the *Hit inputs* will be passed to the *Hit outputs* directly. If the instruction is *Pass one*, each of these nine cells can only pass one of its *Hit* to its output. This *Hit* will be the top most one from its four words. From Figure 11, if the first cell and the third cell have no

'hit' to be passed to its output and the second and the fourth cells have 'hit' to be passed to its output, only the second and the fourth cell will assert its *Hit exist* port. The ninth cell gets the *Hit exist* from cell one to cell four and passes only the top most one *Hit* from its input port to its output. This *Hit output* from the ninth cell will send to the sixth cell to enable it. The *Hit output* of the fifth, seventh and eighth cell will be disabled by the control from the ninth cell. So only the topmost *Hit out* of the second cell can be passed to the *Hit output* of the eighth cell.

Figure 9. I/O map for the *Hit finder* cell.

These three circuits should be put together to become a Priority Resolver. Both of *Best finder* and *Pass control* have a tree structure. When we want to put the trees together, a proper organization can simplify their connection. The way we try to organize the two trees is to mix them together as a single tree. Figure 12 is the floor plan of the priority resolver for the eight words of our test model.

SUMMARY

The ability of decoding levels of bit's mismatch during a search operation is the unique feature of this CAM. To decompose a circuit into cells helps the designer to design, simulate, debug and layout the circuit piece by piece. It reduces the design period because the designer can concentrate their effort to each component. The Priority Resolver circuit is the most complex circuit among these three circuits because it used

tree structure for the comparisons of best and highest priority.

In the next Chapter, we will present the circuit design in transistor level. All of the design in the next Chapter will follow the idea in this Chapter.

Figure 10. I/O map for the *Pass control* cell.

Figure 11. The 16 inputs tree structure of the *Pass control* circuit.

Figure 12. Floor plan of the priority resolver for IAM chip.

CHAPTER III

CIRCUITS SCHEMATIC AND NETLISTS

In this Chapter we will implement the circuits that were introduced in Chapter II. The circuits will be separated into the three parts described in Chapter II. The circuit design of each part will be described in detail with its schematic and netlists. Simulation results will also be presented in this Chapter. After each part is described, a whole chip description will give us a clear image of this chip. This design used Spice Version 3d2 (LEVEL 2 MOSFET models), Magic Version 6.3, PRESIM Version 5.0 and RSIM Version 5.0 simulation and layout tools.

CAM ARRAY

In this section, we will present the logic design for these parts. Figure 13 is a typical CAM cell. The six-transistor memory stores one bit of information. The exclusive OR circuit compares the stored information with the search key. The compared result is one input to the NOR gate. The CAM cell is assembled into words with their match line joined as described in Chapter II. The voltage on the match line is divided to represent the number of bits mismatched. By choosing the size of the pull down transistors in the NOR gate, the voltage of the match line can be adjusted. We will use the minimum size devices ($L=2\mu\text{m}$, $W=3\mu\text{m}$) for the remaining transistors.

To determine the pull down transistor size, we have to also consider the pull up transistor size. Figure 14 shows the circuit relation between the pull up transistor M0 and the pull down transistors M1 to M5. When the $\overline{\text{Search}}$ is triggered, current will go through the transistor M0 from Vdd to charge the match line. If any bit is mismatched to

the search key, the voltage of the match line will be divided by the ratio of pull up and pull down transistors. There are two things we have to do for the circuit. The first thing is to control the current dissipation of each word by adjusting the size of M0. The second thing to do is to adjust the voltage level on the match line for varying numbers of bits mismatched by resizing the pull down transistors M1 to M5. In this experiment the current dissipation for each word was held below $100\mu\text{A}$ by its SPICE simulation and the total current dissipation for a 1K words would be less than 0.1A. The simulation result shows that the current will be limited to less than $87\mu\text{A}$ per word if the size of M0 is $W/L=3/5$.

Figure 13. Building block for a typical CAM cell.

For M0 ($W/L = 3/5$), we sized the pull down transistors from M1 to M4 using SPICE simulation. Table II shows the voltage level before and after transistor M5 is added. The simulation result shows that we should put a transistor M5 into this circuit. Because without M5, the voltage for a one bit mismatch is equal to 3.296 volts which

requires a flash inverter with either a long pull down or a wide pull up. If we put the transistor M5 into the circuit, the match line voltage for a one bit mismatch is 2.809 volts. Which means that we can use a flash inverter with nearly minimum size pull up and pull down. We want the Mismatch Detector to work quickly with a small size flash inverter. To do this we must install the pull down transistor M5. The size of the transistor M5 is $W/L=3/25$ in this simulation. The size of the transistors M1 to M4 is $W/L=3/17$.

Figure 14. The multivalue logic search circuit for the IAM.

Figure 15 shows the size of these transistors of a CAM word. The separation of the voltage levels is large enough to allow the inverted voltage of a flash A/D converter to respond. We will find out flash inverters by set its inverted voltage between each voltage levels on a match line when we design the Mismatch Detector circuit.

After the size of each pull down transistor, M1 to M4, of the CAM are determined, we can layout a word of the CAM. Figure 16 is the layout of a CAM cell. For the 2 micron process, the size of this CAM cell is $50\mu\text{m}$ in height and $76\mu\text{m}$ in width.

The point we are concerned with now is the region of operation of these transistors. For 0, 1, 2, 3 bits of mismatch functions of this IAM, are the transistors M1 to M4

in the linear or saturated region? And the more importantly, does the region of operation change for different numbers of mismatch bits? We will investigate these questions by the first-order phenomena in MOS transistors [7]. Table III shows the three first order regions of operation for a MOS transistor.

