Methods and Technologies for Functional Memories with Content Addressability, Optimized Bandwidth and Scalability

Kazunari Inoue

CONTENTS

List of Figures and Tables
Acknowledgements
Abbreviations and Glossary
Abstract

Chapters

1. Introduction 11
   1.1 History and technical trend of scalable semiconductor memories 11
   1.2 Objectives for this work's functional memories with respect to application 13
   1.3 Structure of the thesis 13
References

2. Classification of memories and their bandwidth capability 17
   2.1 Introduction 17
   2.2 Overview of semiconductor memories with respective features and benefits 17
   2.3 Bandwidth requirements for frame-buffer displaying in graphics applications 18
   2.4 Bandwidth requirements for the packet forwarding in network applications 20
   2.5 Content addressability viewed as a value-added high-bandwidth solution 21
References

3. Bandwidth optimization in memories for graphics applications 25
   3.1 Introduction 25
   3.2 Technologies carried out by the integration on memory 27
      3.2.1 Z-compare and A-blend units on 3D frame-buffer memory 27
      3.2.2 Pre-add multiplier for cost reduction of integration 30
   3.3 Technologies related to the memory-internal data bus 37
      3.3.1 Low voltage read-modify-write memory-internal data bus 38
      3.3.2 Duplicated page function with the utilization of the sense-amplifier 41
References

4 Content addressability as further enhanced effective bandwidth capability 47
4.1 Introduction 47
4.2 Physical structures performing binary CAM and ternary CAM functions 47
4.2.1 Conventional CAM, binary CAM (BCAM) 48
4.2.2 Ternary CAM (TCAM) 49
4.3 Search operation and the application of signature-matching 51
4.3.1 Conventional search operation with tree algorithm 53
4.3.2 CAM based search with its advantages and problems 56
References

5 Experimental study of content-addressability integration onto memory 61
5.1 Introduction to the previous state of the art 61
5.2 Hierarchical pipelined partial search 61
5.3 Comparison of DRAM based CAM cell and SRAM based CAM cell 65
5.4 Unsolved serious problems related to the defect rate 67
References

6 Newly developed power reduction technologies for CAM 70
6.1 Introduction to the previous state of the art 70
6.2 Improved hierarchical pipelined partial search 71
6.3 2-bit encoded storage and for power-reduction of search-line data generation 73
6.4 Dynamically configurable flexible partitioning 75
References

7 Newly developed CAM-defect repairing technologies enabling cost-per-bit scalability 78
7.1 Introduction to the previous state of the art 78
7.2 Negative binominal model and defect analysis 79
7.3 Issue of priority resolving and its effect on the repairing technology 81
7.4 Developed redundancy-based repairing technology 82
References

8 Fabricated examples of CAM-based application specific VLSI circuits 86
8.1 Introduction 86
8.2 High-density 18M-bit full ternary CAM VLSI 86

8.2.1 Dynamically-configurable flexible partitioning 87
8.2.2 Intelligent aging function for CAM entries 88
8.2.3 Remaining issue of power-supply noise due to large *di/dt* 91
8.3 Signature-matching co-processor VLSI 97
8.3.1 Search-request data for thoroughly byte-shifted payload 98
8.3.2 Secondary lookup table with programmable logic operations 98
References

9 Conclusion 104

Published Paper List 107
Referenced Paper List 107

List of Figures and Tables

Figures

Figure 1.1: Memory and Scaling
Figure 1.2: Structure of this Thesis
Figure 2.1: Trend of memory bandwidth
Figure 2.2: Search with read operation
Figure 2.3: Search with content addressability function
Figure 3.1: Rendering with Flat shading
Figure 3.2: Rendering with Smooth shading
Figure 3.3: Block diagram of experimental 3D frame-buffer
Figure 3.4: Experimental frame-buffer for Z and for color
Figure 3.5: Conventional carry-save algorithm
Figure 3.6: Proposed pre-add algorithm
Figure 3.7: Example of multi-sample filtering for anti-aliasing
Figure 3.8: Various multi-sample filtering and equations
Figure 3.9: Weighted interpolation with proposed pre-add MPY
Figure 3.10: Memory architecture and bandwidth capacity
Figure 3.11: Worst case benchmark
Figure 3.12: Conventional differential I/O bus
Figure 3.13: Read-modify-write bus in this work
Figure 3.14: Waveform of read-modify-write bus
Figure 3.15: Benchmark with read-modify-write bus
Figure 3.16: Duplicate page (DUP) scheme
Figure 3.17: Die photo of experimental frame-buffer with Z-compare and A-blend units
Figure 3.18: Die photo of fabricated frame-buffer with Pre-Add MPY in 0.25μm CMOS
Figure 4.1: CAM cell and array
Figure 4.2: Hierarchically established rules for TCAM
Figure 4.3: Ternary CAM cell and array
Figure 4.4: Tree algorithm for 16-character search
Figure 4.5: Conventional pipelined signature search based on tree algorithm
Figure 4.6: Concept of CAM based signature search
Figure 5.1: Hierarchical pipelined search concept

Figure 5.2: Example of ML control in a conventional pipelined search
Figure 5.3: Time charts of a conventional pipeline for (a) the miss case and (b) the match case
Figure 5.4: Relation between power dissipation and number of multiple matches
Figure 5.5: DRAM based CAM
Figure 5.6: Proposed transparent schedule to hide refresh
Figure 5.7: Test yield difference between SRAM and CAM
Figure 6.1: Proposed circuitry for pipelined search with power reduction
Figure 6.2: Time chart of the proposed improved pipelined search
Figure 6.3: Conventional scheme of SL generation
Figure 6.4: Proposed 2-bit encoding for reduced SL power dissipation
Figure 6.5: Multiple purposes of table lookup
Figure 7.1: Defect analysis method
Figure 7.2: Defect rate vs. particle size
Figure 7.3: Probability vs. particle size
Figure 7.4: True defect particle probability vs. particle size
Figure 7.5: Proposed SW/HW combined redundancy
Figure 8.1: Proposed flexible partitioning with extended DX bit
Figure 8.2: Hierarchical search scheme with DX
Figure 8.3: Extended function of empty/occupied indicator and aging
Figure 8.4: Proposed aging scheme in 18M-bit CAM
Figure 8.5: Example of I-R drop simulation
Figure 8.6: VDD waveform with 4.5Mb CAM array
Figure 8.7: Schmoo plot with 4.5Mb CAM array
Figure 8.8: VDD waveform with 9Mb CAM array
Figure 8.9: Schmoo plot with 9Mb CAM array
Figure 8.10: VDD waveform with 18Mb CAM array
Figure 8.11: Schmoo plot with 18Mb CAM array
Figure 8.12: VLSI and PKG modeling
Figure 8.13: Effect of on-package capacitor
Figure 8.14: Experimental 18M-bit TCAM VLSI
Figure 8.15: Block diagram of signature-matching co-processor
Figure 8.16: Die photo of fabricated signature-matching co-processor in 130nm low-leakage CMOS

Figure 8.17: Measured power dissipation of fabricated signature-matching co-processor
Figure 8.18: Analysis of relative contributions of the different power saving proposals

Tables

Table 3.1: Summary of experimental frame-buffer with 0.5μm CMOS technology
Table 3.2: Summary of experimental frame-buffer with 0.25μm CMOS technology
Table 4.1: Examples of identification data for useless packets
Table 6.1: Technology trend and the scaling of power
Table 6.2: Encoding for data storage and corresponding truth table
Table 7.1: Yield prediction with proposed redundancy
Table 8.1: Summary of features for 18M-bit ternary CAM
Table 8.2: Summary of features for the signature-matching co-processor

List of Abbreviations and Glossary

Abbreviation		Glossary
(V)LSI	(Very) Large Scale Integration	Integrated circuitries realized (typically) on silicon
SoC	System on Chip	Various application-oriented specific LSIs, which typically integrate user's logic, memory, I/O, and glue logic on same die area.
I/O	Input and Output	Input pin and output pin of (V)LSI
SDR	Single Data Rate	One of standard I/O specifications. Single event per clock cycle.
DDR	Double Data Rate	One of standard I/O specifications. Two events per clock cycle, hence it doubles the I/O performance in comparison with SDR.
QDR	Quad Data Rate	One of standard I/O specifications. Four events per clock cycle, hence it doubles the I/O performance in comparison with DDR. However common QDR is provided by separated I/O functions, in other words, DDR Input and DDR Output are combined together.
DRAM	Dynamic Random Access Memory	Typical DRAM cell consists of 1-transistor and 1-capacitor, and cell data is stored onto the capacitor as dynamic electrical charge. Hence, DRAM takes advantage of small cell area, while the necessity of refresh to re-store the stored data remains.
SRAM	Static Random Access Memory	Typical SRAM cell consists of 6-transitors as static latch. Hence SRAM takes advantage of high-speed operation without refresh cycle, while the cost of number of transistor is higher.
CAM	Content Addressable Memory	Each memory cell performs comparing operation with given data, hence CAM is capable of a fast search operation in addition to write, store, and read.
BCAM	Binary CAM	CAM cell which can realize binary states "0" and "1".
TCAM	Ternary CAM	CAM cell which can realize ternary states "0", "1", and "x". Typical TCAM cells consist of two binary cells, each of which can represent 2 states; hence it can actually represent four states.

Abbreviation		Glossary
BL	Bit Line	Data line for realizing the write/read operation to/from a memory cell.
WL	Word Line	Address line to select memory cells for the purpose of read or write operations.
SL	Search Line	Search request data line for data provided by via the input pins, which is applied for a search operation in the CAM.
ML	Match Line	Search result indicator line to know whether the match-comparing result is hit or miss for the input data, which is applied for a search operation in the CAM.
Src.	Source	Typically it means the old data, which is now visible on the screen in the application of computer graphics.
Dst.	Destination	Typically it means the new data, which is writing from now-on to the screen in the application of computer graphics.
XGA	Extended Graphics Array	Standardized screen size typically consisting of 1,024× 768 resolution
SXGA	Super XGA	Standardized screen size typically consisting of 1,280×1,024 resolution
UXGA	Ultra XGA	Standardized screen size typically consisting of 1,600×1,200 resolution
ALU	Arithmetic Logic Unit	Hardware unit, which performs mathematical operations such as add, sub in an LSI.
MPY	Multiplier	A multiplier typically comprises a plurality of adders in hardware.
MUX	Multiplexer	A selector for one out of multiple inputs such as one out-of 2 or one out-of 4.
Ethernet	originated from "ether"+"network"	Typical LAN connection standardized by IEEE 803.2
MAC	Media Access Control	This is involved in the packet header in Layer 2.
IP	Internet Protocol	This is involved in the packet header in Layer 3.
TCP	Transmission Control Protocol	This is involved in the packet header in Layer 4.
UDP	User Datagram Protocol	This is involved in the packet header in Layer 4.
DoS	Denial of Service	Most commonly used in connection with the DoS attack, resulting from various misused packet in the network.
QoS	Quality of Service	Service for making the network more reliable, for example by prioritizing the packet forwarding.

ACKNOWLEDGEMENTS

I would like to thank to Professor Hans Jürgen Mattausch and Professor Tetsushi Koide at Research Center for Nano-devices and Systems, Hiroshima University, for their detailed comments, suggestions, and constant support. This thesis summarizes the main results of researches from 1995 to 2003 in system LSI department, Mitsubishi Electric Corporation, and from 2003 to 2005 in Research Center for Nano-device and Systems, Hiroshima University and in System Solution department, Custom LSI design division, Renesas Technology Corporation. I am grateful to Dr. Kazutami Arimoto, Dr. Tadato Yamagata, Dr. Kazuyasu Fujishima, Dr. Michihiro Yamada, Hideyuki Noda, Hideaki Abe, Hideto Matsuoka, Kaori Hayashi, Shuji Fukagawa and Yuji Yano at Renesas Technology Corporation for giving me valuable advices. I thank Toshiyuki Ogawa, Masatomi Okabe, and Tsunesato Munakata at Renesas Technology Corporation, and also would like to thank Robin Chan, Masahiro Suzuki, Yoichi Goi and Hideo Kameda at Renesas Technology America for giving me the opportunity of carrying out the reported research. I also would like to thank Mike Lavelle, David Kehlet, Michael Deering, Olivet Chou, Ewa Kuwalsuka, and Yan Tang for their technical advices and comments in the graphics system application field during the co-development between Mitsubishi Electric Corporation and SUN Micro Systems Corporation. Furthermore, I would like to thank Michael Regal, Fred Schindler, Greg Dejager, Rob Listen, JR River, Jaushin Lee Ph.D, Sanjay Desai, Ron Poon, Mani Raposa, and Kevin Morishige at CISCO Systems Inc. for their technical comments and advices in the network system application field.

Abstract

Methods and technologies for high-bandwidth functional memories especially with a content addressability are addressed. While great many applications can benefit from the availability of high-bandwidth carried out by embedded SRAM (eSRAM) and embedded DRAM (eDRAM), approaches based on the content addressability such as embedded CAM were much less seen due to lack of scalability described as follows.

- Large bandwidth capacity provided by content addressability comes with tremendous cost of power. According to the quantitative scaling analysis, power keeps increasing with factor of x1.6 as high as that of the previous generation in conventional CAM architecture.

- Content addressability carried out by every memory cell functioning simultaneously results in a physical cell layout of complicated structure. It unfortunately affects the defect rate seriously, despite the benefit of bandwidth. The cost of a bit-cell is the second concern in the content addressable memory. In actual experiment, examined defect rate in CAM was almost 2x as high as that of SRAM, even though the minimization technology is further validated.

The importance of reliable scaling cannot be overemphasized. Promotion of CAM based high-bandwidth functional memories dedicating specific applications as one of major SoC will never be successful unless these scaling problems completely disappear. In this thesis, I set out to conduct various experiments and analyses to establish the scalability.

1. An improved hierarchical partial search on the basis of the conventionally proposed pipelined architecture successfully shrinks the active area while maintaining the benefit of high-bandwidth. Measured power is 29% as low as for the conventional model, which will certainly eliminate the negative scaling with the factor of x1.6.

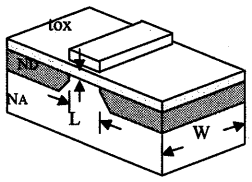
2. Proposed redundancy by software and hardware approaches are combined together effectively and allow to repair the defects. The cost of a bit-cell can be reduced by an amount of 22% as low, which also eliminates the experienced factor of x2.0 as high.

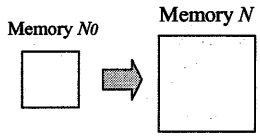
Given scalability drives the development of CAM based high-bandwidth functional memory with the evidence, for example, of a signature-matching co-processor in 130nm CMOS technology for the network security field.

Chapter 1
Introduction

1.1 History and technical trend of scalable semiconductor memories

Traditionally, semiconductor memories have been evaluated by their storage capacity, which has followed an exponential curve, referred to as Moore's Law (proposed in 1965). [1,2] According to this growth, "each new generation of memory integrated circuits contained roughly twice as much storage capacity as its predecessor, and each generation was released just within 18-24 months after the previous generation." Moore's prediction is deeply concerned with "scaling". A technology based on reliable scaling safely keeps evolving the memory towards higher density, lower cost, higher speed and lower power consumption. The key for driving this technical trend of continuous memory improvement are various rules for scaling factor of the devices structures on the silicon surface and the operating voltages [3]. To keep the power dissipation constant during scaling is often an important boundary condition. For example, even if memory density increases two times, as the area can be shrunk with the scaling factor of 1/2, reduction of the core voltage with the scaling factor of 0.7 (sq. 0.7 = ~0.5) keeps the power dissipation per area unit constant, as illustrated in Figure 1.1.





Channel length	$L = L0/S$
Channel width	$W = W0/S$
Device area	$A = A0/S^2$
Device voltage	$V = V0/S$
Density	$N = N0 \cdot 10^{0.15year}$
Memory LSI area	$A = A0$

When Scaling Factor S=0.7

Memory density	$N=2 \cdot N0$
Memory LSI cost	*Cost=Cost0 (Const.)*
Memory LSI power	$P=P0$ *(Const.)*

Figure 1.1 Memory and Scaling

The importance of scaling cannot be overemphasized. Without safe scaling, the exponential growth in integrated memory capacity is hardly possible. It can be named in this thesis that a reliable growth driven by various technological scaling factors, "scalability".

Taken in the light of recent semiconductor technologies, it is an open question whether such exponential growth continuous to be valid, since many technological difficulties are arising [4]. For example, it is reported that the actual minimization performed on silicon has not been by a factor 1/2, in spite of a 2 times higher density. Hence it is necessary for semiconductor technology to explore other cost reduction factors in terms of silicon area and power dissipation.

Alternatively, a more important present problem is that the storage capacity alone does not fairly evaluate and qualify today's semiconductor memories, especially of those memories used for many newly appearing applications. It is well known that the access bandwidth of a memory, which refers to the data transferring capacity, is another important quality factor of the memory. Although certain growth is recognized in the bandwidth, the bandwidth growth is not as fast as the density growth. As a result, access-bandwidth limitations often restrict the performance of applications. Since the middle of the 1980s, specific high bandwidth memories were proposed such as VRAM [5] and volume productions have started in the graphics-application field. Nowadays, these application specific high-bandwidth memory VLSIs are nothing special in the graphics field seen in PC-based 3D graphics [6]. As an additional example, memory with content addressability has been highly evaluated in the field of networking application since the beginning of the new century, though application specific memories based on the content addressability are not popular yet. Graphics as well as networking applications are suffering tremendously from the lack of memory bandwidth, and that is the reason why they have escaped or are just in the process of escaping from the solution based on commodity memory. Unfortunately, this high bandwidth approach with application specific memory is typically more expensive in terms of cost per bit and cost of power dissipation. It might become a serious concern if such problems keep growing with the technology generation changes. Therefore, in the reported research work, I have paid great attention to scaling in the various experiments and formal analysis.

1.2 Objectives for this work's functional memories with respect to application

The objective of this work attempts firstly to examine the true bandwidth desired by applications, and secondly to develop the technology for optimizing the bandwidth. The subject of high-bandwidth semiconductor memories is already a well-researched area, however very few attempts have kept a close relation with the actual applications. To clearly see this application relation, the bandwidth in this work is quantified and qualified in the applications measuring units, namely in pixels-per-second (-pps) and packets-per-second (-pps) for the graphics-application field and for the networking-application field, respectively. These units can replace the commonly used unit of bits-per-second (-bps).

There are two major approaches to high bandwidth technology used in the reported research. One is the functionality integration onto the memory and the other is the utilization of internal busses within the memory. I particularly focus my attention on the content addressability as a new candidate for high-bandwidth and intelligent solutions. A given bandwidth can be maximized by efficient utilization in the memory cells, and particularly beneficial is in the content addressability. Part of the reported results and investigations aimed at providing a reliable scalability for the field of memories with content addressability. I actually experienced two serious problems in an experimental study of memories with content addressability, namely huge power dissipation and low-yield related high cost per bit. The two factors unfortunately indicated that reliable scaling is not possible for conventional content-addressability concepts of memories. Therefore, in order to successfully establish a reliable scalability for content-addressable memories, a number of experiments were conducted and various technical solutions were proposed in the course of the reported research work. The verified results provide a new opportunity for the development of high bandwidth content addressable VLSI dedicating the specific applications.

1.3 Structure of the thesis

Structure of the thesis is shown in Figure 1.2. Chapter 2 reviews the technological fundamentals of the memory hardware and presents a discussion with respect to the applications' requirements. I noticed that the memory bandwidth required, does not necessarily result in high bandwidth at the external data pins of the chip. Rather, the

integration of specific components onto the memory can effectively reduce the external bandwidth demand and also result in a gain of quality. The effectiveness of functionality integration is verified by an experimental frame-buffer VLSI design for graphics applications reported in Chapter 3. Chapter 4 reviews the field of content addressability, which is a certain candidate providing further improvement towards intelligent high bandwidth memories. Note that conventional content addressable memory (CAM) cannot state high bandwidth without a tremendous cost in power dissipation. Therefore, Chapter 5 describes an experimental study for the purpose of CAM power-dissipation evaluation as a pre-study in the development of application specific VLSIs based on CAM. An analysis of power dissipation during execution of the content addressable functionality and technologies enabling reduced power dissipation are described. Incidentally, this pre-study conducted with fabricated silicon revealed an unexpected problem of high defect rate. Apparently, seriously increased defect rate is caused by the complicated physical cell structure necessary to integrate the content-addressability function into each bit-storing cell. The consequence of degraded scalability, resulting from both power dissipation and defect rate problems, is also discussed in Chapter 5. A further power reduction technology based on the evaluation of the experimental study is proposed and described in Chapter 6. I then examined the defect versus particle probability, verified the analysis methodology, and developed a unique repairing technology for CAM in Chapter 7. With the results of Chapter 6 and 7 I am able to establish a new method for improved CAM scalability. Chapter 8 presents experimental CAM-based application specific VLSI circuits for the networking field based on the developed power-reduction and repairing technologies, and finally Chapter 9 concludes this thesis.