TABLE II
SIMULATION RESULTS OF THE
VOLTAGE LEVEL ON THE MATCH LINE

Number of mismatch	Voltage level	
	Without M5	With M5
0	5.00 volt	4.11 volt
1	3.30 volt	2.81 volt
2	1.60 volt	1.55 volt
3	1.00 volt	1.00 volt

Figure 15. Transistor size for a multivalue logic search circuit.

For this analysis, there is one pull up pMOS transistor M0 and five pull down nMOS transistors from M1 to M5 as shows in Figure 15; V_M is the match line voltage; V_{TN} is the zero-bias threshold voltage of all of the nMOS transistors and V_{TP} is the zero-bias threshold voltage of the pMOS transistor.

Figure 16. Layout of a CAM cell.

TABLE III

THREE OPERATING REGIONS FOR A GENERIC MOS TRANSISTOR

1. Cutoff:	$(V_{GS} - V_{TE}) \leq 0,$	(1)
	$i_D = 0.$	
2. Nonsaturation:	$(V_{GS} - V_{TE}) \geq V_{DS},$	(2)
	$i_D = K'S (2(V_{GS} - V_{TE})V_{DS} - V_{DS}^2).$	(3)
3. Saturation:	$(V_{GS} - V_{TE}) \leq V_{DS},$	(4)
	$i_D = K'S (V_{GS} - V_{TE})^2.$	(5)

The pMOS transistor M0 will operated in its saturation region when

$$V_{GS} - V_{TE} \leq V_{DS} \quad (4)$$

which with the proper absolute value replacements gives

$$V_{dd} - |V_{TP}| \leq V_{dd} - V_M$$

$$V_M \leq |V_{TP}| \quad (6)$$

the nMOS transistors M1 to M4 will operated in their saturation region when

$$V_{GS} - V_{TE} \leq V_{DS} \quad (4)$$

in this case V_{GS} is approximately $V_{dd} - V_{TN}$ because, in Figure 13, the *bit* or $\overline{bit}$ voltage V_{dd} is reduced by a V_{TE} because of the pass transistor in the exclusive OR gate. Replacing V_{GS} , V_{TE} and V_{DS} we can get

$$(V_{dd} - V_{TN}) - V_{TN} \leq V_M \quad (7)$$

$$V_M \geq V_{dd} - 2V_{TN}$$

the diode nMOS transistor M5 will always operation in its saturation region because its

$$V_{GS} - V_{TE} \leq V_{DS} \quad (4)$$

its $V_{GS} = V_M$, so

$$V_M - V_{TN} < V_M$$

Because Eq.(6) and (7) are relations for the match line voltage V_M , we can find the saturation region shown in Figure 17.

Figure 17. The saturation region of the transistors in the search circuit.

We know that $|V_{TP}|$ is always less than $V_{dd} - 2V_{TN}$ because in the MOSIS N14Q SPICE model V_{TP} is -0.80 volt and V_{TN} is 0.75 volt. When

$$V_{dd} - 2V_{TN} \geq V_M \geq |V_{TP}| \quad (8)$$

transistors M0 to M4 are all operating in their nonsaturation region. From this condition we can determine the number of mismatched bits that shifts one or more of the transistors into saturation.

When M0 is in saturation and M1 to M4 are in nonsaturation the number of bits mismatched can be evaluated by letting $V_M = |V_{TP}|$. If we consider the match line node, the current from the pull up transistor M0 must equal to the total current through the m mismatch transistors M1 to Mm. The total match line current can be calculated using Eq.(3) and Eq.(5)

$$K'_P S_P (V_{GS} - |V_{TP}|)^2 = m K'_N S_{N1} (2(V_{GS} - V_{TN})V_{DS} - V_{DS}^2) + K'_N S_{N2} (V_{GS} - V_{TN})^2$$

where S_P , S_{N1} and S_{N2} are the shape factors for M0, M1 to M4 and M5 respectively. Substituting for V_{GS} and V_{DS}

$$K'_P S_P (V_{dd} - |V_{TP}|)^2 = m K'_N S_{N1} (2(V_{dd} - 2V_{TN})(|V_{TP}| - V_{TP}^2) + K'_N S_{N2} (V_{dd} - 2V_{TN})^2 \quad (9)$$

K'_P and K'_N are the transconductance of pMOS and nMOS respectively. V_{TN} and V_{TP} are the zero-bias threshold voltages of nMOS and pMOS transistors respectively. Their values are in APPENDIX. Equation (9) can be solved for the number of resolvable mismatch bits, $m=3.57$. For the model and this design, $m = 3$. So M0 is working in its nonsaturated region.

When M0 is nonsaturated and M1 to Mm are saturated, the number of bits mismatched m is found by letting $V_M = V_{dd} - 2V_{TN}$ (See Figure 17). The match line currents are calculated by Eq.(3) and (5) as

$$K'_P S_P (2(V_{GS} - |V_{TP}|)V_{DS} - V_{DS}^2) = m K'_N S_{N1} (V_{GS} - V_{TN})^2 + K'_N S_{N2} (V_{GS} - V_{TN})^2$$

Replace the V_{GS} and V_{DS} by V_{dd} and $(V_{dd} - V_M)$, we get Eq.(10)

$$\begin{aligned} & K'_P S_P (2(V_{dd} - |V_{TP}|)(V_{dd} - V_M) - (V_{dd} - V_M)^2) \\ & = m K'_N S_{N1} (V_{dd} - 2V_{TN})^2 + K'_N S_{N2} (V_{dd} - 2V_{TN})^2 \end{aligned} \quad (10)$$

The number m is 0.73 after calculate Eq.(10). This means that the current provided by the pull up transistor is not enough to saturate a pull down transistor. Therefore the pull up transistors, M1 to M4, will always operation in their nonsaturation region. This is reasonable because if these nMOS transistors are operation in their saturation region, their

$$V_{DS} \geq V_{GS} - V_{TE}$$

$$\begin{aligned} V_{DS} &\geq V_{dd} - 2V_{TN} \\ &\geq 3.5 \text{ volt} \end{aligned}$$

compare the voltage with the one bit mismatch of the simulation result in Table II, we will find that when there is one pull down transistor turn ON, the match line voltage is 2.8 volt. For this transistor, $V_{GS} - V_{TN}$ is 3.5 volt and V_{DS} is 2.8 volt. Its

$$V_{GS} - V_{TE} > V_{DS} \quad (2)$$

which agrees with the nonsaturation condition in Table III.

This IAM can determine up to 3 bits of mismatch. And all devices, the pull up and pull down transistors M0 to M4, are working in their nonsaturation region.

MISMATCH DETECTOR

Mismatch Detector circuit decodes the voltage information from each *match line* into a two bit binary code. We use a flash A/D converter to identify each voltage level of the *match line*. For zero to three mismatches, there are four voltage levels on the *match line*. If we want to identify them using a flash A/D converter, we have to preselect the inverter voltage of each flash inverter to be between each of these four levels. Figure 18 shows the circuit of a flash A/D converter. The transistor's sizes shown in the figure were determined by SPICE simulation.

TABLE IV shows the encoding of the most significant bit (MSB) and the least significant bit (LSB). From Figure 19, the logic function of the binary code can be seen to be

$$\begin{aligned} MSB &= Out_2 \\ LSB &= Out_1 \cdot \overline{Out_2} + Out_3 \end{aligned}$$

If we translate the expression of LSB for the CMOS pull down circuit by the DeMorgan's Laws. The LSB will become

$$LSB = Out_1 \cdot \overline{Out_2} + Out_3$$

Figure 18. Schematic of a flash A/D converter.

TABLE IV

BINARY CODE FOR THE NUMBER OF MISMATCHED BITS

Number of bits mismatched	Output of inverter			Encoded number	
	Out_1	Out_2	Out_3	MSB	LSB
Totally match	0	0	0	0	0
One bit mismatch	1	0	0	0	1
Two bits mismatch	1	1	0	1	0
Three bits mismatch	1	1	1	1	1

$$\begin{aligned}
 LSB &= \overline{Out_1} \cdot \overline{Out_2} + Out_3 \\
 &= \overline{(Out_1 \cdot Out_2)} \cdot Out_3 \\
 &= \overline{(Out_1 + Out_2)} \cdot Out_3
 \end{aligned}$$

The combined flash A/D converter is shown in Figure 20. Simulation results for minimum sized transistors in the LSB and MSB circuits are shown in Figure 21. The DC sweep simulation of the match line voltage in Figure 22 shows correct decoding of the match line voltage.

Figure 23 is the predict multivalue logic binary output levels of match line. Figures 24-27 show the response of the LSB and MSB circuits to match line at the frequencies 1, 6, 7 and 8 MHz respectively. Comparison with Table V shows that the LSB and MSB are correctly digitized at frequencies below 7 MHz.

Figure 19. Karnaugh maps of LSB and MSB of the mismatch detector.

Figure 20. The final circuit for the A/D converter.

Figure 21. The operating voltage of the CAM array and the mismatch detector.

Figure 22. Simulation results of the A/D converter.

Store 0000 and searched by 16 patterns	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Number of bits mismatched	4	3	3	2	3	2	2	1	3	2	2	1	2	1	1	0
Voltage on match line by each number of bits mismatched

Figure 23. Predict multivalue logic binary output levels of match line.

Figure 24. The match line voltage operation at 1 MHz.

Figure 25. The match line voltage operation at 6 MHz.

Figure 26. The match line voltage operation at 7 MHz.

Figure 27. The match line voltage operation at 8 MHz.

TABLE V

MATCH LINE VOLTAGE SIMULATED
AT FOUR OPERATION FREQUENCIES

Number of bits mismatched	Voltage on the match line (volt)			
	1 MHz	6 MHz	7 MHz	8 MHz
Totally match	4.10	4.07	4.05	4.00
One bit mismatch	2.81	2.65	2.60	2.54
Two bits mismatch	1.69	1.48	1.48	1.48
Three bits mismatch	1.10	1.00	1.00	1.00

PRIORITY RESOLVER

There are three circuit parts in the Priority Resolver as described in Chapter II; *Hit finder*, *Best finder* and *Pass control* cells. The *Hit finder* cell has four 2 bit mismatched inputs and an instruction input port. The four 2 bit mismatched inputs are

The second cell we want to design is the *Best finder* cell. This cell has four pairs of LSB/MSB as its input ports. We want to use a NOR gate as its input gate for the working speed of this cell. For a multiple input gate, we want to use NOR gate instead of NAND gate because the reaction speed of a NOR gate is faster than that of a NAND gate if they are both designed with minimum size transistors.

Figure 29. Layout of a *Hit finder* cell.

We find that it is easier to find the maximum number than to find the minimum number if the NOR gate is used in this circuit. The *Best finder* cell is designed to find the minimum number from all of the input numbers. For the working speed of this circuit, we design this cell to find the maximum number and give it reversed two bits binary

input code. When using this cell, we give it reversed two bits binary code as its input and get the maximum number from its output. Then reverse this output to get the *Best*, minimum, number. Figure 30 shows the circuit of a *Best finder* cell. In this circuit

$$\begin{aligned} Out_1 &= \overline{\bar{m}_0 + \bar{m}_1 + \bar{m}_2 + \bar{m}_3} \\ Out_2 &= \overline{\bar{m}_0 \bar{l}_0 + \bar{m}_1 \bar{l}_1 + \bar{m}_2 \bar{l}_2 + \bar{m}_3 \bar{l}_3} \\ Out_3 &= \overline{\bar{l}_0 + \bar{l}_1 + \bar{l}_2 + \bar{l}_3} \\ \overline{MSB}_{best} &= \overline{Out_1} \\ \overline{LSB}_{Best} &= Out_2 \cdot Out_1 \cdot Out_3 \end{aligned}$$

Figure 30. Circuit schematic for a four word *Best finder* subcell.