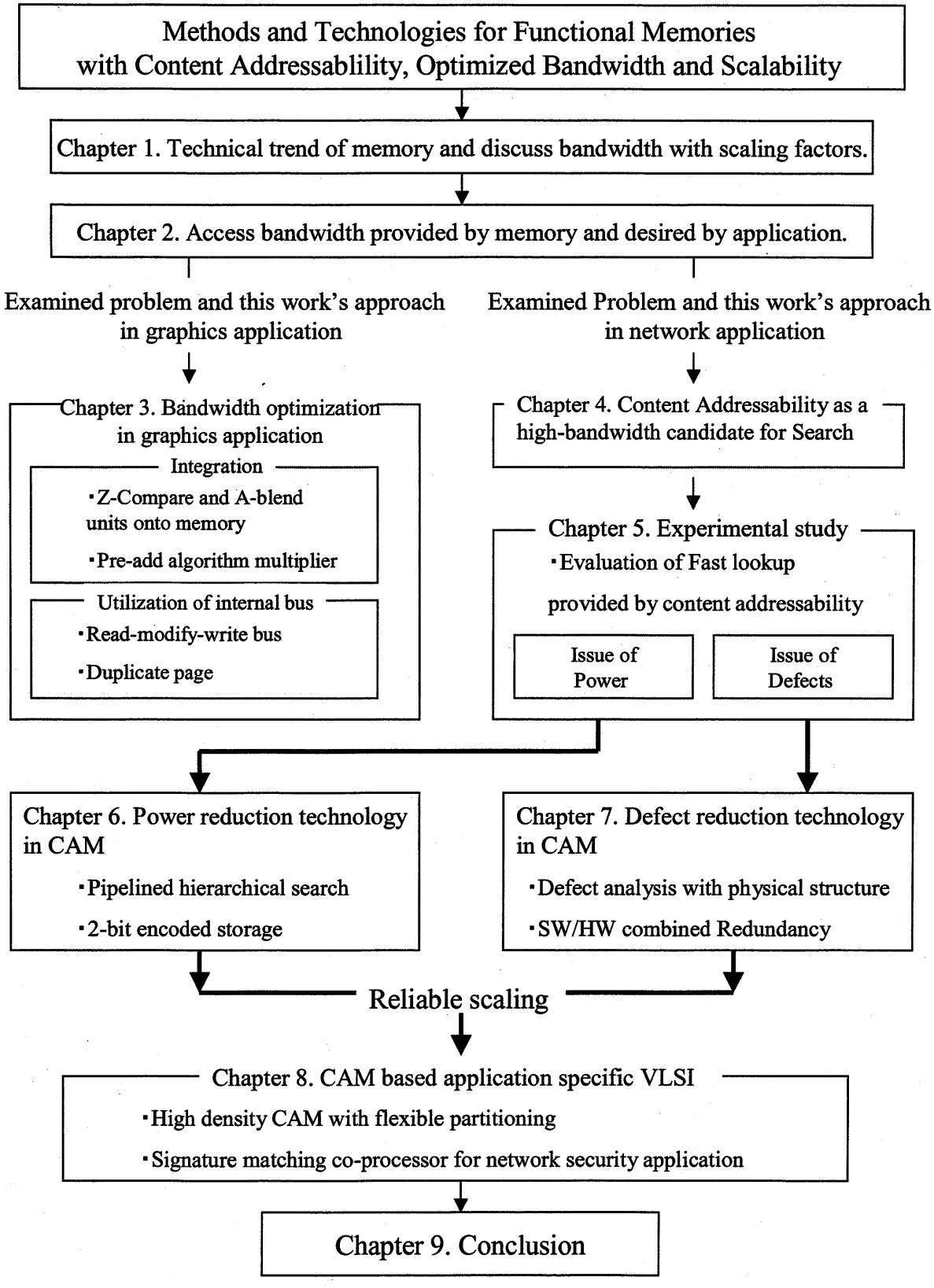


Figure 1.2 Structure of the Thesis

References

[1] G. E. Moore, "Cramming more components onto integrated circuits," IEEE Electronics, Vol. 38, No. 8 (Apr. 1965).

[2] P. P. Gelsinger, P. A Gargini, G. H. Parker, and A. Y.C. Yu, "Microprocessors circa 2000," IEEE Spectrum, pp. 43-47 (Oct. 1989).

[3] B. S. Amrutur and M. A. Horowitz, "Speed and Power Scaling of SRAM's," IEEE Transaction on Solid-State Circuits, Vol. 35, No. 2, pp. 175-185 (Feb. 2000).

[4] V. V. Zhirov, R. K. Cavin, J. A. Hutchby, and G. I. Bourianoff, "Limits to Binary Logic Switch Scaling," Proceedings of the IEEE, Vol. 91, No. 11, pp. 1934-1939 (Nov. 2003).

[5] R. Pinkham, M. Novak, and K. Guttag, "Video RAM excels at fast graphics," Electronic Design, vol. 31, no. 17, pp. 161-168 (Aug. 1983).

[6] H. Kawai, Y. Inoue, J. Kobara, R. Streitenberger, H. Suzuki, H. Negishi, M. Kameyama, K. Inoue, Y. Horiba, and K. Fujishima, "A Programmable Geometry Processor with Enhanced Four-Parallel SIMD Type Processing Core for PC-Based 3D Graphics," IEICE Trans., Electron, Vol. E85-C, No. 5, pp 1200-1210 (May 2002).

Chapter 2
Classification of memories and their bandwidth capabilities

2.1 Introduction

Although semiconductor memories are typically classified by their physical cell structure, each of them is categorized by functional aspect in the application. For example, volatile memory is one category and is used for the applications managing both read and write operations, while the applications using non-volatile memory mostly manage read operations. Dynamic cell vs. static cell memory affects the area on silicon, and hence its usage depends on whether the application is density prioritized or not. Likewise, it is completely the responsibility for users and applications to make a choice from the categories of memory types. It seems that today's various semiconductor memory line-ups have already matched to application's requirement. Nonetheless, some applications prefer their own architecture and functionality in memory hardware because memory performance is often restricted by its narrow bandwidth [7 - 10]. This chapter presents examples of considerable bandwidth differences between what can be provided by the memory hardware and application's requirement. I set out to start the discussion in the field of graphics applications. In addition, a discussion is presented for the area of network applications, where employing of data-search operations for further high bandwidth is demanded.

2.2 Overview of semiconductor memories with respective features and benefits

Figure 2.1 illustrates the technological trend of memory bandwidth. Note that the growth is not viewed for one type of memory, but for various different types of DRAM and SRAM memory [11, 12], e.g. DDR SDRAM achieves 50M bit-per-second (-bps) and QDR SRAM achieves over 200M-bps. The definition of bandwidth in this figure indicates not the peak but the average performance during random access. For example, DDR500

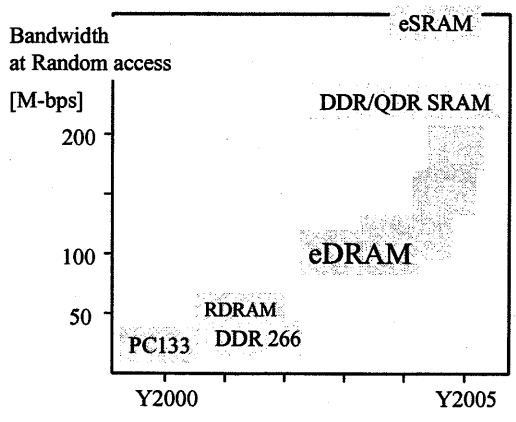


Figure 2.1 Trend of memory bandwidth

(not included in the figure) should almost double the performance of DDR266. However, its actual performance is not that much if it is measured under the condition of random access including page misses. The growth speed of memory bandwidth is slower than that of memory density, and it is often pointed out that the performance bottleneck in many applications is caused by the low access bandwidth provided by external standalone types of memory. A noticeable additional technical trend for increasing memory bandwidth is the solution of embedded types of memory. Embedded memory is now very popular and is the origin of today's "System on Chip" (SoC) trend, aiming at specific applications, where DRAM based technology (eDRAM) as well as SRAM based technology (eSRAM) are involved.

In the following examples, applications cannot accept the lack of sufficient memory bandwidth, and that is why applications are certainly the technology drivers of high-bandwidth specific memory VLSIs. Let us take a look at these examples, which are the graphics application and the network application, and examine the true bandwidth requirements resulting from each of these applications.

2.3 Bandwidth requirements for frame-buffer displaying in graphics applications

A frame-buffer is a specific memory used in the field of computer graphics applications. Its function is to store the entire picture frame of pixel data displayed on the screen. Over the past a few decades, the screen size defined by the number of pixels {Px, Py} in horizontal and vertical direction of the screen keeps increasing as follows; {1,024, 768} (XGA), {1,280, 1,024} (SXGA), {1,600, 1,200} (UXGA) and beyond. The underlying technology direction, namely pursuance of wider screen size and higher resolution, is driven by graphics intensive applications and users, which work on higher system layers than the memory hardware layer. Additionally, the steady increase of the density is a standard technical trend for bottom layer of memory hardware. The problem is that the growth speeds of the bandwidth delivered by the memory and the system requirements for the frame buffer do not match each other. As a result the difference between the bandwidth provided by the memory and the bandwidth desired by the frame-buffer keeps increasing. For example, a slow refresh cycle in combination with a large screen size is very uncomfortable for the human eye. Larger screen size therefore constrains the frame-buffer to function at a higher refresh rate. Typical

refresh-frequency requirements coming from the human visual sensitivities are found to be 60 frame-per-second (-fps), 75-fps and 85-fps for the screen sizes of XGA, SXGA and UXGA, respectively. Consequently, the required bandwidth for the frame-buffer display is generally viewed as a strong growth curve given by the relation.

Application: Frame- buffer display

Bandwidth required = ~1.4 · [screen size] · [refresh cycle] · [-bpp]

Here, the multiplying factor ~1.4 comes from the blanking time on the screen, and allows the display to return from the end-pixel position {Px, Py} to the start-pixel position {0, 0}. Not only the larger screen size, but the shorter refresh cycle requires a further increase of the memory bandwidth. The last multiplying factor bit-per-pixel (-bpp), meaning the number of bits which represent a pixel, is also a driver that demands more memory bandwidth. For example, the conventional 8-bit per pixel representation, which provides a resolution of 256 colors, has grown to 24-bit per pixel and thus a resolution of 16,777,216 colors. Consequently, totally required bandwidth for the frame-buffer of the display has to follow the increase in screen size, the shorter refresh cycle and also the higher color depth. The resulting bandwidth demands are:

Bandwidth [XGA] = 1.4 · [1,024 · 768] · [60-fps] · [8-bpp]
= 528.5M-bps

Bandwidth [SXGA] = 1.4 · [1,280 · 1,024] · [75-fps] · [16-bpp]
= 2.21G-bps

Bandwidth [UXGA] = 1.4 · [1,600 · 1,200] · [85-fps] · [32-bpp]
= 7.32G-bps

The major usage of screen resolution keeps increasing such as from XGA to SXGA and from SXGA to UXGA, and it seems to be happening approximately every four years. Therefore, the scaling factor for the frame-buffer bandwidth in the graphics-display application is about 4 times growth per 4 years time period.

Present Bandwidth (BW) = 4 • *Bandwidth 4 Years ago (BW0)*

The growth of bandwidth coming from reliable frame-buffer scalability should therefore realize this factor four in four years, when comparing the bandwidth capacity of previous technology generations. This requirement for the scaling factor of the frame-buffer bandwidth is unfortunately much higher than the real bandwidth scaling provided by the technology development of DRAM families. A solution would be that the frame-buffer application migrates from the use of DRAM to the use of SRAM. However, this would certainly affect another type of scaling factor, namely the reduction of cost per bit of the frame buffer in a very negative way. It is therefore understandable that the frame-buffer application has been exploring adequate DRAM-based high-bandwidth solutions as an application specific frame-buffer memory, which can maintain all required scaling factors.

2.4 Bandwidth requirements for the packet forwarding in network applications

In this section, I will explain a second example of bandwidth differences between the requirements of the application's system algorithm layer and the bandwidth deliverable by the bottom hardware layer, which seriously affected the field of networking applications.
The advent of the Ethernet has dramatically improved the network infrastructure, and has achieved tremendously increased data transferring capacity, viewed as a magnitude scaling factor from 100M-bps, G-bps, to 10G-bps. With this growth of network communication speed, the packet forwarding algorithms have to be executed faster and are becoming also more complicated. While packet forwarding only evaluated the destination address in older network-router generations, nowadays much larger data structures have to be managed. For example, the previously 32-bit IP header is increased to 128-bit to perform various filtering and/or queuing-priority algorithms, during the change from IP version 4 to 6. The bandwidth required for the packet forwarding is given by product of following three factors.

Application: packet forwarding

Bandwidth required = [packet flow speed] • [header size to refer to] • [-spp]

The first 2 factors are the growth of the "packet flow speed" and the "header size to refer to"

for executing the forwarding algorithm. As quantitatively mentioned above, both factors have recently increased. The packet flow speed is grown by about 2 orders of magnitude and the header size is grown by about a factor 4. The additional multiplication factor "searches-per-packet (-spp)" depends on the database size of reference data.

Typically, possible operations managed by memory hardware are write and read, in addition to just hold the data (store) which is also a fundamental task of the memory. While a search operation for something previously memorized is a commonly managed task in our human brain, basic operations manageable by the memory hardware do not involve search as a basic operation. To accomplish a search operation with conventional memory, it is necessary to read the entire data of the memory and to perform external compare operations. Once the stored data is read out of memory with its bandwidth capacity, it has to be compared with the search-request data one by one. This external search methodology is relatively easy and sufficiently fast when the database size is small, but becomes more expensive in terms of more latency until completion with the increase of the database size. The search performance and the rate of searches-per-second [-sps] can be calculated as below.

Search rate [-sps] = [database size] / ([-bps] bandwidth of read operation)

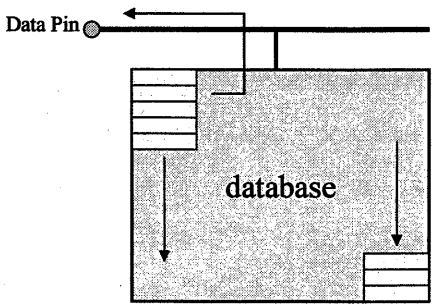
It is certain, that the factor of searches-per-second is deeply related to the entire database size of Internet population, which grows very rapidly. Similar to the graphics application, but more serious in quantity, the difference between bandwidth requirements by network applications and bandwidth capabilities of the memory hardware is constrained by the increase of packet flow speed, packet header size and database size. I am most concerned here with the scaling of bandwidth due to the database size, as this size has already grown by orders of magnitude. Just the task to find an appropriate solution with sufficient reading bandwidth among existing memories is already hopeless. As a result, such frustrating situations accelerated the appearance of functional memory, devoted to supply the high bandwidth capability needed by the specific application.

2.5 Content addressability viewed as a value-added high-bandwidth solution

The memory-access bandwidth discussed in terms of the unit of bits-per-second [bps]

alone is often not sufficient to reflect the bandwidth requirements and the actual memory usage in the application. As explained previously in the description of the bandwidth requirements for the frame-buffer in graphics applications, the frame-refresh cycle indicated by frames-per-second [fps] and the pixel configuration within the frame indicated by bits-per-pixel [bpp] are also important units to evaluate in the graphics field. Similarly, in the application of packet forwarding, the packet-flow speed indicated by packets-per-second [pps], and the number of search operations per packet indicated by searches-per-packet [spp] are also important units to qualify the bandwidth. It is not surprising that specific applications prefer their own unique indicators to examine and express their true bandwidth requirements.

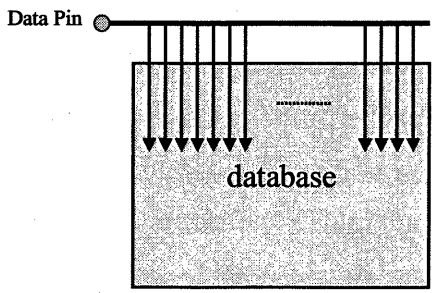
Expensive search operations are among the commonly operated tasks in the network application field for network functions such as packet forwarding. Unfortunately, due to the functional limits of conventional memory technology, the bandwidth provided by conventional memories has hardly ever caught up with the requirements of such search operations. As shown in Figure 2.2, the bandwidth provided by the external data pins of a conventional memory is limited because data blocks, whose data capacity is illustrated by the white rectangles in the figure, have to be consecutively accessed and read out from the entire database until the final block address is processed.



Bandwidth capability
= Data Pin Counts times Operating Frequency

Search Performance = Database / Read Bandwidth

Figure 2.2 Search with Read Operation



Bandwidth capability
= Full Database size times Operating Frequency

Search Performance = Database / Bandwidth
= Const.

Figure 2.3 Search with content addressability function

On the contrary, a memory with content addressability, where the search-request data given

at the input data pins can be fully compared with entire stored database in parallel as shown in Figure 2.3 [13, 14], and is thus capable of search operations with a robust and maximized effective bandwidth. An increase of the database size is a serious influential factor for the search with a conventional memory by exploiting the read operation, since the search performance is determined by the entire database divided by the memory's external bandwidth capability. On the other hand, the external bandwidth required for a memory with content addressability can remain constant as the performance in terms of the number of searches is independent of the memory's database size.

The following chapters describe the conducted research and the supporting experiments for the memory's effective bandwidth optimization through functionality integration. The case of graphics applications is covered in chapter 3 and the case of network applications is covered in chapter 4 through 7, respectively. The main reason for devoting more space to the network than to the graphics application originates from the additional "scalability" problem of conventional memories with content addressability. I had to solve this scalability problem, before being able to develop application specific VLSI based on content addressability. Furthermore, content addressability is of greater current interest in recent years as a main candidate for intelligent high-bandwidth solutions.

References

[7] D. Burger, J. R. Goodman, and A. Kagi, "Limited Bandwidth to affect Processor Design," IEEE Micro, pp.55-62 (Nov./Dec. 1997).

[8] T. Tsuruda, M. Kobayashi, M. Tsukude, T. Yamagata, K. Arimoto, and M. Yamada, "High speed high-bandwidth design methodologies for on-chip DRAM core multimedia system LSI's," IEEE Journal of Solid-State Circuits, Vol.32, pp.477-482 (Mar. 1997).

[9] S. Miyano, K. Numata, K. Sato, T. Yabe, M. Wada, R. Haga, M. Enkaku, M. Shiochi, Y. Kawashima, M. Iwase, M. Ohgata, J. Kumagai, T. Yoshida, M. Sakurai, S. Kaki, N. Yanagiya, H. Shinya, T. Furuyama, P. Hansen, M. Hannah, M. Nagy, A. Nagarajan, and M. Rungsea, "A 1.6GB/s data-transfer rate 8Mb embedded DRAM," ISSCC95, Dig. of Technical Papers, pp.300 (1995).

[10] T. Watanabe, R. Fujita, K. Yanagisawa, H. Tanaka, K. Ayukawa, M. Soga, Y. Sugie, and Y. Nakagome, "A modular architecture for a 6.4GB/s, 8M-bitmedia chip," Symposium on VLSI Circuits, Dig. of Technical Papers, pp.42 (1996).

[11] M. Kumanoya, T. Ogawa, Y. Konishi, K. Dosaka, and K. Shimotori, "Trends in High-Speed DRAM Architecture," IEICE Trans. Electron, Vol.E79C, No.4, pp.472-481 (Apr. 1996).

[12] M. Kumanoya, T. Ogawa, and K. Inoue, "Advances in DRAM interfaces," IEEE Micro, Vol.15, No.6, pp.30-36 (Dec. 1995).

[13] M. Uga, M. Omotani, and K. Shimoto, "A High-Speed Packet Classification Using TCAM," IEICE Trans. Communication, Vol.E85-B, No.9, pp.1766-1773 (Sept., 2002).

[14] Nikkei Network, pp.63-68 (June, 2001).

Chapter 3
Bandwidth optimization in memories for graphics applications

3.1 Introduction

A computer graphics displayed on the screen consists of plurality of polygons, which comprises a number of pixels, and the computation of inner pixels referred to as vertex pixels is a task named rendering. Pixel generation exactly means to be consecutively demanding bandwidth and therefore it makes clear that various high bandwidth approaches and researches are originated in the application of computer graphics. [15]

Figure 3.1 shows the example of flat shading, which inner pixels comprise constant and same data as that of vertex pixels.

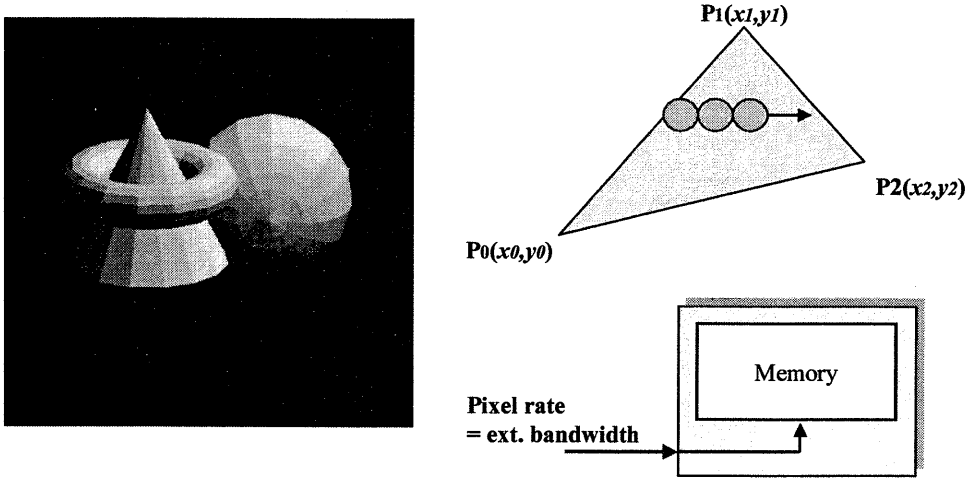


Figure 3.1 Rendering with Flat shading

The task of rendering repeatedly demands bandwidth as many as the number of inner pixels per polygon. When a polygon consists of 15-pixel, external I/O pin provides the pixel data consecutively for fifteen times. It is known that the internal data bus is capable of more bandwidth, and therefore it should take place of consecutively demanding bandwidth at I/O pin. Apparently, the utilization of internal bus in this application can save the bandwidth demanded at I/O pin with the factor one over fifteen if internal data bus provides pixel data

fifteen times more. Fortunately, it is not difficult to implement more data bus internally than that of I/O pins, therefore that utilization of internal bus instead of I/O pin takes advantage of fast rendering speed with high bandwidth provided by wide bus. In fact, this concept is well researched and the origin of today's memory embedded SoC LSI for various applications.