The *Pass control* cell is the third cell we want to design for the Priority Resolver circuit. The function of this cell is to pass its Hit_{in} to its Hit_{out} by user request to pass one or all of them. Figure 31 shows its circuit. Its equation is

$$\begin{aligned} Hit_{out} &= Hit_{in} \cdot Enable_{in} \\ Enable_{out} &= \overline{All} + Enable_{in} \cdot \overline{Hit_{in}} \end{aligned}$$

By putting four of these subcells together, we have the *Pass control* subcircuit as Figure 32. Figure 33 shows its layout.

Figure 31. Circuit schematic for 1 bit of the *Pass control* subcell.

WHOLE CHIP

These cells are assembled as three parts, CAM Array, Mismatch Detector and Priority Resolver for a prototype of eight words. Using the 40PC22X22 Standard Frame from MOSIS. The chip is a 2 micron SCN (N-well) TinyChip and the pad set was designed with the MOSIS scalable CMOS design rules (Rev 6) at a lambda of 1.0 micron. Figure 34 is the pads arrangement of this chip.

The usage of these IO ports is arranged as Table VI. Four analog pads are used for the match line of the first four words because the voltage on the match line will be measured during testing. The IO pad will be a digital input pad when its IO control is to be set on '0' or it will be a digital output pad when their IO control is to be set on '1'. Two of these pads are the control pads which control a group of pads to become the input or output pads. PIN 4 is to control the IO of Pin 6 to 9 and PIN 11 to 14. These pads will be controlled as input pads when we write into the CAM Array through *bit* and $\overline{bit}$.

Figure 32. Circuit schematic for 4 bits of the *Pass control* cell.

These pads will be controlled as output pads when the *Hit_{out}* of the Priority Resolver is tested. PIN 34 is to control PIN 1 to 3 and PIN 36 to 40. They will be controlled as output pads for the test of the Mismatch Detector and as input pads for the test

of the Priority Resolver. PIN 24 and 26 have no connection. The way to operate the IAM by using these pads will be described as follow.

The first step is to set the control PIN 4 and 34 to enable two groups of control pads for CAM Array and Mismatch Detector only. Then we can input data using *bit* and $\overline{bit}$ of the CAM Array and load it to any word. After the data is loaded, we can search by triggering the $\overline{search}$, PIN 32, and with the search key on *bits* and $\overline{bits}$.

Figure 33. Layout of the *Pass control* cell.

Figure 34. Floor plan of the IAM chip.

Match line voltages can be measured from the four analog pads and the binary code of the mismatched number can be checked from the outputs of *l* and *m*, PIN 1 to 3 and PIN 36 to 40, of the Mismatch Detector. Now we can stop the search and switch the pads Pin 6 to 9 and PIN 11 to 14 into output pads to display the searched result of these eight words in the CAM array that are transferred and stored in *Hit finder* circuit of Priority Resolver. We will not switch the pads, PIN 1 to 3 and PIN 36 to 40, into input pads because we want to use the stored search result instead of using a direct input to the test pattern of *l* and *m* for the first four words to test the Priority Resolver directly. Now we can input any instruction code we want into pads PIN27, 28, 29 and 31 to test the function of Priority Resolver.

The hit results of these words can be checked at the pads Pin 6 to 9 and PIN 11 to 14. The Priority Resolver can also be tested independently by setting test codes on the

pads PIN27, 28, 29 and 31 when these pads, PIN 1 to PIN 3 and PIN 36 to PIN 40, are set up as input pads. The function of the *Pass control* circuit of the Priority Resolver can be checked by trigger pad PIN 33. Figure 35 is the picture of the whole chip.

SUMMARY

In this Chapter, the whole chip is designed at the transistor level. Most of these circuits are designed as simple cells by following the specification of Chapter II. CAM Array, Mismatch Detector and Priority Resolver are three main circuits of this chip. Layout of the whole chip makes these three main circuits to be tested by joining them together or by independently. This chip is fabricated by MOSIS with 2 μ N-well technology.

TABLE VI
I/O PORT ARRANGEMENT OF THE IAM CHIP

PIN	PAD ID	CAM Array	Mismatch Detector	Priority Resolver	Pads IO Switch
16	Analog 1		$match_0$		
17	Analog 2		$match_1$		
18	Analog 3		$match_2$		
19	Analog 4		$match_3$		
14	ioP_2-0	$\overline{bit}_0$		Hit_{out0}	
13	ioP_2-1	$\overline{bit}_0$		Hit_{out1}	
12	ioP_2-2	$\overline{bit}_1$		Hit_{out2}	
11	ioP_2-3	$\overline{bit}_1$		Hit_{out3}	
9	ioP_2-4	$\overline{bit}_2$		Hit_{out4}	
8	ioP_2-5	$\overline{bit}_2$		Hit_{out5}	
7	ioP_2-6	$\overline{bit}_3$		Hit_{out6}	
6	ioP_2-7	$\overline{bit}_3$		Hit_{out7}	
24	ioP_2-8				
23	ioP_2-9	$load_0$			
22	ioP_2-10	$load_1$			
21	ioP_2-11	$load_2$			
20	ioP_2-12	$load_3$			
36	ioP_2-17		m_{out3}	m_{in3}	
37	ioP_2-18		l_{out3}	l_{in3}	
38	ioP_2-19		m_{out2}	m_{in2}	
39	ioP_2-20		l_{out2}	l_{in2}	
40	ioP_2-21		m_{out1}	m_{in1}	
1	ioP_2-22		l_{out1}	l_{in1}	
2	ioP_2-23		m_{out0}	m_{in0}	
3	ioP_2-24		l_{out0}	l_{in0}	
4	ioP_2-25				ioP 2-0 to 2-7
26	ioP_2-26				
27	ioP_2-27			$\overline{l}$	
28	ioP_2-28			l	
29	ioP_2-29			$\overline{m}$	
31	ioP_2-30			m	
32	ioP_2-31		$search$		
33	ioP_2-32			All	
34	ioP_2-33				ioP 2-17 to 2-24
10	VddP				Gnd
30	GndP				Vdd

Figure 35. Photomicrograph.