This work is intended to further increase the performance and rendering quality without the need to increase the external bandwidth. Figure 3.2 shows another example named smooth shading, in which inner pixels are not carried out by external I/O bus, but are provided by the integration on memory which inner pixel is computed as linear interpolation between plural vertex pixels. The comparison of these figures simply tells us that the functional integration onto memory can improve the rendering quality as well as the advantage of described bandwidth saving.

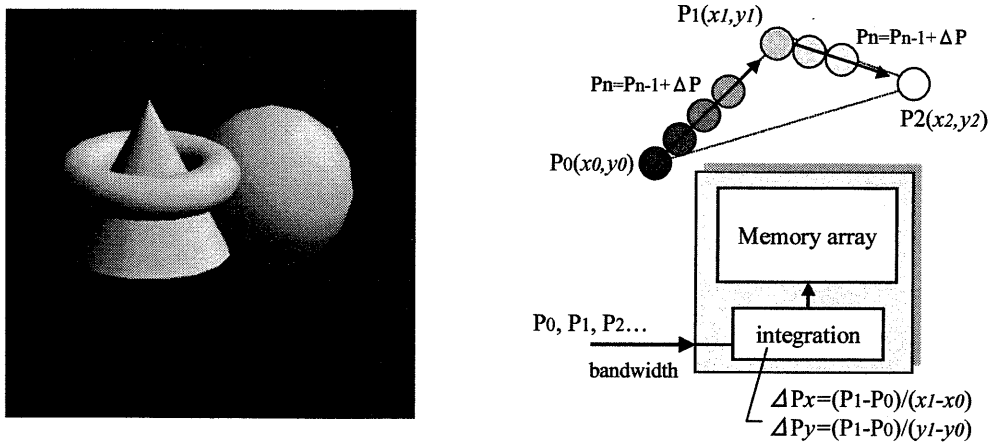


Figure 3.2 Rendering with Smooth shading

Following sections present further effects provided by the functional integration in the experimental frame-buffer for 3D graphics application.

3.2 Technologies carried out by the integration on memory

This chapter presents the experimental frame-buffer with features implemented on the memory. The key concept in this high bandwidth approach is not provided by external I/O bus, but the functional integration and the utilization of internal bus. Needless to say, the integration causes overhead in hardware, hence the effect of performance versus cost of hardware is a serious matter of concern.

3.2.1 Z-compare and A-bend units on 3D frame-buffer

In typical usage of high-end 3D graphics system, a pixel consists of at least 32-bit RGBA for color and 32-bit Z for depth. The 24-bit RGB encodes 16.7M colors, termed True Color, and 8-bit A encodes the transparency. In low-end 2D graphics systems, by contrast, a color is typically expressed by only 8-bits per pixel, for 256-color resolution. A unique aspect of 3D graphics is depth. A 32-bit Z is used to position the object towards the front or back. Therefore, a 3D system typically comprises of 64-bit per pixel in total, which is eight times more costly than a 2D system. Similarly, the bandwidth demanded for the 3D frame-buffer is eight times as costly as that of 2D. However, cost of pixel is not the only problem, the read-modify-write operation employed by fundamental 3D algorithm creates a bottleneck in 3D rendering performance. The frame-buffer bandwidth has to be divided into half for the read operation, which should be done before write operation starts, and the other half for the write operation. This means that 50% of the bandwidth is lost due to the writing into the frame-buffer when compared to 2D. The first graphics operation where read-modify-write is required is in the depth test. In Z-buffer algorithm, the current object on the screen is called Z-source (Z-src.), while the depth data being written into is called Z-destination (Z-dst.), and similarly Color-source (Color-src.) means the existing old color data on the screen, while Color-destination (Color-dst.) means the new color data being written into. First, Z-src. is read out of the frame-buffer and is compared with Z-dst. If the value of Z-dst. is smaller than Z-src., then destination is in front of source, therefore it is judged as dst. is visible on the screen. This is called Z-pass and following color can be changed from color-src. to color-dst. On the other hand, if the value of Z-dst. is larger than Z-src., it is judged as Color-dst. should be behind Color-src., and Color-dst. is invisible on the screen. This is called Z-fail and the color should keep the source data. The depth test between Z-src. and Z-dst. should be identified before the operation regarding color. This is the first read-modify-write function

required for the 3D frame-buffer.

The second read-modify-write function required in 3D graphics is in the color. Similar to the depth test, Color-src. is read out of the frame-buffer and is namely blended with the Color-dst. In a typical 3D pixel, color consists of RGB and the measure of transparency A ($0 < A < 1$). If factor of A=0 in the destination, the color blending result consists of 100% color-src. and 0% color-dst. even if the result of depth-test is Z-pass. Factor A=0.5 means opaque, hence the blending result consists of 50% of the color-src. and the other 50% of the color-dst. Lastly, if factor A=1, the blending result of color is completely changed from Color-src. to 100% Color-dst. Described blending formula is summarized as follows.

$$\text{Result color} = (1\text{-}A) \cdot [\text{color-src.}] + A \cdot [\text{color-dst.}]$$

Figure 3.3 shows the block diagram of experimental frame-buffer, which contains a couple of new functional integrations. DRAM divided into Bank-1 through Bank-4 is used for the frame-buffer to store the pixel data, and Video-buffer devoted to the display on the screen. The first integration of SRAM based level-1 cache is placed between DRAM and Pixel ALU to hide the refresh penalty of DRAM and to enable the consecutive pixel generation. [16,17] The second integration pixel ALU consists of two serial processors, one is performing the depth test and the other is performing the blending operation, both Z-compare unit and the A-blend unit are fed through 7-stage pipeline. The write address is generated 7-stages later from a read address, therefore the source data will be written back to the same address in the cache after passing through the pixel ALU. [18]

These functional integrations remove the necessity of read operation at external I/O pin. The experimental 3D frame-buffer converts the conventional read-modify-write into pure write which reduced the external bandwidth with the factor of 50%.

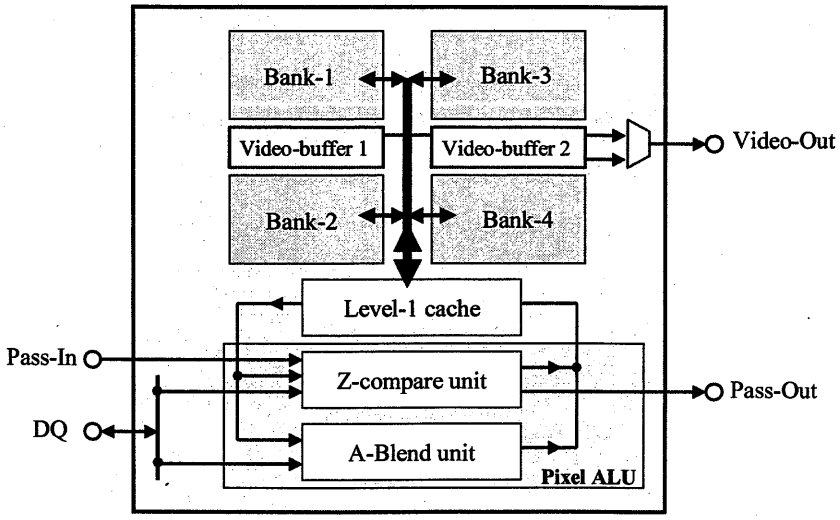


Figure 3.3 Block diagram of experimental 3D frame-buffer

Figure 3.4 shows the typical usage of experimental frame-buffer, which can be used for both Z-chip and the color-chip. As described in this figure, bandwidth demanded at I/O pin is pure-write instead of expensive read-modify-write, because the read-modify-write is completely replaced to the internal bus by the effect of functional integration.

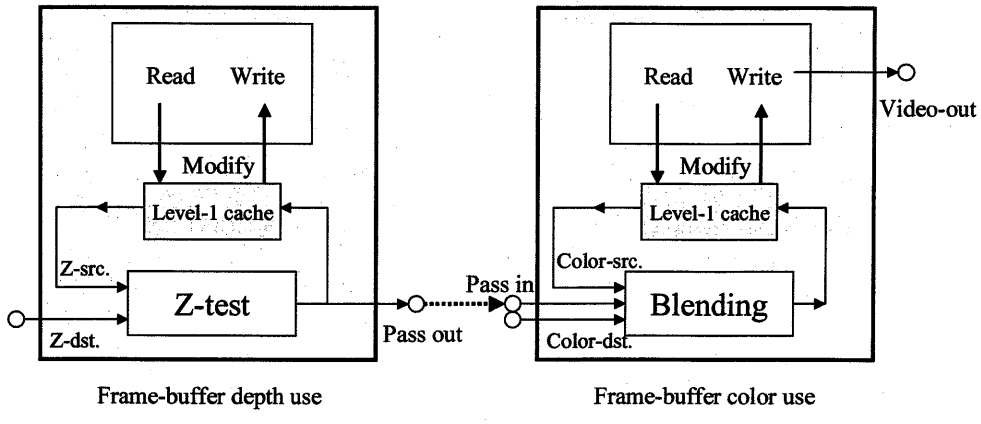


Figure 3.4 Experimental frame-buffer for Z and for color

The result of the depth test is available at the pass out pin of the chip holding the Z values. The color chip receives the output of Z-chip at the pass in pin. If the output is Z-pass, blending is executed and new frame data will be put back into the cache. If the output is Z-fail, blending is ignored but src. data should be put into the cache. In this way, a 3D system using this experimental frame-buffer for Z and for color performs the depth test and the blending in parallel without demanding the bandwidth caused by the read-out of source data externally. That is the benefit given by the usage of functional integration. Actual performance of pixel through-put transferring rate verified in fabricated 3D frame-buffer is 32-bit times 100MHz and 66-bit times 166MHz, in a design with 0.5um CMOS technology and 0.25um CMOS technology, respectively.

I have outlined the performance benefit provided by functional integration with the example of experimental 3D frame-buffer. This is the evidence that the depth test and the blending can be consecutively performed in pixel-by-pixel manner, without demanding bandwidth externally. However this evidence carries out other technical problems, one is the cost of hardware caused by the integration and the other is the replacement of performance bottleneck from the external I/O pin to the internal bus. The integration certainly causes an overhead on the silicon area, which means the cost of bit-cell is relatively increased by the integration. The implemented circuitry should be therefore as compact as possible. In this 3D frame-buffer, the integration related to the depth-test is easy because the components required are magnitude-comparing which exclusive-or circuitry takes place, on the other hand, the integration performing the blend is rather expensive because it consists of the multiplier and adder for each color channel A, R, G, and B. I am therefore concerned with the architecture of blend unit as compact. The other problem is that performance bottleneck is replaced from the external I/O pin to internal bus. In general, performance gain is ensured by the complete removal of all bottleneck factors. These problems are arisen in experimental frame-buffer fabricated in 0.5um technology, and therefore described additional problems are discussed and proposed solution can be applied into the other experiment fabricated in 0.25um technology.

3.2.2 Pre-add multiplier for cost reduction of integration

The concept of proposed Pre-add multiplier is to make two separated multiplying and final adding compact. Figure 3.5 shows the conventional hardware structure to perform the

equation A*B+C*D, which is applied to the application blending described in previous section. Here, I used the assumption which A (C) consists of [n]-bit length and B (D) consists of [m]-bit length, respectively. The intermediate result A*B (=S0) and C*D (=S1), should work in parallel to keep the throughput rate as constant and as high. Therefore, the expense of hardware is 2*[m][n]-bit adder-cell in this adder tree. Also, the final addition A*B+C*D (=S0+S1) seen in bottom, comprises [m+n+1]-bit adder-cell. Hence, total number of adder-cell is summed up to [2mn+m+n+1]-bit. Note that, the discussion above follows to the conventional carry-save algorithm, which is one of major multipliers in hardware.

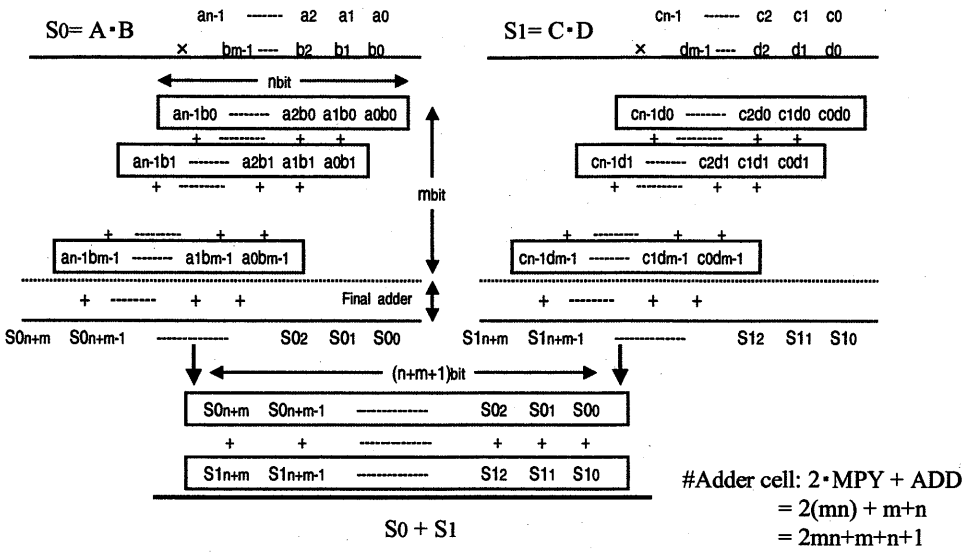


Figure 3.5 Conventional carry-save algorithm

Figure 3.6 illustrates the hardware structure of proposed Pre-add algorithm multiplier performing the composite equation A*B+C*D in a single adder-tree.

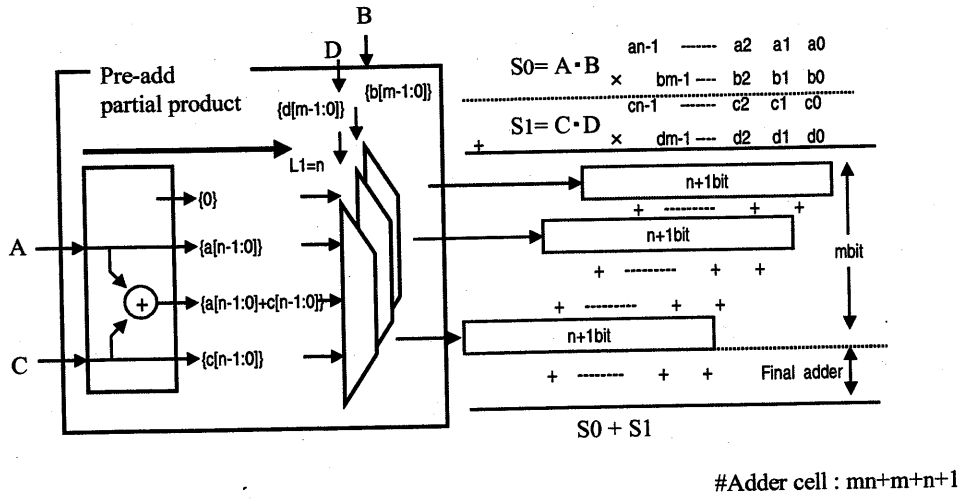


Figure 3.6 Proposed Pre-add algorithm

A partial product seen in the left contains signal "A","C" for the input of the pre-adder, and signal "B","D" for the input indicator of MUX. The partial product results "0","A","C", or "A+C" by the input indicator of MUX, because that is provided as one of following four cases.

$$
\begin{aligned}
[A*B+C*D] &= 0 \quad \text{if } \{B=0 \;\&\&\; D=0\} \\
[A*B+C*D] &= A \quad \text{if } \{B=1 \;\&\&\; D=0\} \\
[A*B+C*D] &= C \quad \text{if } \{B=0 \;\&\&\; D=1\} \\
[A*B+C*D] &= A+C \quad \text{if } \{B=1 \;\&\&\; D=1\}
\end{aligned}
$$

The concept of proposed Pre-add algorithm shown in the right side of the figure combines two multiplications and the final addition together. The first column's [n+1]-bit represents {{an-1:a0}b0}+{{cn-1:c0}d0}. Similar to the above, possible result is given by one of the following four cases.

The first [n+1]-bit

$$
\begin{aligned}
&\{\{an\text{-}1:a0\}b0\} + \{\{cn\text{-}1:c0\}d0\} = \{0\} && \text{if } \{b0=0 \;\&\&\; d0=0\} \\
&\{\{an\text{-}1:a0\}b0\} + \{\{cn\text{-}1:c0\}d0\} = \{an\text{-}1: a0\} && \text{if } \{b0=1 \;\&\&\; d0=0\} \\
&\{\{an\text{-}1:a0\}b0\} + \{\{cn\text{-}1:c0\}d0\} = \{cn\text{-}1: c0\ \} && \text{if } \{b0=0 \;\&\&\; d0=1\} \\
&\{\{an\text{-}1:a0\}b0\} + \{\{cn\text{-}1:c0\}d0\} = \{an\text{-}1 + cn\text{-}1 : a0+c0\} && \text{if } \{b0=1 \;\&\&\; d0=1\}
\end{aligned}
$$

The second column represents {{an-1:a0}b1}+{{cn-1:c0}d1}, so that "b1" and "d1" indicate the result of the second [n+1]-bit . That is,

The second [n+1]-bit

{{an-1:a0}b1} + {{cn-1:c0}d1} = {0}	if {b1=0 && d1=0}
{{an-1:a0}b1} + {{cn-1:c0}d1} = { an-1: a0}	if {b1=1 && d1=0}
{{an-1:a0}b1} + {{cn-1:c0}d1} = { cn-1: c0 }	if {b1=0 && d1=1}
{{an-1:a0}b1} + {{cn-1:c0}d1} = {an-1 + cn-1 : a0+c0}	if {b1=1 && d1=1}

With repeating until {{an-1: a0}bm-1}+{{cn-1: c0}dm-1}, which is seen in the last column of the figure, the behavioral adder-tree is exactly the same as the equation A*B+C*D. That is to say, two multiplications and the final addition can be put into a single adder-tree unit. Regarding the expense of hardware, first [n]-bit adder-cells appear in the partial product seen in the left side. [m(n+1)]-bit adder-cells appear in the adder-tree seen in the right side. Hence, total number of adder-cell is summed up to [mn+m+n]-bit. In comparison with the conventional carry-save algorithm, hardware reduction provided by proposed Pre-add multiplier is [mn-1]-bit times adder-cell as simple. The hardware reduction in multiplier is well-researched area such like Booth's algorithm and Wallace tree [19,20], which can be widely used. While proposed Pre-add algorithm states significant hardware reduction, that is applied to the specific application such as blending in 3D graphics.

The equation A*B+C*D can be additionally used in other graphics applications in the filtering theory. Figure 3.7 shows the comparison between the case of non-filtering and filtering. The right figure, which a pixel consists of four sub-pixels, uses frame-buffer memory four times as many to display the pixels on screen, and the final pixel (S) is indicated by the average of four sub-pixels (S1, S2, S3 and S4). This filtering theory is especially named multi-sample filtering, that is expensive but provides higher quality.

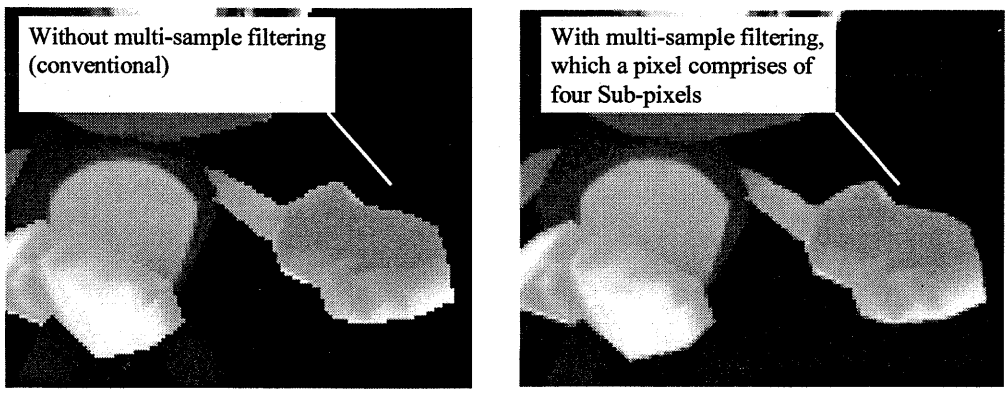


Figure 3.7 Example of multi-sample filtering for anti-aliasing

There are several filtering theories and techniques described as follows. The final value S can be indicated by numbers of Sn as shown in Figure 3.8. [21]

Box filtering

The interpolation here is the simple averaging from given samples. In the left drawing, the final pixel "S" is computed by the average of the range from –1-pixel to +1-pixel. The example in this figure is to average "S1" to "S2", because "S0" and "S3" are out of this range.

Result S =(S1+S2) /2

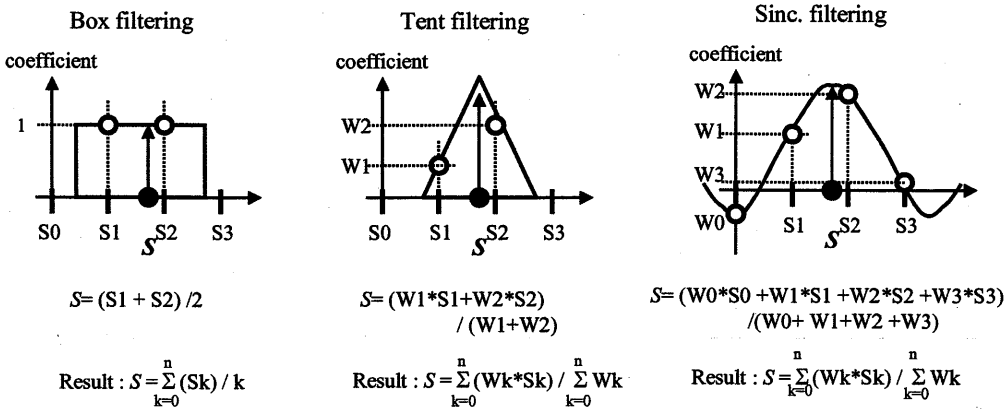


Figure 3.8 Various multi-sample filtering and equations

Tent filtering

The Tent filtering is a generalization of two linear interpolations. The assumption of the gradient between samples is constant. The example in the middle drawing ranges from –1-pixel to 1-pixel, but each sample indicates different coefficient as a weight. The weight is indicated as maximum-one at the point of resulted "S", and as minimum-zero if it's farer than 1-pixel.

Result S =(W1*S1+W2*S2) /(W1+W2)

Sinc. function filtering

Sinc. function interpolation is based on Fourier domain. The weighted coefficient W(x) is given by the following equation.

$$W(x) = \sin(\pi\ \chi) / (\pi\ \chi)$$

W(x) never goes to zero but we commonly cut-off far area just the definition of the limited range which is seen in the right.

$W(x) = \sin(\pi\ \chi) / (\pi\ \chi)$, $|x| < 2\text{-pixel}$

$W(x) = 0$ $|x| >= 2\text{-pixel}$

In actual HW, all the above filtering theories can be put into the sum of the flat level multiplication per each sample.

$$\text{Result } S = \Sigma(Wn*Sn) / \Sigma(Wn)$$

S : result

Wn : weighted coefficient

Sn : given sample

When Wn is normalized, $/ \Sigma(Wn)$ can be ignored. ($\Sigma(Wn)=1$).

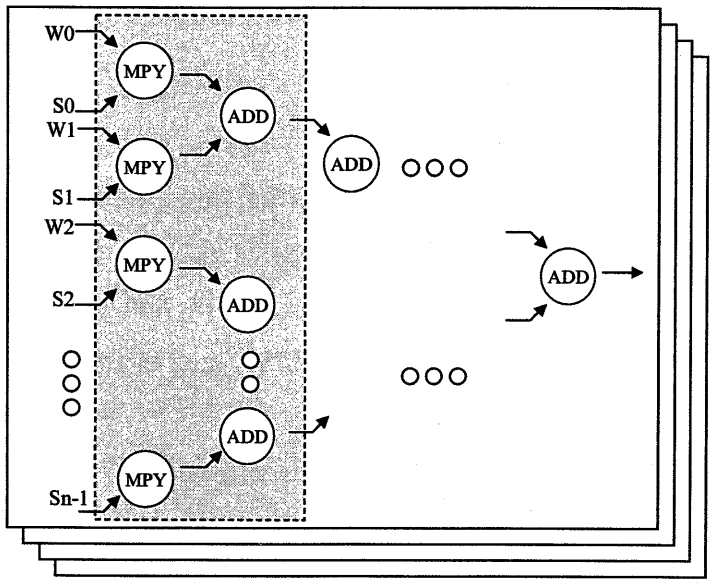


Figure 3.9 Weighted interpolation with proposed pre-add MPY

With respect to the structure of equations, here is an additional example that proposed Pre-add multiplier can be applied, that is shown as shadow area in Figure 3.9.

Here is a quantitative analysis between conventional carry save algorithm and Pre-add multiplier. Assuming that a pixel consists of A, R, G and B channel comprising 8-bit for each, and the number of sample to indicate the pixel is 16-sample and 16-weight of co-efficient for each color channel, totally required adder-cell for conventional carry save algorithm and for proposed Pre-add MPY algorithm are computed as follows.

#Adder tree in Carry save algorithm = 4-channel · 16-sample · (2mn+m+n+1)

When m=n=8bit, #Adder tree = 9,280

#Adder tree in Pre-add algorithm = 4-channel · 16-sample · (mn+m+n+1)

When m=n=8bit, #Adder tree = 5,184

This is an evidence that proposed Pre-add MPY significantly saves the number of adder tree, which is a cost effective algorithm despite the restriction within the specific application applied to the functional integration onto memory.

3.3 Technologies related to memory-internal data bus

I have described the integration performed onto memory and placed in conjunction with the internal data bus, which effectively improved the performance as well as the benefit of high bandwidth capability. I also showed the composite multiplying computation which is used for the component of integrations in the memory embedded LSI for graphics application can be compact by proposed Pre-add multiplier.

The second high bandwidth technology in this work is related to the utilization of internal data bus. [22-24] Figure 3.10 illustrates internal bus structure with representing possible bandwidth capability in memory. The bandwidth of external data pin is typically limited by I/O pin count. On the other hand, it is relatively easy to increase the bandwidth internally, which is provided by larger number of data bus functioning in parallel. As shown in this figure for example, 100M-bps at I/O pin can be improved as G-bps at the level of the internal data-bus. Furthermore, 100G-bps at the level of the sense-amplifiers and finally can be maximized as several T-bps at the memory cell level.

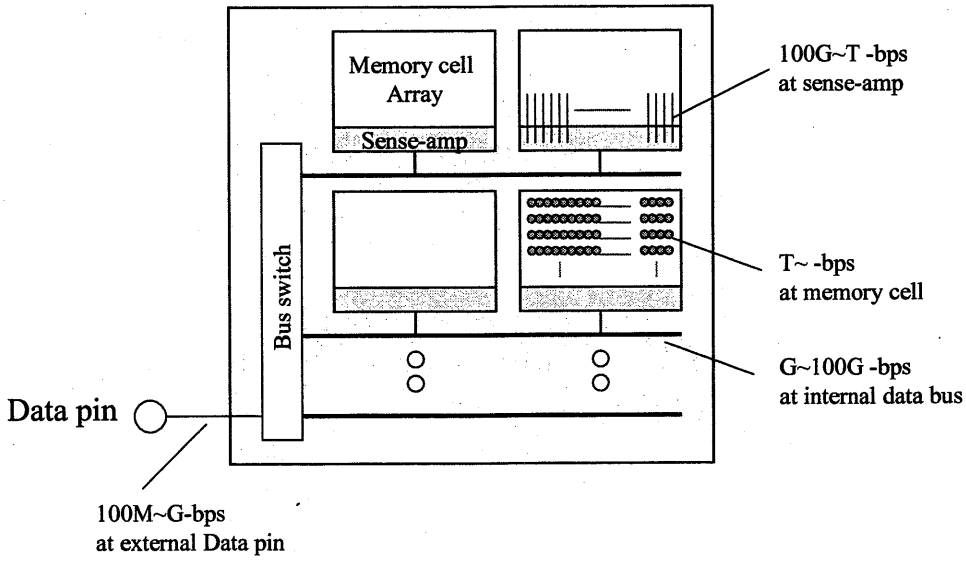


Figure 3.10 Memory architecture and bandwidth capacity

As described previously, the integration of pixel ALU has successfully converted the bottleneck, in read-modify-write operated by I/O pin into pure-write. The performance of rendering can be improved by as high as 2x, however the performance bottleneck is simply shifted from I/O-pin to the internal bus, which read-modify-write is still required. Figure 3.11 shows the example of benchmark in actual rendering application. There are a lot of open spaces in the column of I/O access, despite busy DRAM access. This means the effect of bandwidth saving at I/O pin is not as much as previously expected. This is one of the worst cases, however over 70% is idle at I/O access due to newly appeared bottleneck in the internal data bus. Therefore I am concerned with the optimization of internal bus, described in following section.

timescale	I/O access	DRAM access
10		ACP Bank A Page 0
20		ACP Bank B Page 0
30		ACP Bank B Page 0
40		ACP Bank B Page 0
50		RDB Bank A Blk 0
60		NOP
70	SFW pixel 0	RDB Bank A Blk 1
80	SFW pixel 1	NOP
90	SFW pixel 2	RDB Bank B Blk 0
100	SFW pixel 3	NOP
110	SFW pixel 4	RDB Bank B Blk 0
120	SFW pixel 5	NOP
130	SFW pixel 6	WTB Bank A Blk 0
140	SFW pixel 7	NOP
150		WTB Bank A Blk 1
160		NOP
170		WTB Bank B Blk 0
180		NOP
190		WTB Bank B Blk 1
200		NOP
210		RDB Bank C Blk 0
220		NOP
230	SFW pixel 8	RDB Bank C Blk 1
240	SFW pixel 9	NOP
250	SFW pixel 10	RDB Bank D Blk 0
260	SFW pixel 11	NOP
270	SFW pixel 12	RDB Bank D Blk 1
280	SFW pixel 13	NOP
290	SFW pixel 14	WTB Bank C Blk 0
300	SFW pixel 15	NOP
310		WTB Bank C Blk 1
320		NOP
330		WTB Bank D Blk 0
340		NOP
350		WTB Bank D Blk 1
360		NOP

I/O access activation ratio
= 22.2%
(77.8% is idle at I/O pin)

Figure 3.11 Worst case benchmark

3.3.1 Low voltage read-modify-write memory internal data bus

The differential I/O-bus architecture is commonly used technology in DRAM internal bus.

The reason of differential bus structure is because the sense-amp is too weak to drive the internal data bus completely during read operation. In write-operation, on the other hand the differential I/O-bus is fully driven by the write-driver and data is easily written into the sense-amp. In typical differential bus architecture, the access time is an issue in the path from sense-amp to the end of I/O-bus due to the difficulty of slight voltage management, while the power is an issue in the path from the write-driver to sense-amp due to the fully amplified data bus. Since differential I/O-bus only works as either the destination for read or the result-pixel for write, the actual pixel-rate is reduced to one half of I/O-bus bandwidth as shown in Figure 3.12.

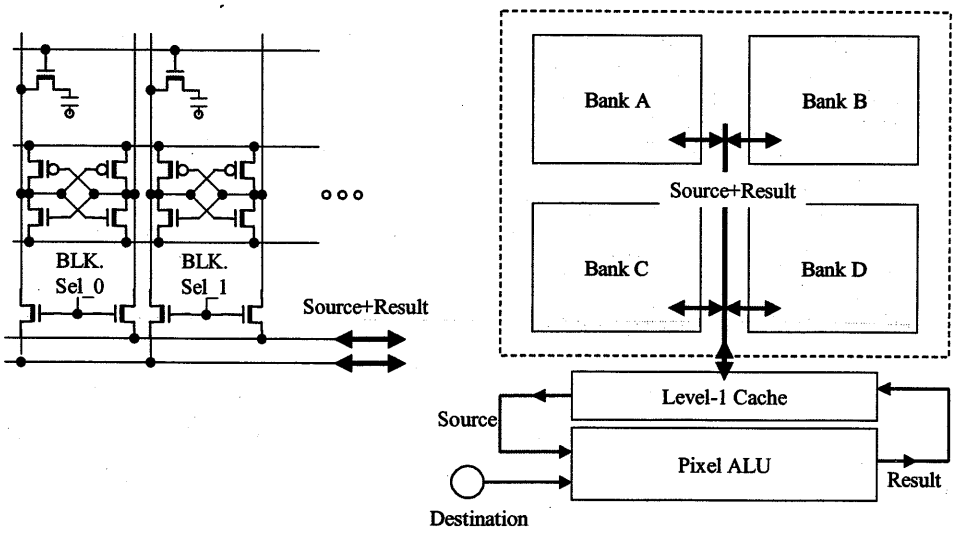


Figure 3.12 Conventional differential I/O bus

That is exactly same situation as I/O pin with read-modify-write in conventional frame-buffer. Therefore, this work should be intended to the optimization of internal bus for the purpose that destination and result-pixel work concurrently. It makes the pixel rate twice as fast as the differential I/O-bus. However, considerable problems arise from the proposed read-modify-write bus. One is the increase of bus size, and the other more serious problem is the noise due to the concurrent operation. Since the write-bus switches from ground-level to VDD-level quickly, it may affect the read-bus where signals appear slightly. A shielded power-line between write-bus and read-bus is a solution to eliminate this noise coupling problem, however it obviously required more space on silicon.

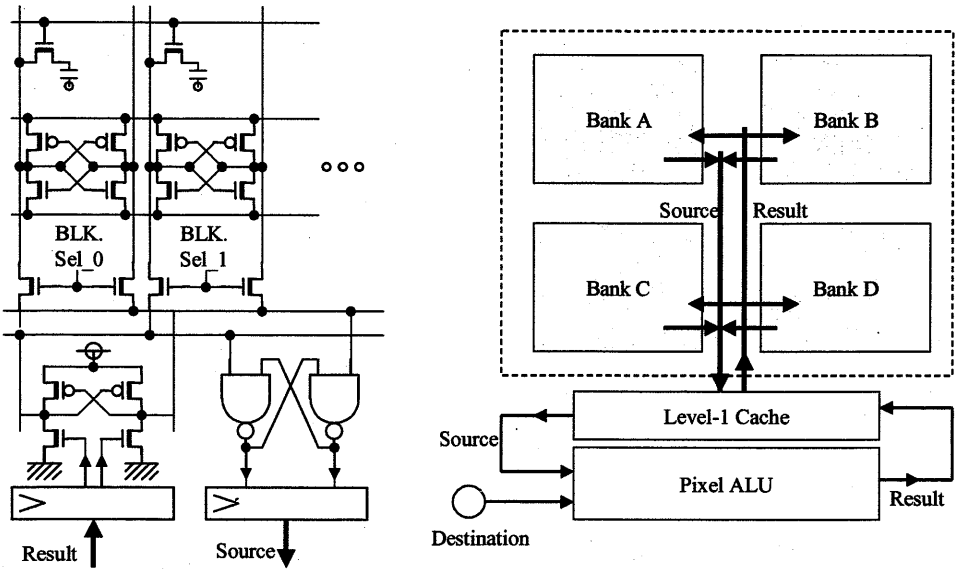
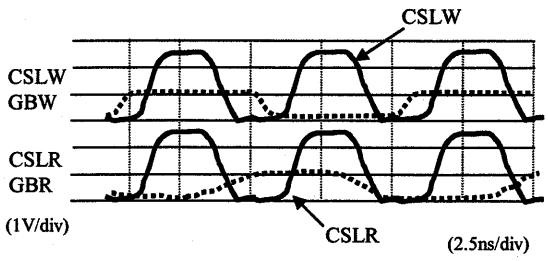


Figure 3.13 Read-modify-write bus in this work

In this work, the sense-amp is disconnected to the read/write bus directly, and a Data Transfer Buffer (DTB) is placed between the sense-amp and the read/write bus. Data bus consists of Global bus for Read (GBR) and Global Bus for Write (GBW) to achieve the concurrent operation, and separated Column Select Line (CSL) also works concurrently for the read of source address and for the write of result-pixel address.



CSLW : Column Select Line for Write
GBW : Global Bus for Write
CSLR : Column Select Line for Read
GBR : Global Bus for Read

Figure 3.14 Waveform of read-modify-write bus

timescale	I/O access	DRAM access	
10		ACP Bank A Page 0	
20		ACP Bank B Page 0	
30		ACP Bank B Page 0	
40		ACP Bank B Page 0	
50		RDB Bank A Blk 0	
60		NOP	
70	SFW pixel 0	RDB Bank A Blk 1	
80	SFW pixel 1	NOP	
90	SFW pixel 2	RDB Bank B Blk 0	
100	SFW pixel 3	NOP	
110	SFW pixel 4	RDB Bank B Blk 0	
120	SFW pixel 5	NOP	NOP
130	SFW pixel 6	RDB Bank C Blk 0	WTB Bank A Blk 0
140	SFW pixel 7	NOP	NOP
150	SFW pixel 8	RDB Bank C Blk 1	WTB Bank A Blk 1
160	SFW pixel 9	NOP	NOP
170	SFW pixel 10	RDB Bank D Blk 0	WTB Bank B Blk 0
180	SFW pixel 11	NOP	NOP
190	SFW pixel 12	RDB Bank D Blk 1	WTB Bank B Blk 1
200	SFW pixel 13	NOP	NOP
210	SFW pixel 14	RDB Bank A Blk 2	WTB Bank C Blk 0
220	SFW pixel 15	NOP	NOP
230	SFW pixel 16	RDB Bank A Blk 3	WTB Bank C Blk 1
240	SFW pixel 17	NOP	NOP
250	SFW pixel 18	RDB Bank B Blk 2	WTB Bank D Blk 0
260	SFW pixel 19	NOP	NOP
270	SFW pixel 20	RDB Bank B Blk 3	WTB Bank D Blk 1
280	SFW pixel 21	NOP	NOP
290	SFW pixel 22	RDB Bank C Blk 2	WTB Bank A Blk 2
300	SFW pixel 23	NOP	NOP
310	SFW pixel 24	RDB Bank C Blk 3	WTB Bank A Blk 3
320	SFW pixel 25	NOP	NOP
330	SFW pixel 26	RDB Bank D Blk 2	WTB Bank B Blk 2
340	SFW pixel 27	NOP	NOP
350	SFW pixel 28	RDB Bank D Blk 3	WTB Bank B Blk 3
360	SFW pixel 29	NOP	NOP

I/O access activation ratio
= 83.3%
(16.7% is idle at I/O pin)

Figure 3.15 benchmark with read-modify-write bus

DTB converts the slight signal appeared by the sense-amp into MOS-level signal to secure the read operation against the concurrent write operation, and that is transferred to GBR. In write operation, on the other hand DTB drives the sense-amp with respect to the data of GBW. Additional objective of DTB implementation is the power reduction to eliminate the noise problem. Proposed DTB and GBR/GBW convert the core voltage to 1V for their power-supply, while the sense-amp and the memory cell still use 2.5V. Figure 3.14 shows the waveform of GBR and GBW with CSL. The worst case benchmark with proposed read-modify-write bus is shown in Figure 3.15. The column of I/O access is dramatically improved in comparison with that of Figure 11, that is functional ratio at I/O pin is 83.3%. This is the evidence that the read-modify-write bus eliminates the bottleneck appeared in the internal data bus.

3.3.2 Duplicate page function with the utilization of sense-amplifier

In the experimental 3D frame-buffer, the integration is applied to the internal data bus with the benefit of high bandwidth provided by wider bus size than I/O pin. Here is another wide bus utilization provided by sense-amplifier to maximize the data bus for writing, that is applied to the application of fast erase of screen. The experimental frame-buffer operates a Duplicate Page Function (DUP) to accelerate the performance of screen erase.

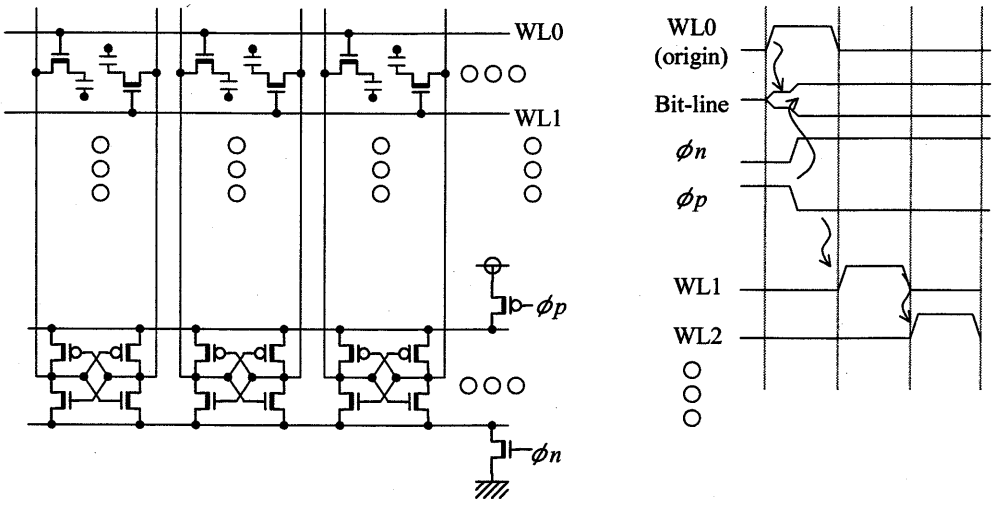
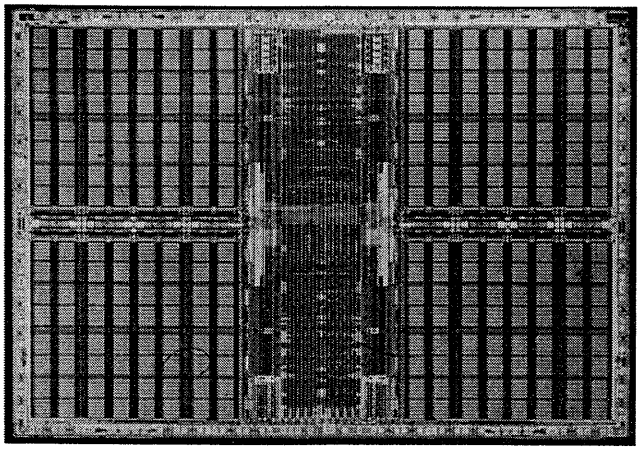


Figure 3.16 Duplicate Page (DUP) scheme

For example, to clear as a color of white means that every pixel should be written to RGB=24'h000000. Similarly, all pixels are set to RGB=24'hFFFFFF for black and RGB=24'hFFFF00 for yellow. Note that this application of screen erase does not require the read bandwidth at all, but only a high write bandwidth is an issue.