CHAPTER IV

EXPERIMENT RESULTS

In this Chapter, we will describe the method use to test this chip. A test circuit has been designed and installed. The test will focus on the following points: the *match line* voltage, the invert voltage of the Mismatch Detector, the successful functioning of the CAM Array and Mismatch Detector, the function of the Priority Resolver and the working speed of this chip.

TEST CIRCUIT

A test circuit has been designed. A counter has been used in this test circuit for the speed test. This counter acts the input of the search key to the CAM. By adjusting the speed of this counter and measuring the output results, the working speed of this chip can be determined. The circuits for the test circuit are shown in Figure 36. This circuit also has three switch arrays to control its input, two LED arrays to display its output and four outputs from the analog pads. The data or the search key are input from the first switch array. This is input to the counter (74191), the inverter (7404) and the buffer (74245) to the CAM Array of the device under test. When we want to input specific data into the CAM Array, we stop the clock and load the counter with an input pattern. This way, each word of the device under test can be stored with a unique pattern array. We can then input a search key for a static test. If we want to test the speed, just enable the clock after the search key is input. The inverter generates the $\overline{bit}$ for the CAM array. Bypass capacitors were added to stabilize the local power. Some ICs also have a local capacitor. We also can check the input of the switch array using the LED display.

Figure 36. Block diagram of the IAM test circuit.

The second switch array is designed for the input of control codes and instruction codes. There are eight control codes and four instruction codes. These control codes are: four *Load* for the first four words in the CAM Array, the $\overline{Search}$ for the Mismatch Detector, the $\overline{All}$ for the *Pass all* of the Priority Resolver and two I/O controls to control two groups of pads as an input or an output pads. The inverter generates the $\overline{LSB}$ and $\overline{MSB}$ of the instruction code. The third input switch array sets the patterns of mismatched

number for testing the Priority Resolver. These input patterns can be displayed from the LED array. Figure 37 is the photograph of the IAM test circuit.

Figure 37. Photograph of the IAM test circuit.

To test this circuit, we input words into the CAM array by enabling input switches displayed on the LED array. Then we turned on the $\overline{\text{Search}}$ switch to enable a search. The voltage on the first four match lines can be measured at the analog pads. After the voltage on these match line is measured, we measure the invert voltage of the Mismatch Detector. For this measurement, a 'high' is given to the $\overline{\text{Search}}$ to turn off the pull up transistor on all *match lines*. We then store the search key into each word to force an exact match. An exact match cuts off the pull down transistors of each word on a match line. After these pull up and pull down transistors have been turned off, we put voltage on these match lines and check the second LED array to find out the invert voltage of the Mismatch Detector.

The way to display the mismatched result on the LED array is to turn off all the switches of the third switch array. If the output from the CAM chip is 'high', the LED

turns on. If the output is 'low', the LED will turn off. By checking the LED array, we can determine the invert voltage of the Mismatch Detector. After we obtain the match line voltage and the invert voltage, we can predict the match ability of the CAM Array and the Mismatch Detector. The Priority Resolver is easy to test because it is a digital logic circuit. By setting a test pattern of the mismatched number using the third switch array and put the instruction pattern of the mismatched number to the second switch array, the result of hit word will be seen from the second LED array. To switch the $\overline{All}$ control code, we can observe the output modes of Priority Resolver for the commands *Pass all* and *Pass one*.

MATCH LINE VOLTAGE

The first four words have their match lines connected to the analog pads. We will test the voltages of these four words by storing a fixed pattern in these words and measuring the voltage on the match line by input various search key. A fixed pattern '0000' or '1111' is stored in each word and the match line voltage is measured for every possible search key pattern. Table VII reports the result of this test. We next compare these voltages to the invert voltage of the Mismatch Detector to see if they can work together.

INVERT VOLTAGE OF THE MISMATCH DETECTOR

There are three inverters in the mismatch detector as described in Chapter III. This test is going to determine the invert voltage for each of these three inverters. Only the first four words in this CAM can be tested for this test because they are the only four words which have their match line connected to analog pads. The way to do this test is to store a word pattern in the four words and to match them by the same word pattern. This is an 'exact match', i.e. no pull downs on. We also input zero volt to the $\overline{Search}$ (Global in the CAM). This turns the pull up transistor off, so that the match lines voltage

cannot be set by the circuit but is controlled by the analog pad it connected to. An adjustable voltage is now put on the match line of a word and the voltage is increasing from zero to five volts. The output, number of bits mismatched for this word, will vary from three or more bits of mismatch to the exact match. We record the voltage between each two states as its invert voltage listed in Table VIII. Figure 38 shows the experimental result of the match line voltage and the invert voltage of the mismatch detector.

Figure 38. Experimental result of the match line and invert voltage.

Comparing the match line voltage of the CAM Array and the invert voltage of the Mismatch Detector in Figure 38, we can see that these two parts can work together. Table IX shows the tolerance between each working range of the match line voltage and the invert voltage.