As shown in Figure 3.16, proposed DUP function does not use column address access. Data stored in sense-amps is copied directly from one row to another in a single cycle. Word-lines performing row-address per Bank are consecutively activated one after another to transfer data latched in sense-amps to each row. DUP function provides the maximized writing data capacity, which 4-Bank times 10,240-bit with the cost of 40ns, the bandwidth provided is 128G-Bps.

Die photos are shown in Figure 3.17 and 3.18 for fabricated 3D frame-buffer with 0.5um CMOS technology and 0.25um CMOS technology, respectively. Also Table 3.1 and 3.2 constitute the major characteristics.



10M-bit 3D frame-buffer	9.94 x 14.18 mm2	
Integration	21.6mm2	(15.3%)
Level-1 Cache	6.2mm2	(4.4%)
Z-Compare unit	6.2mm2	(4.4%)
A-Blend unit	9.5mm2	(6.7%)
others and glue logic	2.9mm2	(2.1%)

Figure 3.17 Die Photo of experimental frame-buffer with Z-compare and A-blend units

Table 1.1 Major characteristics of fabricated 3D frame-buffer with 0.5um CMOS

10M-bit 3D Frame-buffer		
	DRAM embedded	10M-bit DRAM consists of 4-Banks
	Functional integrations	Z-compare and A-blend
Data transferring rate		
	DRAM – Level 1Cache	256-bit at 50MHz
	Level 1Cache - Integration	32-bit at 100MHz
	External I/O	32-bit at 100MHz
Power consumption		
	DRAM	0.3W at 50MHz
	Cache and Integration	0.6W at 100MHz
Technology	0.5um CMOS 4-poly and 1-AL	

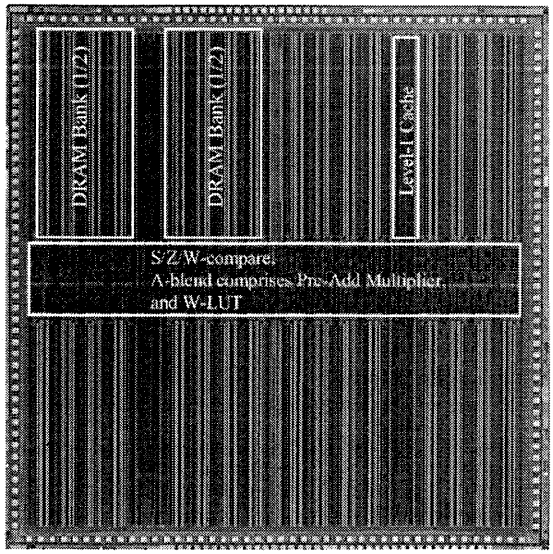


Figure 3.18 Die Photo of fabricated frame-buffer with Pre-Add MPY and 0.25um CMOS

Table 1.2 Major characteristics of fabricated 3D frame-buffer with 0.25um CMOS

40M-bit 3D Frame-buffer		
	DRAM embedded	40M-bit DRAM consists of 4-Banks
	Functional integrations	S/Z/W-compare, A-blend comprises Pre-Add Multiplier, and W-LUT
Data transferring rate		
	DRAM - Level 1Cache	1.2Kb for Read and 1.2Kb for Write at 83MHz (8-pixel/CLK for Src. and 8-pixel/CLK for Result)
	External I/O	66-bit at 166MHz (1-pixel/CLK)
Power consumption		
	DRAM	0.6W at 83MHz
	Cache and Integration	1.0W at 166MHz
Technology	0.25um CMOS 4-poly and 3-AL	

Whereas I described DUP can maximize the write data transferring capacity, that is not correct in actual. Because true maximized data capacity is rather provided by memory cells in the place of sense-amplifier as described in Figure 3.10. Following chapter discusses the use of such true maximized bandwidth, which every memory cells function simultaneously.

References

[15] K. Suizu, T. Ogawa, and K. Fujishima, "Emerging memory solutions for graphics applications," IEICE Trans., Electron, Vol.E78-C, No.7, pp.773-781 (Jul. 1995).

[16] K. Dosaka, K. Konishi, K. Hayano, A. Yamazaki, C. Hart, M. Kumanoya, H. Hamano, "A 100-MHz 4-Mb Cache DRAM with fast copy-back scheme," IEEE Journal of Solid-State Circuits, Vol.27, No.11, pp.148-149 (1992).

[17] S. Tanoi, Y. Tanaka, T. Tanabe, A. Kita, T. Inada, R. Hamazaki, Y. Ohtsuki, and M. Uesugi, "A 32-bank 256Mb DRAM with Cache and TAG," IEEE ISSCC94, Dig., pp.144-145 (1994).

[18] K. Inoue, H. Nakamura, and K. Kawai, "A 10M-bit Frame-buffer memory with Z-Compare and A-Blend units," IEEE Journal of Solid-State Circuits, Vol.30, No.12, pp.1563-1568 (Dec. 1995).

[19] A. D. Booth, "A Signed Binary Multiplication Technique," Quarterly Mechanical Applications in Math., Vol.4, part 2, pp.236-240 (1951).

[20] C. S. Wallace, "A Suggestion for a fast multiplier," IEEE Trans. Computer, Vol.13, No.2, pp.14-17 (Feb.1964).

[21] A. Glassner, K. Turkowski, "Graphics Gems," Academic Press, pp.147-165 (1990).

[22] K. Inoue, H. Abe, K. Mori, and S. Fukagawa, "A Low-voltage 42.4G-Bps Single-ended RMW bus and Programmable Page-size on a 3D frame-buffer," IEICE Trans., Electron, Vol.E83-C, No.2 (Feb. 2000).

[23] R. Torrance, I. Mes, B. Hold, D. Jones, J. Crepeau, P. DeMone, D. MacDonald, C. O'connel, P. Gllingham, R.White, S.Duggins, and D. Fielder, "A 33GB/s 13.4Mb integrated graphics accelerator and frame-buffer," ISSCC'98 Dig. of Tech., pp.340 (1998).

[24] A. Yamazaki, T. Fujino, K. Inoue, I. Hayashi , H. Noda, N. Watanabe, F. Morishita, K. Dosaka, K. Arimoto, S. Wake, K. Fujishima, and H. Ozaki, "A 0.18um 32Mb Embedded

DRAM Macro for 3D Graphics Controller," IEICE Trans., Vol.E85-C, No.9, pp.1697-1708 (Sept. 2002).

Chapter 4
Content addressability as further enhanced effective bandwidth capability

4.1 Introduction

The benefit of content addressability is derived from the utilization of maximized data bus with memory. Every internal memory cell comprises of special integration to carry out a comparison functions between storage data and search request data in parallel. Such memory, where functional integration into memory cells is used to perform content addressability functions, is called content addressable memory (CAM). [25,26] I will first describe how this comparing function is performed by memory cell with the explanation of physical cell structure.

Functionally, a CAM can be classified into binary CAM (BCAM) which single bit CAM cell indicates binary states "0" and "1", and ternary CAM (TCAM) which single bit CAM cell indicates ternary states "0", "1" and "x" means don't care. Nowadays, application prefers TCAM rather than BCAM in terms of the benefit of performance per cost. This chapter examines that performance advantage with reference to the cost of physical structure for each. As further development of this research, the discussion of CAM based application specific VLSI has started from the network application because it manages the search operation against a large database much more often than other applications. Similar to the physical TCAM analysis, I will examine the performance benefit of CAM based search operation with respect to the actual application. I will address in the application of signature-matching, which is one of network applications, in comparison with conventional tree algorithm.

4.2 Physical structure performing binary CAM and ternary CAM functions

CAM is capable of fast search in addition to the other standard memory functions: write and read with of course storing data. Following section explain how search operation is conducted in accordance with physical cell structure of binary CAM and ternary CAM described in 4.2.1 and 4.2.2, respectively.

4.2.1 Conventional CAM, binary CAM (BCAM)

The physical structure of single bit CAM cell and an array architecture are shown in Figure 4.1.

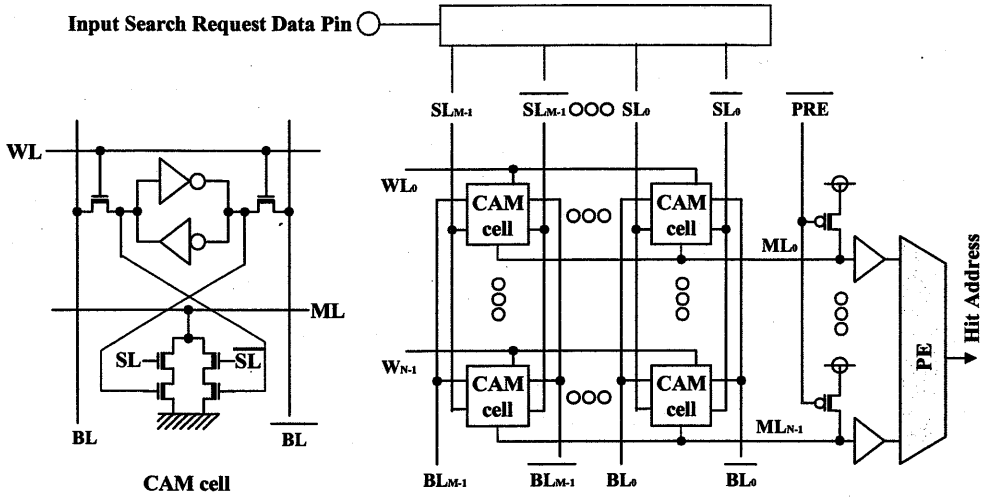


Figure 4.1 CAM cell and array

The search operation is taken place in the match comparing function between the datum given at input search request data pins and CAM's internal datum arranged horizontally. Each CAM cell consists of a search line (SL) pair and a match line (ML), as well as a bit line (BL) pair and a word line (WL). The plurality of CAM cells arranged horizontally represents a single entry address with sharing a common ML. Storage data is written with BL and WL, similar to other memories. Search request datum is transferred from the input pins to the corresponding SL pair, and ML transmits the output of the search result. When a stored datum in the entry matches with the searched datum, the previously pulled-up ML keeps its high level. On the other hand, ML goes low when stored datum does not match with the searched datum. As a single ML is shared by the number of CAM cells within an entry, the actual comparison task is done between plurality of search request data pins [n:0] and plurality of CAM cells [n:0] in the place of single bit comparison. As every CAM cell simultaneously participates in the search operation, the bandwidth provided by memory is

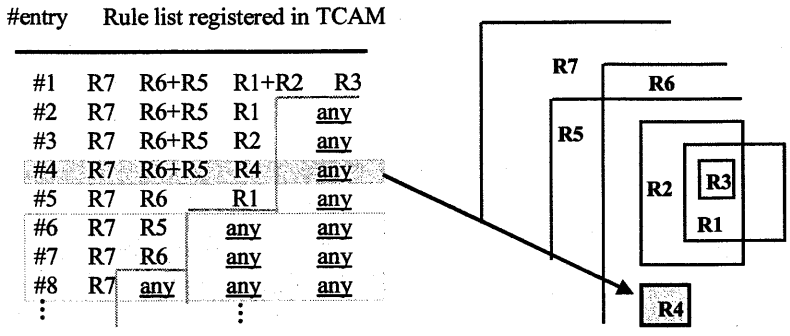
certainly maximized as indicated by the entire bit capacity times the operating frequency, and that is the benefit of content addressability. Also, when multiple entry addresses in CAM are matched to the search request datum at the same time, then the implemented encoder placed in conjunction with CAM array resolves the upper address as the final winner, hence it is typically named Priority Encoder (PE).

4.2.2 Ternary CAM (TCAM)

Recent applications rather prefer the feature of ternary states than binary value stored in each cell. CAM's internal data should consist of "0", "1", and "x" means "don't care". The following example of signature-matching application will illustrate the advantage of ternary states. In the application of network security, signature-matching is a valuable method to detect packets that contain some bad viruses. For example, when detecting the word "apple", the additional similar word "Apple" should also be noticed. The difference of small "a" and capital "A" can be identified by ascii-code "61" and "41" respectively. Both "a" and "A" can be detected by the single stored pattern "01x0 0001", instead of two different patterns "0100 0001" and "0110 0001". Thus, described application prefers the ternary CAM with the cost advantage of saving the memory space stored.

Another example emphasizing on the benefit of ternary states is viewed in the application of packet classification. As described in Figure 4.2, its general classification manner is referred to the hierarchically established rule, named Longest Prefix Match. [27,28]

In the sense that the priority of a rule becomes higher if it is more specific, those rules are arranged in the TCAM in decreasing order of priority. The most specific match involving the longest matched pattern is finally chosen. Although other entries at addresses #6, #7, and #8 are also matched due to the contained 'don't care' positions in the example, the upper most specific address #4 is finally chosen by the previously described function of priority encoding.



any means 'x' data written in TCAM.
The search result is always 'match'

Figure 4.2 Hierarchically established rules in TCAM

I will now return to the physical hardware structure to illustrate the function of TCAM. Single bit TCAM cell consists of a comparing circuitry, which is placed between two storage cells as shown in Figure 4.3.

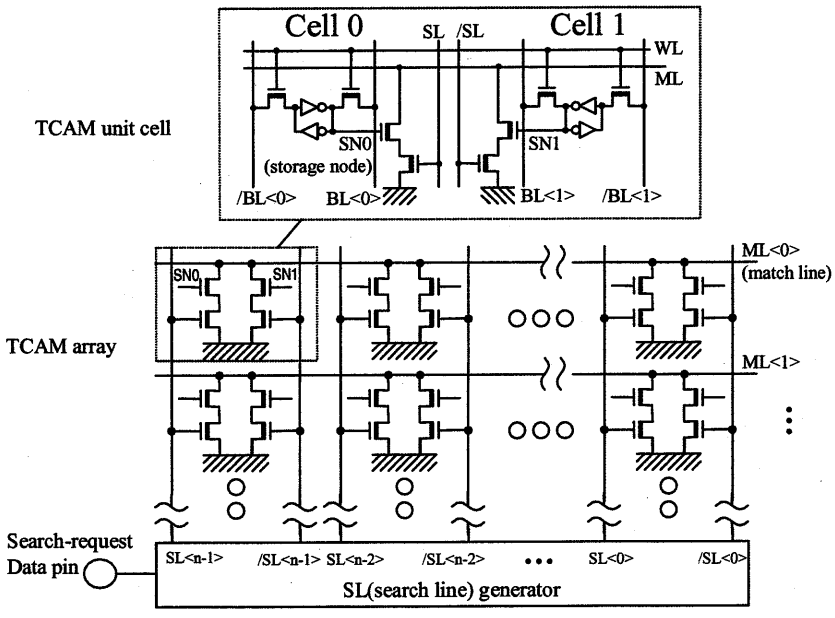


Figure 4.3 Ternary CAM cell and Array

Functional storage data "0" means that Cell_0 stores High and Cell_1 stores Low.

When Search data is "0", which means SL is Low and /SL is High, the result is "match", hence ML isn't discharged.

When Search data is "1", which means SL is High and /SL is Low, the result is "miss", hence ML is discharged.

Functional storage data "1" means that Cell_0 stores Low and Cell_1 stores High.

When Search data is "0", which means SL is Low and /SL is High, the result is "miss", hence ML is discharged.

When Search data is "1", which means SL is High and /SL is Low, the result is "match", hence ML isn't discharged.

Functional storage data "x" means that Cell_0 stores Low and Cell_1 stores Low.

This is the "don't care" state. Regardless of search data, the result is always "match". Hence ML isn't discharged by any search data.

Note that it might not correct to use "bit" to count TCAM unit cell, since TCAM involves three states instead of two states. Actual cost comparison in terms of physical hardware structure is ten-transistors and sixteen-transistors in the binary CAM cell and in the ternary CAM cell, respectively. Although ternary CAM states functional cost advantage as 50% cheaper, i.e. "apple" and "Apple" can be compacted to the single pattern as described, the physical overhead costs 60% more. Therefore the fair cost advantage of ternary CAM should be corrected to 1.6/2=80% of binary CAM. Note that the described sixteen-transistors TCAM cell is also 2.5x more expensive than a single bit SRAM cell, which consists of only six-transistors. It should therefore provide at least 2.5x performance gain by TCAM. The performance advantage of CAM in comparison with the conventional search method using read operation of SRAM, will be examined in following section.

4.3 Search operation and the application of signature-matching

A search is one of the expensive operations for hardware. Moreover it is now in the serious performance bottleneck because the database is getting larger and larger as viewed in the worldwide internet population. The conventional search performance has been indicated

by the entire data size divided by capable bandwidth during read operation, hence the performance of search is reduced with the growth of database size. On the other hand, CAM keeps the performance constant regardless the database size. This section described the conventional search method using the tree algorithm with the read operation of memory in particular application in detail.

Despite improved network infrastructure, which is now an essential part in our daily life, the network infection with fast increasing number of useless packets is real serious concern. These useless packets are classified into anomalous packets or misused packets. Anomalous packets can paralyze the network traffic with an uncommonly huge data size and can lead to the Denial-of-Service (DoS) effects, while misused packets can intrude the network services with various kinds of viruses. Since an efficient and reliable security policy for coping with this problem is an urgent task, public and private organizations such as the Open Source Network Intrusion Detection System (from Snort™) [29] are working on the methods for useless-packet detection and removal to secure the network traffic. A typical misused packet with some virus can be identified by a particular combination of network address, port number, and specific signature. Such knowledge is applied in conventional software-oriented virus detection like firewalls. However, due to the increasing network speed and the increasing number of misused and anomalous packets, dedicated hardware named Intrusion Detection Systems (IDS) have appeared. An IDS uses the entire list of identification data for known types of useless packets and checks each incoming new packet. The detection of a misused packet is carried out by searching for a specific signature within the packet, which is known to be the characteristic for this type of useless packets. Table 4.1 shows examples of misused packets together with the information necessary for identification, which is freely downloadable from Snort™.

Table 4.1 Examples of identification data for useless packets

Snort ID	TCP/UDP source	TCP/UDP destination	Specific signature	Detection task
SID 512	Port 5631, 5632	any	Invalid login	PC remote login fail
SID 709	any	Port 23	login¥:root	Telnet root restriction
SID 1293	Port 139	any	|00|.|00|E|00|M|00|L	Nimda virus
SID 805	Port 80	any	/wsisa.dll/Wservice= && WSMadmin	Administration access

This information is typically given by the combination of a source- and/or destination-port number specified in the packet header and an additional specific signature. As described, conventional detection task such as firewalls were on the basis of software programs, and such software solutions are becoming a serious performance bottleneck Moreover, the detection task of useless packets is more complicated, because it requires a search through the complete payload. We commonly call the part of a packet, which contains the routing information through the network (MAC, IP, TCP/UDP) 'header', and the other part, which involves the contents 'payload'. While the search through the header is relatively easy due to the header's specified structure and its small number of bytes, the search through the packet's payload is quite difficult because the specific signature may be hidden at any position within the data.

4.3.1. Conventional search operation with tree algorithm

Figure 4.4 shows the example of specific-signature search with the tree algorithm [30]. Characters are expressed with bytes in ASCII-code, which uses 128 combinational possibilities (0x00 to 0x7f in hexagonal notation) and reserves the remaining 128 possibilities for other purposes. A single bit-tree unit consists of a comparator and two output paths, i.e. OUT_0 is selected if the comparing result with data-zero is matched while OUT_1 is selected if the comparing result with data-one is matched. With this single-bit unit, an 8-bit

tree can be constructed and can be applied to identify one of 128 characters. The 8-bit tree unit is named 'byte-tree' and can be similarly used to construct a 16-byte tree to identify a specific signature with 16 characters length. The concept of tree algorithm realizes a sequential check with match comparing.

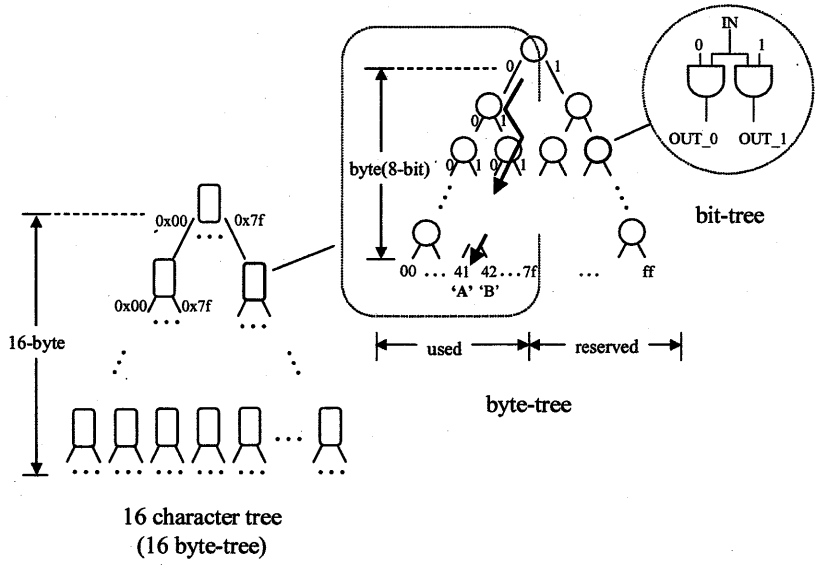


Figure 4.4 Tree algorithm for 16-character search

Total tree-unit size to search for 16-character signatures is given by the formula $Size = \sum_{n=0}^{15} 128^n \bullet \{(\sum_{m=0}^{7} 2^m)/2\}$. As this size is too large to be built completely in hardware, a common compromise is to use a byte-tree unit in hardware, and supply the additional information of the searched signatures by reading 128-bit data words from an outside memory.