TABLE VII
EXPERIMENTAL RESULTS OF THE MATCH
LINE VOLTAGE FOR 16 SEARCH KEYS

Key	Store 1111 before match				Store 0000 before match			
	Word1	Word2	Word3	Word4	Word1	Word2	Word3	Word4
0000	0.59	0.59	0.59	0.59	4.03	4.03	4.03	4.03
0001	0.82	0.82	0.84	0.83	3.17	3.16	3.16	3.14
0010	0.82	0.82	0.82	0.83	3.12	3.09	3.09	3.09
0011	1.46	1.49	1.47	1.48	1.58	1.59	1.59	1.60
0100	0.84	0.84	0.82	0.83	3.07	3.07	3.08	3.08
0101	1.48	1.49	1.48	1.48	1.56	1.54	1.53	1.53
0110	1.46	1.47	1.47	1.47	1.54	1.48	1.51	1.52
0111	3.05	3.05	3.05	3.04	0.87	0.87	0.87	0.87
1000	0.82	0.83	0.83	0.83	3.11	3.12	3.11	3.12
1001	1.49	1.49	1.49	1.49	1.53	1.54	1.54	1.55
1010	1.47	1.47	1.48	1.46	1.54	1.57	1.55	1.56
1011	3.06	3.07	3.08	3.08	0.87	0.87	0.87	0.87
1100	1.48	1.49	1.49	1.48	1.47	1.49	1.49	1.49
1101	3.09	3.09	3.09	3.09	0.85	0.85	0.85	0.85
1110	3.05	3.06	3.06	3.06	0.84	0.84	0.84	0.84
1111	4.00	4.00	4.00	4.00	0.60	0.60	0.60	0.60

FUNCTION OF THE PRIORITY RESOLVER

The input of the priority resolver is two bits of decoded data from each word from Mismatch Detector representing zero, one, two or three bits of mismatch. The Priority Resolver finds the best match from all these words using the *Best finder* circuit. If we want to find the best matched word/words from the CAM, we just feedback the best matched two bit code from the *Best finder* circuit to the *Hit finder* circuit of the Priority Resolver to find all of words with this number of mismatched bits. We can also find the word/words with a specific number of mismatch by using the Priority Resolver instruction set. After the matched words are found, Priority Resolver is instructed to pass all of

these words or just the one (from top to bottom) to its output.

TABLE VIII
EXPERIMENTAL RESULTS OF THE INVERT
VOLTAGE OF THE MISMATCH DETECTOR

Invert voltage between	3 and 2 mismatch	2 and 1 mismatch	1 and 0 mismatch
Word 1	1.29v	2.59v	3.72v
Word 2	1.26v	2.60v	3.77v
Word 3	1.29v	2.59v	3.74v
Word 4	1.28v	2.57v	3.70v

TABLE IX
EXPERIMENTAL RESULTS OF THE
NOISE MARGIN OF THE FLASH A/D

Exact match	4.00 to 4.03 volt
Tolerance	0.23v
Invert voltage between 0 and 1 mismatch	3.70 to 3.77 volt
Tolerance	0.54v
One bit mismatch	3.05 to 3.16 volt
Tolerance	0.45v
Invert voltage between 1 and 2 mismatch	2.57 to 2.60 volt
Tolerance	0.98v
Two bits mismatch	1.47 to 1.59 volt
Tolerance	0.18v
Invert voltage between 2 and 3 mismatch	1.26 to 1.29 volt
Tolerance	0.39v
Three bits mismatch	0.59 to 0.87 volt

The testing method for the *Hit finder* begins by generating all possible search results. And by checking these against *Hit* results, we can determine that this circuit works normally. Table X shows the test result. The test result shows the *Hit finder* circuit of the Priority Resolver is working as we predict. Also the function of the *Hit finder* circuit is seen to be normal.

TABLE X
EXPERIMENTAL RESULTS OF THE HIT FINDER CIRCUIT

Input number		The requested number from instruction			
Word	Input	00	01	10	11
0	00	Hit			
1	01		Hit		
2	10			Hit	
3	11				Hit
0	01		Hit		
1	10			Hit	
2	11				Hit
3	00	Hit			
0	10			Hit	
1	11				Hit
2	00	Hit			
3	01		Hit		
0	11				Hit
1	00	Hit			
2	01		Hit		
3	10			Hit	

The *Pass control* circuit is the next circuit to be tested. By arranging the input number of mismatched bits, all of the possible *Hit* patterns of these four words can be generated. To control the $\overline{All}$ switch, we will see that only the first word to be hit will show up when the *Pass one* is chosen. Table XI shows the *Pass control* circuit works normally.

SPEED

The speed test is focused on the voltage on the match line because of the charge and discharge time of the match line signal. We store '0000' in CAM and use the counter to input the search key. This periodic search key makes the voltage on the match

line sequence through 0 to 3 bits of mismatch. By measuring the voltage on the oscilloscope, we can see if the match line voltage falls within the invert voltage of the Mismatch Detector. Figure 39 shows four voltage levels on the match line. Each of these level from top to bottom, represent the match line voltage of 0, 1, 2, and 3 bits of mismatch. Figure 40 is the output of the MSB and LSB from the Mismatch Detector versus to the match line voltage. Figure 41 is the waveform of the match line voltage, MSB and LSB from simulation. The experimental values of the match line voltage, MSB and LSB of Figure 39 and 40 compare favorably to the simulation. Figure 42 and 43 shows the match line, MSB and LSB voltages with a 2 MHz search pattern. Figure 44 shows the result at 3 MHz. The LSB and MSB work correctly, the Mismatch Detector is decoding the number of bits mismatched from the match line. However, the match line voltage is ugly. The capacitance of the analog pad and probe load the match line voltage and limits the performance. These capacitors combined with the match line pull up pMOS transistor form a low pass filter. Because the small size pull up transistor is a large resistor (a typical MOS transistor is on the order of 20 k Ω), even a small capacitance limits high frequency performance since the time to charge the match line will increased. Normally the match line is internal to the circuit which is not connected to pad. Without the capacitance of analog pad and probe, the match line is simulated to run in excess at 6 MHz.