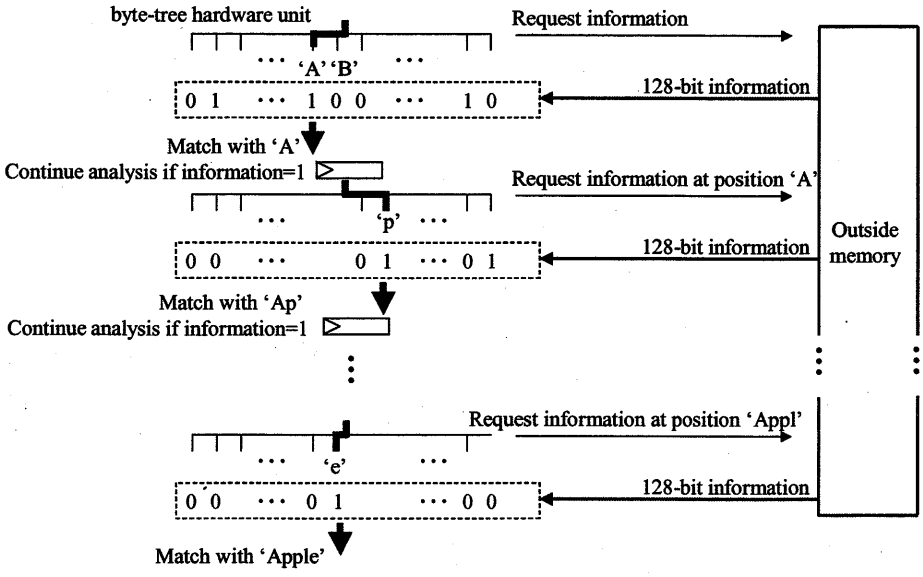


Figure 4.5 Conventional pipelined signature search based on tree algorithm

Figure 4.5 shows a conventionally applied example of pipelined signature search based on described tree algorithm. Each of the 128-bit data corresponds to 1 of the 128 characters, which may or may not match to a specific signature in the 16-character string at the currently checked position. Namely, a logical 1 means that the character maybe appearing in one of the searched-for specific signatures, while a logical 0 means that none of the specific signatures continues with this character at the currently checked position. Therefore, once a 0 appears during the sequential check of a 16-character string, it is known that this string doesn't match to any of the searched-for signatures and further processing of this string can be stopped.

In fact, it is also difficult to store the 128-bit data words for all positions in the virtual tree, i.e. for $\sum_{n=0}^{15} 128^n$ positions. The information data typically stored in the outside memory is at the positions where the result of the sequential check keeps logical 1, while the information data is no longer stored once logical 0 appears. Advantages of the described sequential search with a byte tree are an easy search stop with the appearance of logical 0 at a certain character position and an easy reconfiguration feature in case of newly appearing signatures because the specific signatures of useless packets are updated so often. Since it is completely unknown where a specific signature is hidden in the packet payload, a byte-shifted pipeline

is a desired feature to examine the payload in character-by-character manner. However, a serious problem of this signature search is an extremely high access bandwidth required for the outside memory. In the example of a search for 16-character signatures, which uses 16 pipeline stages with byte trees, each stage may need a 128-bit data word from the outside memory in each clock cycle. At an operating frequency of 50MHz, which corresponds to 50M-Bps (400M-bps) signature-search speed, the required bandwidth for the outside memory is already up to 12.8G-Bps (128-bit x 16-pipeline x 50MHz). Although 400M-bps signature-search speed is below today's Ethernet speed of 1G-bps, 12.8G-Bps memory bandwidth is a big challenge for existing memories.

4.3.2 CAM based search with its advantages and problems

The example shown in Figure 4.6 introduces the concept of CAM based search operation in the application of signature-matching comprising 8-byte length for signatures. Character strings of 8-byte length are extracted from the packet contents and are compared to the signature database registered in the CAM. The strings are continuously extracted in byte-shifted manner until the end of the packet contents is reached. [31,32] The difficult problem in the virus detection is that it is not easy to know the position of specific signature in the packet contents. As a virus signature maybe hidden anywhere, a simpler search within a restricted range is not sufficient. Described concept is capable of virus detection hidden anywhere with CAM's performance advantage. The performance level described in the previous section would require 50M-sps, which is already possible with CAM-parts presently available on the market. Therefore, it is likely that TCAM based signature matching is easier and can replace the conventional tree algorithm. However, further serious problems as discussed below should be addressed and fixed.

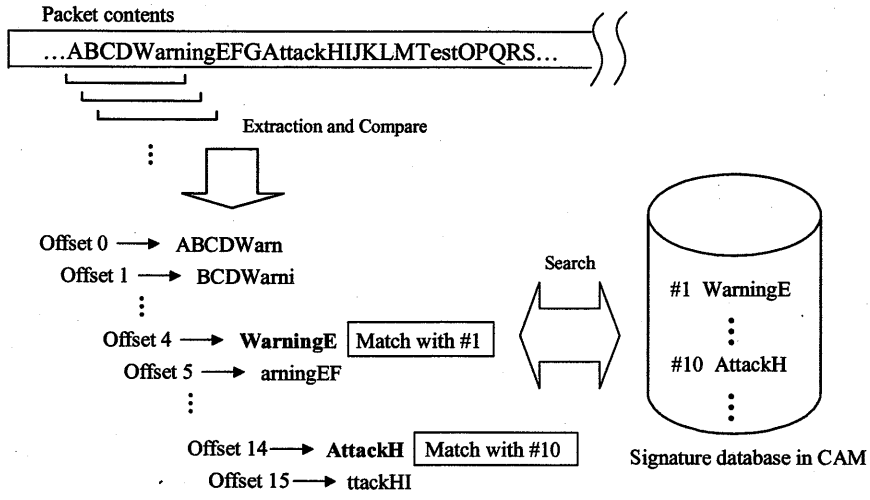


Figure 4.6 Concept of CAM based signature search

High bandwidth with the cost of power dissipation

CAM's internal bandwidth during the search operation sums up to total storage capacity times search speed, e.g. 6.25T-Bps in case of 1Mb CAM with the performance of 50M-sps. There is no doubt that bandwidth capacity provided by CAM is much higher than other commodity memories, however this performance comes at the cost of power dissipation. Typical power dissipations of CAM production part at the search speed of 50M-sps are approximately 2W and 6W with storage capacity of 1Mb and 4.5Mb, respectively. In recent years, a considerable number of studies have been conducted on methods for CAM's power reduction. Proposed methods are to eliminate extra comparing operations by means of a pre-search [33] as well as the minimization of the number of ML pre-charges [34]. Also, a special sensing technology [35] and a special swing [36] have been proposed for ML. Unfortunately, the contribution from these research results isn't visible on the market yet. Here we report additional power-reduction methods, which are driven by the special properties of the signature-matching application.

Lack of external bandwidth

Signature-matching with a specific network address and/or port number is surely useful for

virus detection. However, the problem is the search-request data length is typically long. For example, in case of the combination of 32-bit for the address and 32 characters for the signature, it becomes necessary to supply 288 bits of search-request data to the CAM in each search cycle. The actual search performance of the CAM can often be limited due to the lack of external I/O bandwidth.

Complicated lookup method

Today's detection of useless packets is getting more and more complicated. Sometimes even multiple-matches with various signatures and/or a magnitude-compare operation for the network address are needed. Since presently available CAM is just capable of an exact match, additional developments and research efforts are desired for this kind of further application.

These problems, the huge power dissipation in particular, also gives us a serious question with reference to various scaling factors. Both the advantage of fast search and the problem of power dissipation greatly depend on the number of hardware elements on the silicon functioning simultaneously, and therefore the technological scaling factors only drive power in the increase direction. This serious problem with scaling will be investigated prior to the discussion of high bandwidth specific VLSI. Following chapter describes the experimental study to ensure the scalability.

References

[25] F. Shafai, K. J. Schults, G.F.Randall Gibson, A. G. Bluschke, and D. Somppi, "Fully Parallel 30MHz, 2.5Mb CAM," IEEE Journal of Solid-State Circuits, Vol.33, No.11, pp.1690-1696 (Nov. 1998).

[26] K. J. Schults, and P. G. Gulak, "Fully Parallel Integrated CAM/RAM Using Pre-classification to Enable Large Capacities," IEEE Journal of Solid-State Circuits, Vol.31, No.5, pp.689-699 (May 1996).

[27] P. Gupta and N. McKeown, "Algorithm for Packet Classification," IEEE Network, pp.24-32, (Mar./Apr. 2001).

[28] M. Uga and K. Shiomoto, "A longest Match Table Look-up Method Using Pointer Cache," IEICE Trans. Communication, Vol.E84-B, No.6, pp.16640-1673 (Jun. 2001).

[29] http://www.snort.org

[30] Nikkei Network, pp.65 (June 2001).

[31] H. Yamada, M. Hirata, H. Nagai, and K. Takahashi, "A high-speed string-search engine," IEEE Journal of Solid-State Circuits, Vol.SC-22, pp.829-834 (Oct. 1987).

[32] H. Yamada, Y. Murata, T. Maeda, R. Ikeda, K. Motohashi, and K. Takahashi, "Real-time string search engine LSI for 800-Mbit/sec LANs," CICC Dig., pp.21, (May 1998).

[33] C-S. Lin, J.-C. Chang, and B.-D. Liu, "A Low-Power Pre-computation-Based Fully Parallel Content-Addressable Memory," IEEE Journal of solid-state circuits, Vol.38, No.4, pp.1512-1519 (Apr. 2003).

[34] H. Miyatake, M.Tanaka, and Y. Mori, "A Design for High-Speed Low-Power CMOS Fully Parallel Content-Addressable Memory Macros," IEEE Journal of solid-state circuits, Vol.36, No.6, pp.956-968 (Jun. 2001).

[35] I. A. T. Chandler, and A. Sheikholeslami, "A Ternary Content-Addressable Memory (TCAM) Based on 4T Static Storage and Including a Current-Race Sensing Scheme," IEEE Journal of solid-state circuits, Vol.38, No.1, pp.155-158 (Jan. 2003).

[36] K. Pagiamtzis and A. Sheikholeslami, "Pipelined Match-Line and Hierarchical Search-Lines for Low-Power Content-Addressable Memories," IEEE Custom Integrated Circuits Conference, Dig., pp.383-386 (2003).

Chapter 5
Experimental study of content addressability integration onto memory

5.1 Introduction to the previous state of art

So far, I have outlined features and benefits of fast search provided by CAM with network applications, whereas I encountered the difficulty of power dissipation when I aim to develop the discussion toward CAM based specific VLSI. Although there are many reports on application specific integrated circuits, which are based on DRAM or SRAM cores (AS memories), reports on CAM-based are very rare. In my view, the main reason for this situation is closely related to the issue of power dissipation. I have therefore prioritized the work for the minimization of power in CAM core. Allowing the scaling issue of power, I conducted an experimental study to examine the power reduction technology as well as the qualification of fast search before developing the discussion of CAM based application specific VLSI. In the experiment, SRAM based CAM cell and DRAM based CAM cell [37,38] are implemented together to evaluate each individual benefit. A good place to start my power reduction technology in this experiment is a hierarchical pipelined partial search. [39-41] In fact, I also expected the additional benefit, that is high-speed operation can be carried out by the pipelined architecture. Despite the examination of power, a quite unexpected defect problem has arisen in this experiment, in which the defect rate of CAM was approximately 2x higher than SRAM, although they are placed on the same chip silicon space. That deserves careful attention as serious negative scaling, because both the cost of power and the cost defect seem to keep increasing with the growth of CAM density driven by various scaling factors performed on the silicon surface. The primary consideration should be turned to make such scaling factors of the content addressability more reliable. This chapter limits the discussion in the experiment with its evaluation. Further technical proposals related the cost of power and the cost of defect are discussed in chapter 6 and chapter 7, respectively.

5.2 Hierarchical pipelined partial search

A major contribution to power reduction can be carried out from previously proposed pipelined search. Figure 5.1 shows the block diagram and the time chart of that power

reduction scheme, where pipelining structures are implemented in ML direction as well as in SL direction. The concept is to forward the search process from one pipeline stage to the next stage only for MLs which resulted a match in the previous stage. Furthermore, if not-matching results occurred on all MLs of the previous sub-array, the SLs of the sub-array of the following pipeline stage are completely deactivated for the purpose of further power saving.

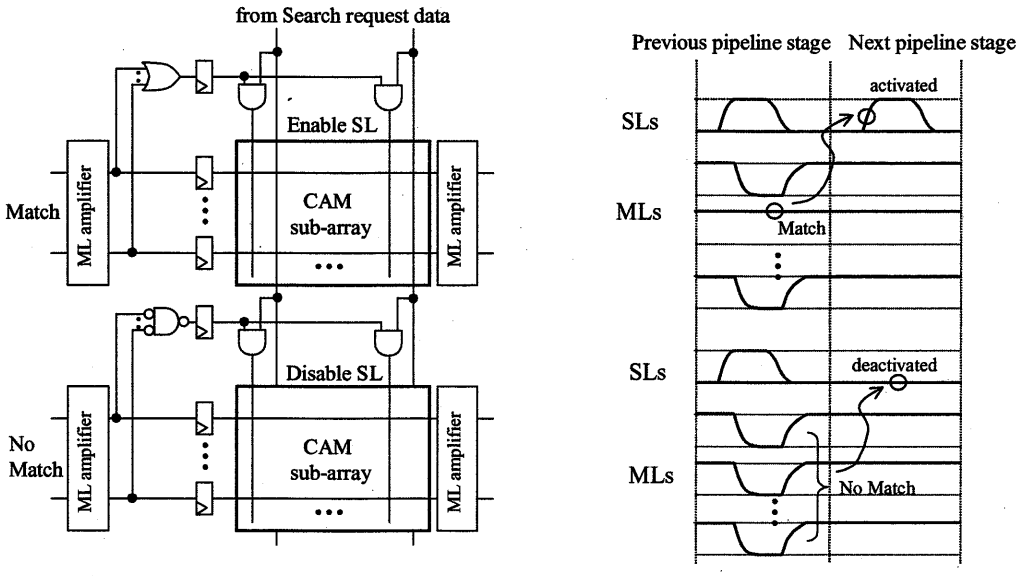


Figure 5.1 Hierarchical pipelined search concept

Figure 5.2 illustrates more detailed ML control scheme in the sub-array. Conditional ML charging has to be applied, which depends on the search result in the previous pipeline stage. Note that all ML are reset to zero-level in all pipeline stages after each clock cycle. The point is that ML reset is inevitable since remaining high-levels from a match on a sub-array ML can falsify the search result in the next clock cycle. For example, when all MLs in a particular sub-array are missed, the SLs of the next pipeline stage are totally deactivated as described in the timing chart of Figure 5.1. Without an overall ML reset, the next pipeline stage would completely lose the function to reset its MLs. Therefore high levels would remain mistakenly on MLs where the search result in the previous clock cycle was a match.

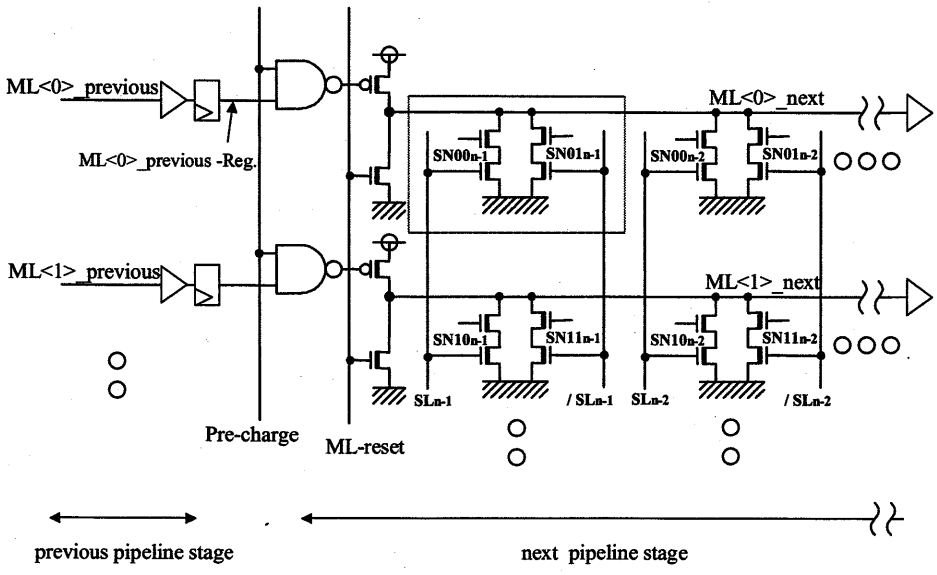


Figure 5.2 Example of ML control in a conventional pipelined search

Figure 5.3 explains the overall ML-reset with timing charts for the miss case (Fig. 5.3-a) and the match case (Fig. 5.3-b) and highlights additionally power dissipation by shaded areas.

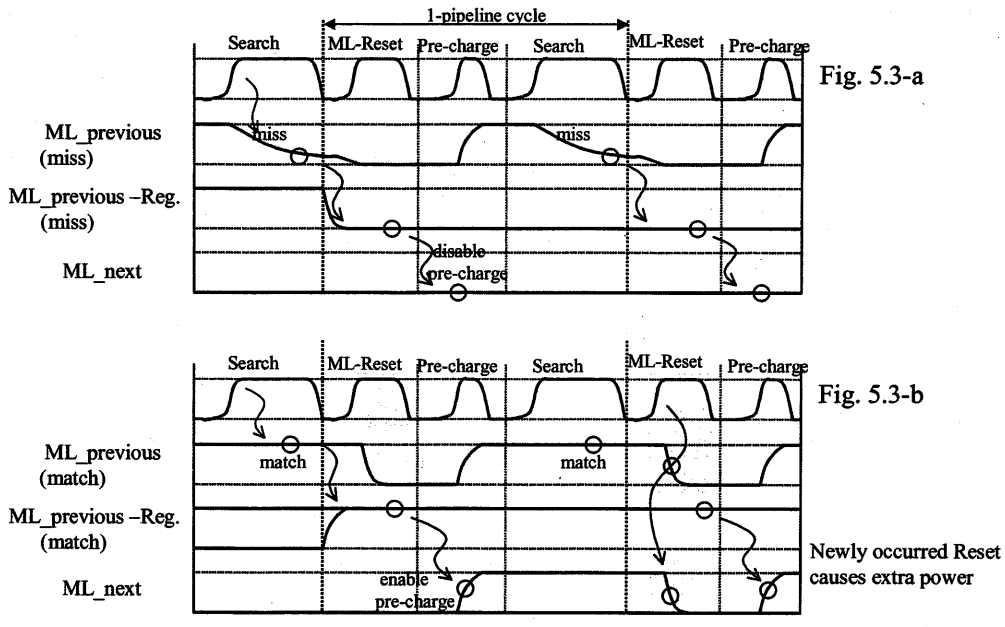


Figure 5.3 Time charts of a conventional pipeline for (a) the miss case and (b) the match case

In the conventional non-pipelined search, MLs stay at high-level if the search result is a match, hence no power is consumed. On the other hand, MLs in each sub-array of the pipelined search have to be discharged every clock cycle to avoid falsified results.

In the evaluation of the experiment, extra power dissipation is notable especially in the case that large numbers of matching results are forwarded to the next pipelining stage. Consequently, the main power consumption in pipelined search occurs in the CAM area with matches, while the main power consumption for conventional non-pipelined search occurs in the CAM area with misses. Due to these considerations with that experimental evaluation, an application with a high probability of large number of multiple matches will not benefit much from this pipelined search power reduction scheme. Unfortunately, the desired feature of ternary states in a CAM definitely increases the number of multiple matches. This is also evident in the longest prefix match described in the previous hierarchically established rule in previously described Figure 4.2

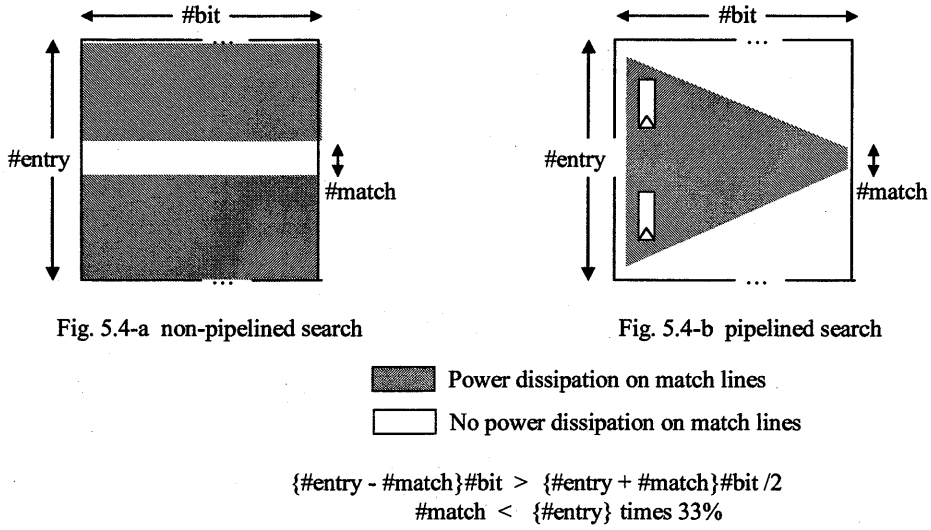


Fig. 5.4-a non-pipelined search

Fig. 5.4-b pipelined search

Figure 5.4 Relation between power dissipation and number of multiple matches

Figure 5.4 explains a simple geometrical estimate of the CAM areas where power dissipation occurs under multiple matches for the conventional non-pipelined (Fig. 5.4-a) and pipelined (Fig. 5.4-b) search schemes. According to Fig. 5.4-a, the area of power dissipation for the conventional non-pipelined search is proportional to {#entry - #match} times #bit. On the

other hand, Fig. 5.4-b shows that if the match area is approximated to reduce linearly with proceeding pipeline stages, the power-dissipation area for the pipelined search becomes {#entry + #match}/2 times #bit. Accordingly, the active-area comparison suggests that the conventional pipeline concept can be expected to offer a reduction of power dissipation only if the number of matches is less than one third of the number of entries in the CAM, i.e. #match should be less than #entry/3. Actually, the circuit overhead for pipelining must be included in this estimate, hence the critical number of matches should be considerably lower than one third of the CAM-entry number. Despite the expectation of double benefits in the hierarchical pipelined partial search scheme, one is power saving and the other is fast search, I noticed that further improvement work is desired.

5.3 Comparison of DRAM based CAM cell and SRAM based CAM cell

Another evaluation conducted in the experimental study is the characterization of DRAM based CAM. Similar to the comparison between DRAM cell and SRAM cell, experimental DRAM based CAM aims to prioritize the small cell size. A DRAM based cell consists of six transistors and two capacitors. Two transistors and two capacitors are responsible for the storage of ternary states, in actual four possible states, and two serial transistors perform match comparing with the search-line data.

It is previously known that serial transistors seriously affect the cost advantage carried out from DRAM structure. Therefore, the technology used in this experiment is to hide that area overhead caused by serial transistors under the storage capacitor as shown in Figure 5.5.

While the DRAM based CAM cell area performed on the experimental chip is 3.5um2, which is approximately 60% of a SRAM based design, typical area reduction provided by DRAM structure should be 80% smaller than that of SRAM. Honestly, the given contribution number of 60% is unfortunately doubtful to emphasize the advantage of DRAM based CAM cell.

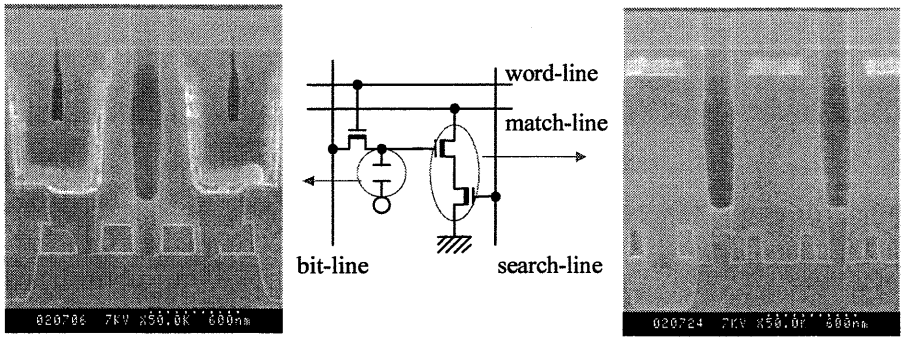


Figure 5.5 DRAM based CAM

An additional functional difficulty of DRAM based CAM in comparison with SRAM based CAM is the need of refresh cycle. [42] Since the electrical charge dynamically stacked on the storage capacitor is unavoidable to be reduced and is finally disappeared, and this electrical charge is also used for gate level voltage at the search transistor, therefore additional concern related the refresh operation arises in DRAM based CAM. General DRAM refresh cycle is managed by the sense-amp to recover the high-level for each cell.

Bit-lines are first equalized, and then word-line asserts for the need of refresh. The electrical charge in a particular cell is transferred to bit-line and the sense-amp re-stores as complete high-level. In this experimental evaluation, it takes several nano-seconds for the completion of refresh, and this refresh cycle is required every hundred nano-seconds. Consequently, the refresh cycles required in DRAM based CAM reduce the search performance by the factor of several percents.

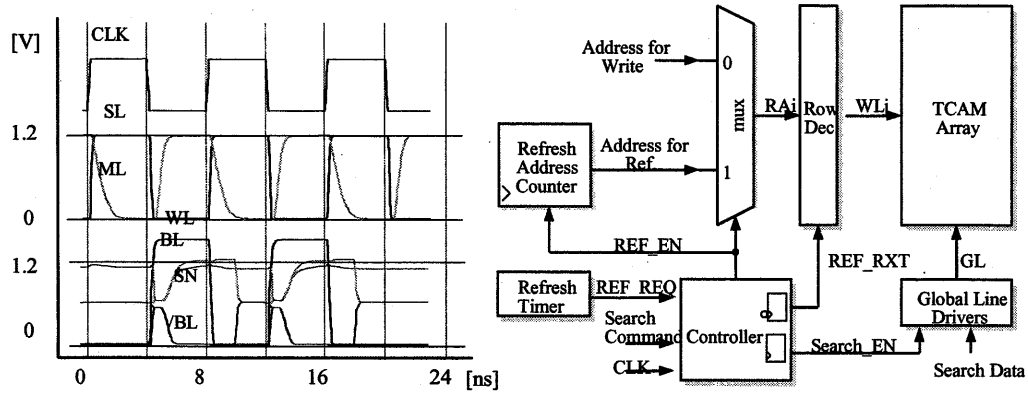


Figure 5.6 Proposed transparent schedule to hide refresh

A Transparently Scheduled Refresh is proposed and verified later. [43] This concept is not to manage the timing of search and refresh simultaneously, but to schedule each event separately. Since the duration of search operation can be divided into the event of pre-charge period and the search period, and the voltage level of storage cell is only referred during the search period, the refresh operation can be allowed to schedule in the duration of pre-charge as shown Figure 5.6.

5.4 Unsolved serious problems related the defect rate

The experiment with its evaluation revealed that the examined hierarchical pipelined partial search is not reliable to apply to real application specific VLSI because the contribution of power saving depends too much on the application's handling of multiple matches. Additional improvement work is desired, and that is described in chapter 6. The examined cell-area reduction carried out by DRAM based CAM structure is 60%, that number unfortunately loses fundamental benefit of DRAM based structure.

However, more serious problem occurred in the experimental evaluation was the defect rate.

Although various memories were placed on the same silicon, the defect rate of CAM was higher than other circuitries. As shown in Figure 5.7, the difference between SRAM and CAM was notable when SRAM yield less than 60%.

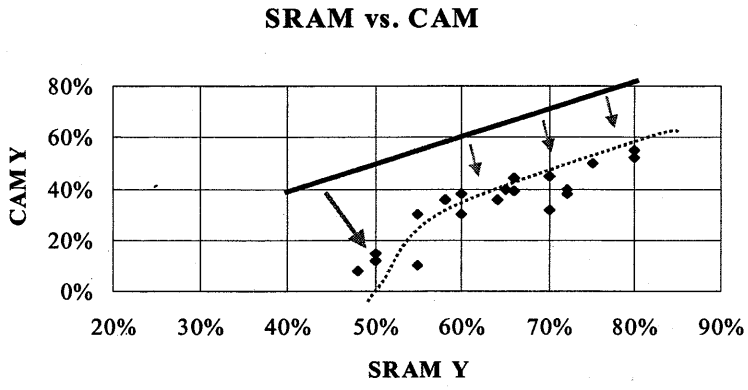


Figure 5.7 Test yield difference between SRAM and CAM

For example, when SRAM yield was 60%, the yield of CAM was approximately 40%. When SRAM yield was 60% to 50%, the yield of CAM was 40% to 10%, that was almost one half or less of SRAM yield.

This is a serious problem despite the advantage of high bandwidth content addressability. Conducted defect analysis with respect to the physical structure of content addressability and further improvement work are described in chapter 7.

References

[37] J.P.Wade and C.G.Sodini, "Dynamic cross-coupled bitline content addressable memory cell for high density arrays," IEDM Tech. Dig., pp.284-287 (Dec. 1985).

[38] T. Yamagata, M. Mihara, T. Hamamoto, T. Kobayashi, and R. Kasai, "A 366k-bit fully content addressable memory using stacked capacitor cell structure," IEEE Journal of Solid-State Circuit, Vol.27, No.12, pp.1927-1933 (Dec. 1992).

[39] S. Hanzawa, T. Sakata; K. Kajigaya, R. Takemura, and T. Kawahara, "Pipeline Dynamic CAM based on One-Hot-Spot Block Code for Use in Network Router," Technical Report of IEICE SDM2004-120, ICD2004-62, pp.1-5 (2004).

[40] K. Pagiamtzis and A. Sheikholeslami, "Pipelined Match-Line and Hierarchical Search-Lines for Low-Power Content-Addressable Memories," IEEE Custom Integrated Circuits Conference, pp.383-386 (2003).

[41] K. Pagiamtzis and A. Sheikholeslami, "A Low-power Content Addressable Memory (CAM) using Pipelined Hierarchical Search Scheme," IEEE Journal of Solid-State Circuits, Vol.39, No.9, pp.1512-1519 (Sept. 2004).

[42] H. Noda, K. Inoue, M. Kuroiwa, F. Igaue, K. Yamamoto, H. J. Mattausch, T. Koide, A. Amo, A. Hachisuka, S. Soeda, I. Hayashi, F. Morishita, K. Dosaka, K. Arimoto, K. Fujishima, K. Anami, and T. Yoshihara, "A cost-Efficient High-Performance Dynamic TCAM With Pipeline Hierarchical Searching and Shift Redundancy Architecture," IEEE Journal of Solid-State Circuits, Vol.40, No.1 pp.245-253 (Jan. 2005).

[43] H. Noda, K. Inoue, H. J. Mattausch, T. Koide, K. Dosaka, K. Arimoto, K. Fujishima, K. Anami, and T. Yoshihara, "Embedded Low-Power Dynamic TCAM Architecture with Transparency Scheduled Refresh," IEICE Trans. Electron, Vol.E88-C, No.4, pp.622-629 (Apr. 2005).

Chapter 6
Newly developed power reduction technologies for CAM

6.1 Introduction to the previous state of the art

There is no doubt that content addressability performed on memory effectively boosts the bandwidth capability during search in place of conventional read operation with Tree algorithm. It was pointed out in previous chapter, however this performance comes with the huge cost of power dissipation and also additionally noticed high defect. I limited this chapter to the discussion of technologies related to the cost of power, while the discussion related the cost of defect is described in chapter 7.

Neither minimization of transistors nor core voltage reduction provided by the scaling can solve the problem of power in CAM. Referring to the power with technological scaling factors as shown in table 6.1, it unfortunately drives the power dissipation to the direction of increase much more. Likewise, the actual production data in the table shows unreliable scaling despite the expectation of content addressability as a candidate of future high bandwidth solution.

Table 6.1 Technology trend and the Scaling of Power

	150nm tech.	130nm tech.	90nm tech.	*Scaling ratio*
CAM density	4.5Mb	9Mb	18Mb	$N = N_0 \cdot 10^{0.15year}$
Core Voltage	1.5V	1.2V	1.0V	$V = \sim.83 \cdot V_0$
Scaling ratio	*x0.80*	*x0.83*		
Performance	83Msps	100Msps	125Msps	$F = \sim 1.25 \cdot F_0$
Scaling ratio	*x1.23*	*x1.25*		
Others technology parameters				
Area(cell) ratio	*x0.7*	*x0.7*		$A = \sim.7 \cdot A_0$
C(1/tox) ratio	*x1.3*	*x1.3*		$C = \sim 1.3 \cdot C_0$
				$Q = N \cdot A \cdot C \cdot V$
				$P = Q \cdot V \cdot F$
Actual production Measured Power	~3.9W	~5.8W	~9.1W	↓
Scaling ratio	*x1.6*	*x1.6*		→ **$P = \sim 1.6 \cdot P_0$**

I have therefore conducted to achieve the reliable scaling in terms of power for the content addressability first. This chapter discusses and proposes power reduction technologies with an improved hierarchical pipelined partial search based on the experiment described in previous chapter. Over and above that, 2-bit encoded cell and flexible internal partitioning are further power reduction technologies directed to the application usage.

6.2 Improved hierarchical pipelined partial search

It is necessary to make an additional proposal over the conventional pipelined search, especially for the case of multiple matches. As described previously, not only signature matching but most major applications of TCAM commonly manage a large number of multiple matches. Figure 6.1 illustrates an improved pipelined-search concept for 2 pipeline stages. Instead of the conventional simple forwarding concept, the final match is generated by a logic AND function of both the 1st stage and 2nd stage match result.

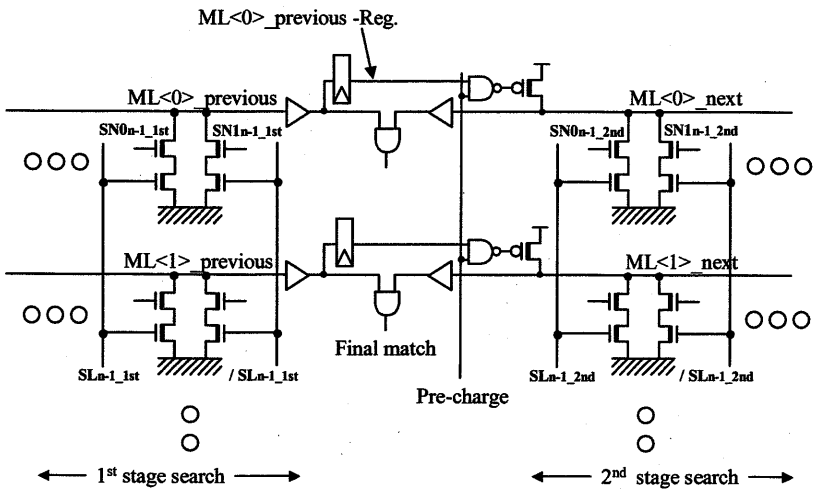


Figure 6.1 Proposed circuitry for pipelined search with further power reduction

As shown in the time chart Figure 6.2, the proposed concept saves power dissipation by the removal of the overall ML-reset. Even if high-levels remain on many ML_next, they do not

need to be discharged, because the final match generation always looks at both ML_previous and ML_next, instead of simply forwarding the result of ML_previous to ML_next.

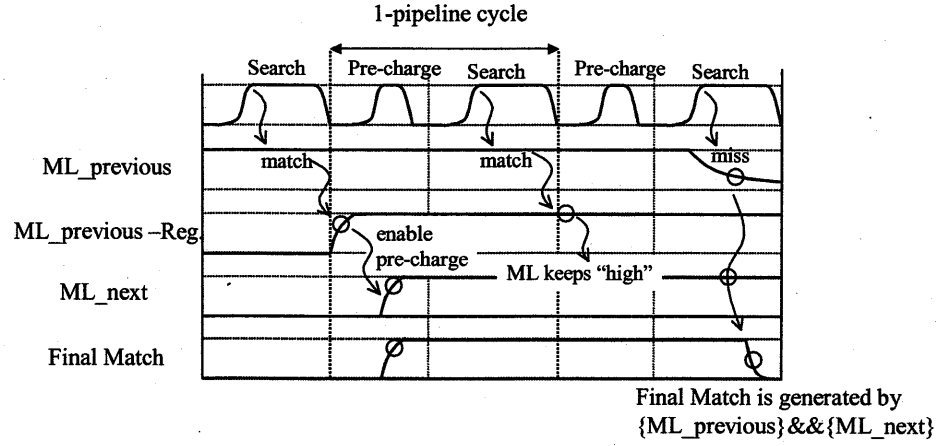


Figure 6.2 Time chart of the proposed improved pipelined search

In the example where 0%-match happens consecutively, that is known as the worst case power dissipation for non-pipelined search, the 2-stage pipelined search reduces 50% of the power dissipation with both the conventional and newly proposed improved concept because the actual search only occurs at the 1st pipeline stage and therefore SLs and MLs of the 2nd stage are completely deactivated by the pipeline control. In contrast, all MLs continuously repeat charging and discharging in the non-pipelined search. The other example where 100%-match happens consecutively, that is known as the best case power dissipation for the non-pipelined search, because all SLs are activated but MLs remain at high-level. In this best case scenario, MLs at the 2nd stage are continuously charged and discharged by the described overall ML-reset in the conventional pipelined search. However, according to the improved pipelined search, MLs at the 2nd stage can keep their high-level due to the removal of the overall ML-reset. As a result, the proposed concept maintains both the benefit of the pipelined search seen in the 0%-match example and the benefit of the non-pipelined search seen in the 100%-match example.

6.3 2-bit encoded storage and for power-reduction of search-line data generation

The second application-driven power saving proposal aims at reducing the SL power dissipation. For the applications of network-address matching, such as for MAC and IP addresses, SL pair needs to be mask-able because each pair corresponds to the network address. Since the smallest unit, which has to be mask-able is not a bit but a byte if the application is limited to signature based, because a signature consists of a byte, e.g. 0100 0001" and "0110 0001" to represent "a" and "A" respectively. Therefore, such application allows that a single mask bit can be used in common for each byte (eight-bit unit) within the search request data. Based on this special property of the signature-oriented application, I can encode the internal search request data and the other data stored in the TCAM by option, to minimize the SL power dissipation. While the encoding of TCAM data was also recently proposed to achieve a minimization of the required storage capacity [44], the purpose of proposed encoding in this work is different, that is focusing on the power reduction by managing the SL generation. According to the conventional SL generation, the eight bits of each byte in the search data are grouped into 4 pairs of two bits and the storage units in the TCAM are correspondingly grouped into pairs of two storage units. As shown in Figure 6.3 which is the conventional model, a pair of two TCAM unit cells comprises four search lines /SL<0>, SL<0>, /SL<1>, and SL<1>.

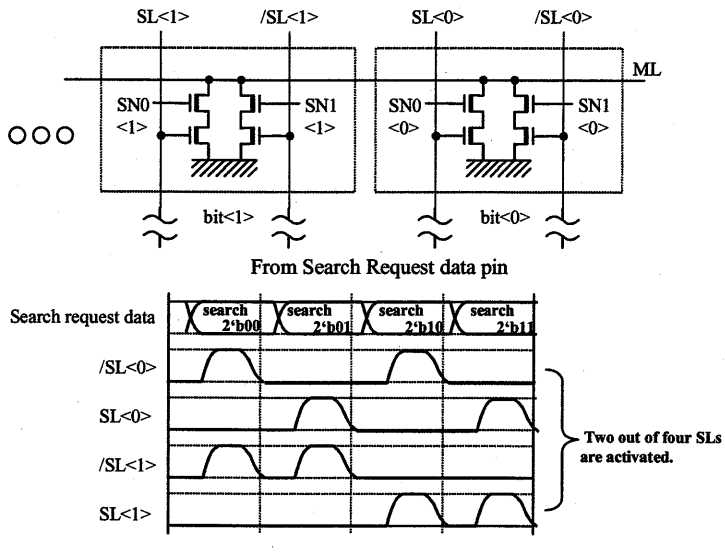


Figure 6.3 Conventional scheme of SL generation

In this architecture, /SL<0> and /SL<1> represent the inverted values of SL<0> and SL<1>, respectively, therefore a high-level for 2 out of these 4 SLs has to be generated under any search configuration. On the contrary, the proposal here is to use these 4 SLs to search for the 4 possible combinations, which are namely bit pairs 00, 01, 10 and 11, as indicated in Figure 6.4. In this way the number of SLs asserted from low-level to high-level can be reduced to only 1 out of 4. SL power dissipation is easily expected to reduce by a factor of 2 to 50%.

In addition, the stored data in cells a, b, c and d in Figure 6.4 has to correctly represent the 9 possible data combinations of the 2 ternary storage units, that is to correct the functional correspondence to the described SL encoding proposal. A suitable encoding possibility and the corresponding functional truth table are listed in Table 6.2.