A voltage follower (NTE 924M) is added on one of these four match lines. It makes the shape of the match line voltage looks better but the pad capacitance still limits performance.

A simulation result, see Figure 45, shows that a 7pF connected to the match line as the totally capacitance of the pad, wire and probe, makes the match line voltage as ugly as we see in Figure 44. This result is consistent with the parallel capacitors of the probe, cable and pad.

TABLE XI
EXPERIMENTAL RESULTS OF THE PASS CONTROL CIRCUIT

[illegible]

Figure 39. Measured Voltage level on the match line.

Figure 40. Measured Voltage level of MSB(Top) and LSB(Bottom).

Store 0000 and searched by 16 patterns.		1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
		1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
		1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
		1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Number of mismatch		4	3	3	2	3	2	2	1	3	2	2	1	2	1	1	0
Output of the MSB and LSB	LSB	1	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0
	MSB	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0
Voltage on match line by each number of bits mismatched	0																
	1																
	2																
	3, 4																
Output of the LSB	High																
	Low																
Output of the MSB	High																
	Low																

Figure 41. Predict multivalue logic binary output levels of match line.

Figure 42. Measured voltage on a match line operating at 2 MHz.

Figure 43. Measured voltage of MSB(Top) and LSB(Bottom) operating at 2 MHz.

Figure 44. Measured voltage on a match line, MSB and LSB (Top-Down) at 3 MHz.

Figure 45. The match line voltage simulated at 3 MHz and connecting 7pF.

CURRENT

Experimental results show the totally current dissipation of this CAM, pads not included, is $965\mu\text{A}$. The current dissipation of each word is $87.07\mu\text{A}$ in average by simulation in Chapter III. There are eight words in this chip. Totally current dissipation will be $696.56\mu\text{A}$ by simulation. The analog to digital converter dissipates most of current in this AM circuit because it discharges the match line when doing a search command.

SUMMARY

The test result in this Chapter shows that the Mismatch Detector can work together with the CAM Array. The voltage tolerance between the match line voltage and the invert voltage of the Mismatch Detector is enough to function at 3 MHz. The voltage tolerance is 0.18 volt between the two bits mismatch and the invert voltage between 2 and 3 bits mismatch. The Mismatch Detector can easily work within this tolerance. If we

properly arrange the the voltage level on the match line, more than three bits could be identified. More than three bits to be identified may be useful when we communicate by the Golay code as described in Chapter I.

The working speed of this chip is 6 MHz in simulation. The picture shows it works on 3 MHz normally even under the influence of the capacitance of pad and probe. The capacitance on the connection wire of the board, the analog pad and the test probe greatly reduce the speed of the charge and discharge of the mismatch line since it is a low pass filter. This chip could be designed to work faster than it is now by reduce the voltage level on the match line. The voltage on the match line is now between 0.59 volt (three bits mismatch) and 4.03 volt (exact match). The charge and discharge of the match line is between these two voltages. If we adjust this voltage difference to 1.5 volt, the charge time could be reduced. Which means that the exact match will be 4 volts and the three bits mismatch will be 2.5 volts. So the one bit mismatch will be 3.5 volt and the two bits mismatch will be 3 volt. There is 0.5 volt between them. If the invert voltage is set between them, the tolerance will be 0.25 volt. From the experimental results, the tolerance can be 0.18 volt. So this arrangement should work and is a way to improve its working speed from design.

One of the three parts, *Best finder*, of the priority resolver has no connection to pads. This does not influence the test of the other two parts because the *Best finder* manages its own I/O.

CHAPTER V

CONCLUSION AND FUTURE WORK

A new CAM is developed in this thesis. This CAM is capable of the word/bit parallel search instead of the word parallel/bit serial search by a regular CAM. This CAM can also identify the words with one, two or three bits mismatch to the search key. It can also find the words which are best matched to the search key. This function is good for database retrieval, communications and error correction circuitry. The ability to find similar words to the search key makes this CAM useful for sorting, set operations and speech recognition. The inexact CAM works without a masking loop because this CAM is a mixed signal circuit capable at word/bit parallel search instead of the word parallel/bit serial search by a regular CAM. Similar words, identified by the search key, can be treated as the data or as a tag for a group of data. The error-correction code is an example of treating the similar words as data.

The function of word/bit parallel searching is made possible by using the mixed analog/digital design. The information of the number of bits mismatched is passed out on one wire for each word as an analog voltage. The voltage on this wire encodes the number of bits mismatched. This design helps to reduce the number of wires into one because the digital information of many wires is replaced by the analog signal on one wire. The saved area can be used for putting more words in the chip. The analog signal is converted to a digital signal by A/D converter and then compared to the search key. A *Best finder* circuit identifies the least number of bits mismatched. If we use this number as the search key, we can get all of the best matched words. A specific number can also be used as a search key to find out the words which have a specific number of bits

mismatched to the search key. The number of words found could be more than one. These words would likely have different contents because the same number of bits mismatched to the search key does not mean they have the same contents. A *Pass control* circuit helps to pass one or all of the hit information for these words. If we use the *Pass control* circuit to pass the hit information one by one, we can reset the tag of each word to make this circuit pass the next one.

Chapter II and Chapter III described the design of this CAM. There are three parts in this CAM. The CAM array is the first part of this CAM. The analog to digital converter is the second part of the CAM. There are eight analog to digital converters in the CAM, one for each word. The Priority resolver is the last part of the CAM circuit. It contains the *Hit finder*, the *Best finder* and the *Pass control* circuits. All of these circuits are simulated and assembled to become the inexact CAM.

A test circuit was prepared for testing the chip. The test result in Chapter VI shows the function of this chip is correct. The working speed of this CAM is 6 MHz in simulation. It is 3 MHz in experiment result. The capacitance of the analog pad and the test probe reduced the speed because the size of the pull up transistor of the match line is very small (3/5).

The number of bits mismatched is limited because the voltage range on the match line is 0 to 5 volt. The voltage level is limited to this range since we wanted to have a reasonable tolerance between each voltage level. Besides, the size of the pull up transistor of the match line is controlled to limit the power dissipation. This transistor will saturate if too many pull down transistor are ON at the same time in a word. The pull down transistor is working when the bit it is connected to is mismatched to the bit of the search key. So the number of bits mismatched is limited by the saturation of the pull up transistor.

This CAM is limited to its number of bits mismatched by the voltage controlled

design. If we design a CAM with a current controlled design [8], the current range may detect larger values of mismatched bits than the voltage controlled one.

REFERENCES

- [1] W. R. Daasch, "Inexact Match Associative Memory Cell," *Electronics Letters*, Vol. 27(18), pp. 1623-1625, August 1991.
- [2] Lawrence T. Clark, Robert O. Grondin, "A Pipelined Associative Memory Implemented in VLSI," *IEEE Journal of Solid-State Circuits*, Vol. 24(1), pp. 28-34, 1989.
- [3] Masaki Hirata, Hachiro Yamada, Hajime Nagai, Kousuke Takahashi, "A Versatile Data String-Search VLSI," *IEEE Journal of Solid-State Circuits*, Vol. 23(2), pp. 329-335, April 1988.
- [4] Luis Sergio Kida, "Associative Processing Implemented with Content-Addressable Memories," *Master Thesis, Master of Science in Electrical and Computer Engineering, Portland State University*, pp. 32-38, 1991.
- [5] Raymond Hill, "A First Course in Coding Theory," *Oxford Applied Mathematics and Computing Science Series*, pp. 1-96, 1988.
- [6] Vera Pless, "Introduction to The Theory of Error-Correcting Codes," *Wiley-Interscience Series in Discrete Mathematics*, pp. 1-31, 1982.
- [7] Lance A. Glasser, Daniel W. Dobberpuhl, "The Design and Analysis for VLSI Circuits," *Addison-Wesley Publishing Company*, pp. 90, 1988.
- [8] Sateh M. S. Jalaieddine, Louis G. Johnson, "Associative IC Memories with Relational Search and Nearest-Match Capabilities," *IEEE Journal of Solid-State Circuits*, Vol.27(6), pp. 892-900, June 1992.

APPENDIX

MOSIS N14Q SPICE MODEL

N14Q SPICE LEVEL 2 PARAMETERS

```
.MODEL CMOSN NMOS LEVEL=2 LD=0.250000U TOX=401.000008E-10
+ NSUB=6.403604E+15 VTO=0.74514 KP=4.771000E-05 GAMMA=0.5354
+ PHI=0.6 UO=554.714 UEXP=0.125874 UCRIT=58894.7
+ DELTA=0.393586 VMAX=58871.8 XJ=0.250000U LAMBDA=3.428428E-02
+ NFS=5.835836E+12 NEFF=1 NSS=1.000000E+10 TPG=1.000000
+ RSH=30.800000 CGDO=3.229249E-10 CGSO=3.229249E-10 CGBO=4.560243E-10
+ CJ=1.115100E-04 MJ=0.707810 CJSW=5.257200E-10 MJSW=0.256976 PB=0.8000
00
```

* Weff = Wdrawn - Delta_W

* The suggested Delta_W is -0.68 um

```
.MODEL CMOSPMOS PMOS LEVEL=2 LD=0.234471U TOX=401.000008E-10
+ NSUB=6.390800E+15 VTO=-0.799827 KP=2.368000E-05 GAMMA=0.5349
+ PHI=0.6 UO=275.495 UEXP=0.250488 UCRIT=19070.8
+ DELTA=1.59261 VMAX=35135.1 XJ=0.250000U LAMBDA=5.599540E-02
+ NFS=3.235210E+11 NEFF=1.001 NSS=1.000000E+10 TPG=-1.000000
+ RSH=96.240000 CGDO=3.028661E-10 CGSO=3.028661E-10 CGBO=3.836660E-10
+ CJ=2.510500E-04 MJ=0.545355 CJSW=3.047600E-10 MJSW=0.327927 PB=0.8000
00
```

* Weff = Wdrawn - Delta_W

* The suggested Delta_W is -0.23 um

Model Parameters		Units
LD	lateral diffusion (length)	meter
TOX	oxide thickness	meter
NSUB	substrate doping density	$1/\text{cm}^3$
VTO	zero-bias threshold voltage	volt
KP	transconductance	amp/volt^2
GAMMA	bulk threshold parameter	$\text{volt}^{0.5}$
PHI	surface potential	volt
UO	surface mobility	$\text{cm}^2 / \text{volt sec}$
UEXP	mobility degradation exponent	
UCRIT	mobility degradation critical field	volt/cm
DELTA	width effect on threshold	
VMAX	maximum drift velocity	meter/sec
XJ	metallurgical junction depth	meter
LAMBDA	channel-length modulation	volt^{-1}
NFS	fast surface state density	$1/\text{cm}^2$
NEFF	channel charge coefficient	
NSS	surface state density	$1/\text{cm}^2$
TPG	gate material type: +1= opposite of substrate -1= same as substrate 0 = aluminum	
RSH	drain, source diffusion sheet resistance	ohm/square
CGDO	gate-drain overlap capacitance/channel with	farad/meter
CGSO	gate-source overlap capacitance/channel with	farad/meter
CGBO	gate-bulk overlap capacitance/channel with	farad/meter
CJ	bulk p-n zero-bias bottom capacitance/area	$\text{farad}/\text{meter}^2$
MJ	bulk p-n bottom grading coefficient	
CJSW	bulk p-n zero-bias perimeter capacitance/length	farad/meter
MJSW	bulk p-n sidewall grading coefficient	
PB	bulk ohmic resistance	ohm