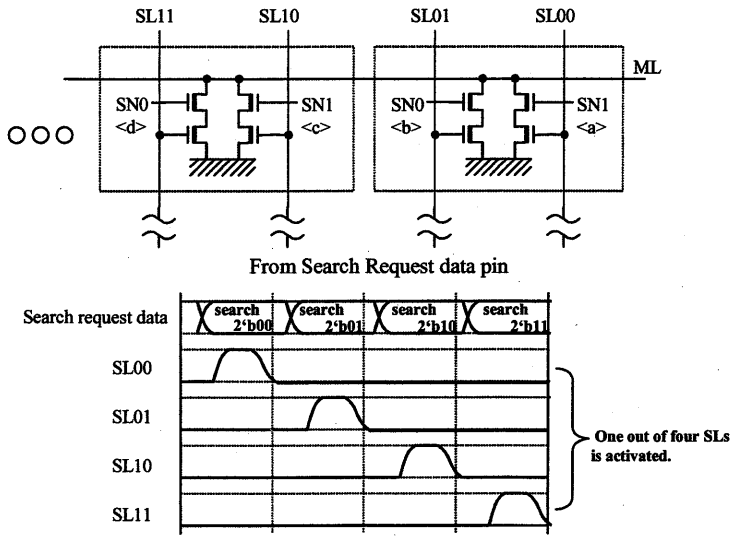


Figure 6.4 Proposed 2-bit encoding for reduced SL power dissipation

Table 6.2 Encoding for data storage and corresponding truth table

	Encoded data storage	Search00 SL00 asserts	Search 01 SL01 asserts	Search10 SL10 asserts	Search11 SL11asserts
#1	Store XX a=0, b=0, c=0, d=0	[match]	[match]	[match]	[match]
#2	Store X0 a=0, b=1, c=0, d=1	[match]	discharge	[match]	discharge
#3	Store X1 a=1, b=0, c=1, d=0	discharge	[match]	discharge	[match]
#4	Store 0X a=0, b=0, c=1, d=1	[match]	[match]	discharge	discharge
#5	Store 00 a=0, b=1, c=1, d=1	[match]	discharge	discharge	discharge
#6	Store 01 a=1, b=0, c=1, d=1	discharge	[match]	discharge	discharge
#7	Store 1X a=1, b=1, c=0, d=0	discharge	discharge	[match]	[match]
#8	Store 10 a=1, b=1, c=0, d=1	discharge	discharge	[match]	discharge
#9	Store 11 a=1, b=1, c=1, d=0	discharge	discharge	discharge	[match]

6.4 Dynamically configurable flexible partitioning

The next power reduction technology is especially addressed to the high-density CAM. A high-density integration is a standard technological trend in memory and is driven by various scaling factors performed in the silicon surface, such as minimization of transistors and implementation of more signal lines.

On the other hand, the high density CAM enables multiple table lookup for the different purposes in a single chip from the application's point of view. Figure 6.5 shows the example of multiple different purposes of lookup in the network application. Nowadays, such multiple table lookups per single packet are commonly required feature. For example, forwarding refers to the destination address contained in the packet header. Classifying refers to the source address, the destination address and often to the upper layers for other purposes of accounting. In addition, as a more complexity manner, table lookups like Quality of Service (QoS) to prioritize the forwarding order and Packet Filtering to exclude particular packets from forwarding are also getting more common.

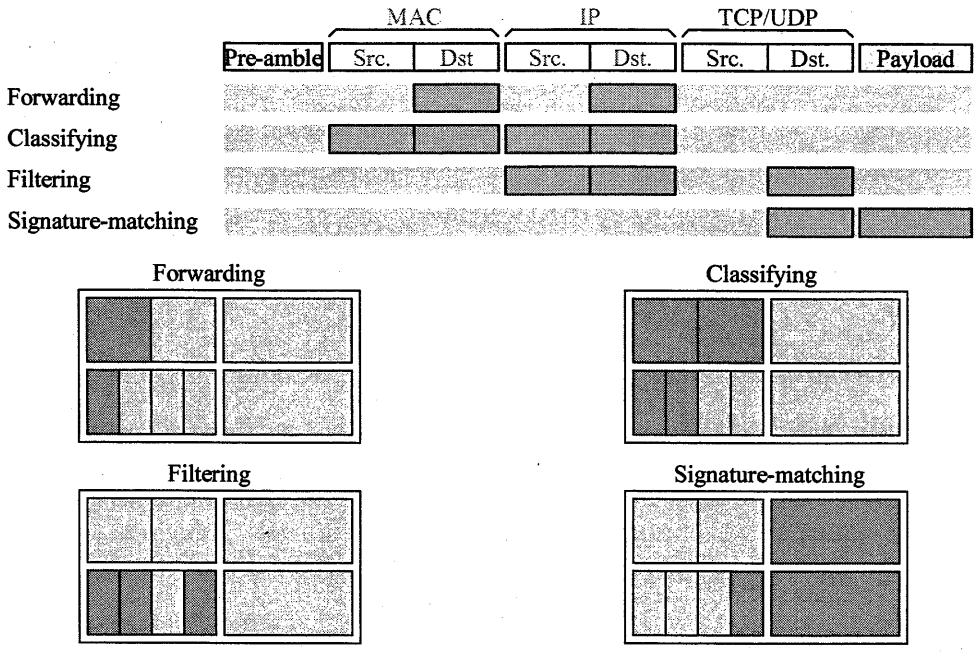


Figure 6.5 Multiple purposes of table lookup

For memory hardware, the difficult table lookup manner in the described network application is that each table size is not constant but each application requires quite unique table size. Moreover, some types of table are overlapped by multiple lookup purposes as shown in the figure. Unfortunately, common partitioning provided by memory has been realized with bank architecture, where the most significant address bit is used, [45] hence, is capable of only partitioning space into banks of the same size. Consequently, it is quite difficult to perform the appropriate partitioning in CAM. Therefore extra memory space often participates to the search operation, even though it might be redundant to lookup in actual application. [46] This has resulted in two problems, one is the concern of unexpected hit occurred from extra lookup table, and the other is the increase of power dissipation. The proposed technology in this work is that users and application should name the additional bits per each bank as the identification of lookup purpose, for example, user can name ID=4'b0000 as MAC src. applied into Bank 0 through Bank3, and names ID=4'b0011 as IP dst. applied into Bank 4 through 7. Having additional 4-bit pins can provide which table is indexed. Although this proposal requires 4-bit extra input pins, flexible partitioning is easily achieved. In this work, additional unique concept applied to the internal partitioning with specialized identification is the use of 'don't care'. More detailed feature, which is namely programmable configurable flexible partitioning, is described in chapter 8.

References

[44] Satoru Hanzawa, Takeshi Sakata, Kazuhiko Kajigaya, Riichirou Takemura, and Takayuki Kawahara, "Pipeline Dynamic CAM based on One-Hot-Spot Block Code for Use in Network Router," Technical Report of IEICE SDM2004-120, ICD2004-62, pp.1-5 (2004).

[45] K. Mal, E. Alon, D. Llu, Y. Kim, D. Patll, and M. Horowitz, "Architecture and Circuit techniques for a Re-configurable Memory Block," ISSCC 2004, Dig., Session 27, 27.5 (Feb. 2004).

[46] G. Kasai, Y. Takarabe, K. Furumi, and M. Yoneda, "200MHz/200Msps 3.2W at 1.5V Vdd, 9.4Mbits Ternary CAM with New Charge Injection Match Detect Circuits and Bank Selection Scheme," IEEE Custom Integrated Circuits Conference, pp.387-390 (2003).

Chapter 7

Newly developed CAM-defect repairing technologies enabling cost-per-bit scalability

7.1 Introduction to the previous state of art

Based on previously described experimental result, this chapter addresses the physical CAM cell structure defect analysis performed on the silicon as well as the influence of particle density. As shown in chapter 5, the defect rate observed in CAM area was approximately 2x higher than other circuitries, even though they were placed on the same silicon. Another related problem to this situation is today's market price. The price-per-bit of CAM is approximately 20x higher than that of SRAM. I can formulate the hypothesis, whereas the market price is used to depend on various factors such as the balance of supply versus demand and/or competitions, the complicated physical cell structure to perform content addressability had the adverse effect of increasing the defect rate. Apparently it affects the production cost, and therefore CAM's market price is extremely high. This raises a serious question in the view of content addressability despite the value of high bandwidth during search. Unfortunately, little attention has been given to the analysis of defect rate especially in CAM. The conventional defect analysis methodology has been mainly focused on the particle density, and hence there is not an adequate explanation of why only CAM has been damaged by the particle more seriously than other circuitries. No speculation has taken place concerning the physical structure. I propose a model of defect analysis, which needs to incorporate the physical structure on the silicon, and verify that the higher defect rate is certainly caused by the complicated physical structure enabling content addressability.
Secondly, I have employed further technologies because I cannot develop the CAM based application specific VLSI unless such unreliable problem occurred in CAM cell is eliminated. It is hard to decrease the particle density, therefore I pay more attention on whether defects are repairable or non-repairable in this analysis, and propose a unique redundancy technology to repair the repairable defects. It must be examined that the cost per bit-cell at post-test is significantly improved by the repair technology, even the yield at pre-test is low. Nevertheless, various scaling factor may keep increasing the defect rate in terms of minimizing line and space. The establishment of the cost per bit-cell is a key for reliable scalability.

7.2 Negative binominal model and defect analysis

As the first step of the defect analysis in this work, the negative binominal equation, which is one of well-known models for the yield prediction, is used. [47-50]

$$Y = 1/(1 + D_0 A / a)^a \qquad 7\text{-}1$$

Y	Yield estimate [%]
D_0	Particle density [pcs. per unit-area]
A	Area [unit-area]
a	Cluster parameter

The equation 7-1 is simply referred to the particle density per unit area, however no speculation has taken place concerning the dependence of physical structure laid-out on silicon. The optimal formula, which encompasses all factors described in the experimental study in chapter 5, has not been proposed yet. There seems to be no established theory to explain why higher defect rate occurs in the structure to perform the content addressability. Figure 7.1 illustrates how I determine the defect rate *F(X)*, by walking a virtual particle around the drawing of entire CAM cell using computer graphics analysis. It depends on the location where the virtual particle hits, whether it results in a defect or not. Also, the defect can be categorized as repairable or non-repairable, which depends on the particle-size and location. For example, it is categorized as repairable if a particle hits the bit-line only, while it is categorized as non-repairable if a particle causes an electrical connection between power and ground.

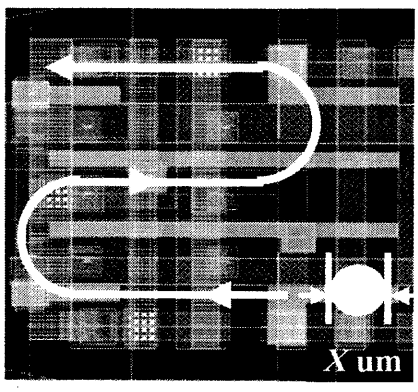


Figure 7.1 defect analysis method

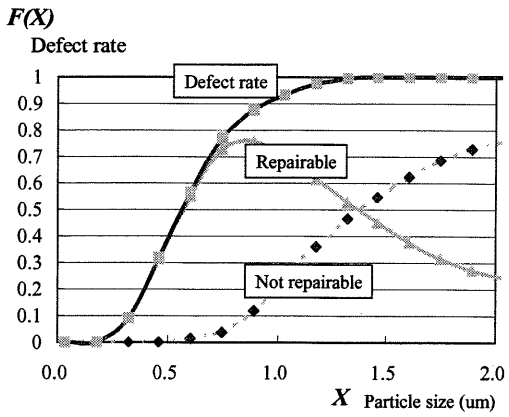


Figure 7.2 Defect rate vs. particle size

The defect rate analysis with the classification of repairable and non-repairable is shown in Figure 7.2. In general, particles with small size occur more often than particles with large size.

$$P(X) \propto X^{-a} \qquad 7\text{-}2$$

$P(X)$	particle probability [pcs. per unit area]
X	particle size [um]
a	depends on the quality level of production site

By combining the defect analysis shown in Figure 7.2 with the particle probability of equation 7-2 this effect is included. The result Figure 7.3 shows a schematic plot of equation 7-2 with the model of typical, best, and worst status of the manufacturing facility, respectively. Note that $P(X)$ just states the particle probability, so that it is not directly related to the yield-relevant defect-particle probability.

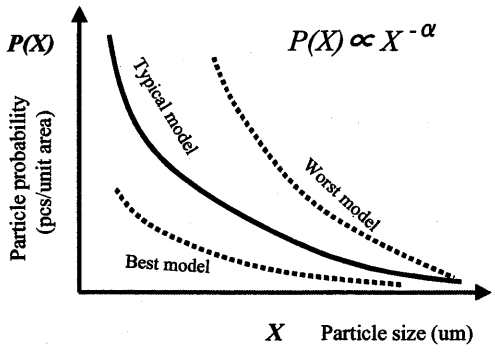


Figure 7.3 Probability vs. particle size

Next, by combining the defect rate analysis for the layout shown in Figure 7.2 with the particle probability shown in Figure 7.3, the resulted actual defect particle probability $F(X)P(X)$ as a function of particle size X represents more accurate account of the probability for failures and yield degradation.

Finally, integrated $F(X)P(X)$ obtaining the relevant defect-particle density D_0_CAM, and also D_0_SRAM in case of SRAM are represented.

$$\int F(X)P(X)dX \Rightarrow D_0\_\text{CAM} \qquad 7\text{-}3$$

$$\int F(X)P(X)dX \Rightarrow D_0\_\text{SRAM} \qquad 7\text{-}4$$

$\int F(X)P(X)dX \Rightarrow D_0$_CAM corresponds to the striped area in Figure 7.4. This area shows

the defect-particle density for the CAM is approximately 1.5 times larger than for the SRAM (D_0_SRAM) in the case of a typical process-line model. The factor increases to almost 2x for a worst process-line model study (not described).

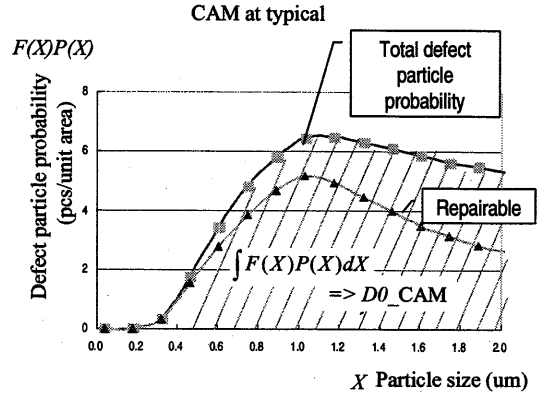


Figure 7.4.a True Defect particle probability vs. particle size

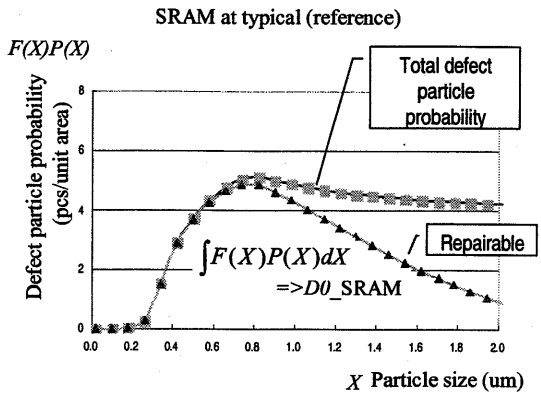


Figure 7.4.b True Defect particle probability vs. particle size

Provided D_0 by the analysis methodology indicates true particle density, which certainly causes the defect in memory cell. This computation is applied to the negative binominal model of equation 7-1 again to predict more accurate yield. I can quantitatively determine the yield reduction due to CAM's unique signal lines SL and ML in comparison to SRAM. With the worst process-line model, the computed pre-test yield is 33.6% in the SRAM case, and only 15.6% in the CAM case per unit density, which is equivalent to the result of experimental study. Note that $\int F(X)P(X)dX$ shown by the striped area consists of repairable and non-repairable defects. It is known that the non-repairable area in CAM is not significantly higher, although total striped area is 2x larger than SRAM. Therefore, even if the yield at pre-test is low, it should be able to improve by repairing with redundant circuitry. That is not special but a commonly used technology in today's standard low cost memories.

7.3 Issue of priority resolving and its effect on the repairing technology

Repair technology can be a solution to make the cost of CAM bit-cell competitive to other memories. Although repairing is a common technology in memory manufacturing [51,52], it is actually not easy to apply to CAM. The dominant factor why CAM is much

more expensive than other memories is hidden in the difficulty of repairing.
CAM's internal address is partitioned into a decoded address just like in other memories, however the output is encoded again. Furthermore, its encoder cell has to communicate with each other for the purpose of priority resolving, and to indicate the lower number of address as the final winner. For example, when three entry-address #0, #1 and #2 are matched at the same time in the search operation, these hit addresses start the communication with each other, assuring that address #1 is stronger than address #2, and address #0 is stronger than address#1, and therefore address#0 is indicated as the final match address. This priority deciding communication must be maintained, even if redundancy is used. It is a very difficult task, because the repairing technology by redundancy conventionally means a complete replacement with memory rows or columns at a different location. As a result, post-test yield cannot be recovered by utilizing this poor repairing method in conventional CAM. Based on the fact that yield loss occurs both at pre-test and post-test due to complicated cell structure and poor redundancy and repair technology, CAM's production yield is significantly low and these are the main reasons why CAM cannot become a price competitive memory.

7.4 Developed redundancy-based repairing technology

The requirement confirmed is to keep the row addressing in the decreasing order of priority. Even if a failure occurs and repair circuitry is used, it is not allowed to re-arrange the order of row addressing. Figure 7.5 shows a block diagram of proposed CAM redundancy.
Two different repairing technologies are combined together to ensure that the row addressing order is unchanged. The first repairing method is managed by software, which is located between the input pin and the CAM core. The second repairing method is achieved by hardware, which is located in conjunction with CAM core array. The failed address detected by the test is pre-programmed in a PROM (Programmable Read Only Memory), by laser trimming. An output of the PROM is transferred to the input for the purpose of the magnitude comparator, which compares the input address to the pre-programmed failed address. This comparison task is executed every input cycle to determine if redundancy is used and that is why I named software-management, while the hardware-management is loaded after power-on only.

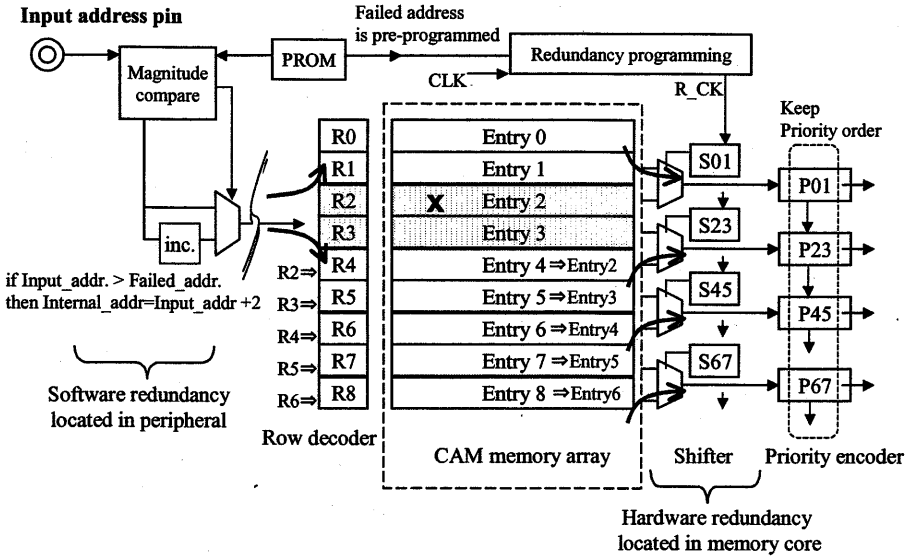


Figure 7.5 Proposed SW/HW combined redundancy

The assumption here is that pre-programmed failed address is R2 (row-address Entry 2). If the input address is smaller than R2, the generated internal address is same as the input address. If the input address is greater than or equal to R2, +2 is added to the internal address to skip the failed address. Magnitude comparator and adder are placed outside of memory array and do not require significant silicon area. This software based repairing methodology is cost-effective, however it will adversely affect the decoding access speed caused by the magnitude comparing function. Another output of the PROM is transferred to the CAM core array via a redundancy-programming circuitry. The components of the hardware redundancy comprises of a MUX and a shift register. The shift register responds to the number of signal R_CK, which the redundancy programming circuitry generates after power on. Similar but reverse to the software-based solution, Entry4, Entry5, and Entry6 are converted to Entry2, Entry3, and Entry4 as illustrated in Figure 7.5, respectively to restore the priority order after repairing. This hardware-based redundancy obviously takes silicon overhead, but doesn't affect the search speed to maintain fast search speed. The reason for a 2-row redundancy is to allow repairing of 2-bit shared failures and to reduce the number of the shift-registers.

This proposed repairing technology with software and hardware takes the priority of fast search rather than slow write, which is preferred by application. The hardware-based redundancy is used during searching, while the software-based redundancy is used during

writing. According to CAM's major usage in network application, the write operation is mainly used in an initialization phase. After all the data is written into the CAM, search will be the main operation, while writing command rarely happens. A representative functional ratio for CAM is >90% for searching and <10% for writing. Thus the busy operations should be hardware-based and the rarely happening operations should be software-based.

The defect analysis described in section 7.2 reveals that CAM's defect-particle density D_0 is higher than that of SRAM, even if the raw particle density is the same. I examined failed bit count in order to optimize the effectiveness of the redundancy. As shown in Table 7.1, when one redundancy set is implemented per 1K-rows, the predicted yield is improved to 74% and 59%, in the best and the worst process-line model, respectively. However, the redundancy implementation also means an area overhead, e.g. 0.9% overhead for one redundancy set per 1K-rows. In the same fashion, I examined various cases of redundancy implementation up to one redundancy set per 128-rows. The most efficient solution in terms of cost-per-bit was determined to be one redundancy set per 256-rows at the bit-cell cost ratio of 0.22. Despite various scaling factors increases raw defect rate, it is verified that proposed redundancy can maintain reliable scalability by the repairing technology.

Table 7.1 Yield prediction with proposed redundancy

Redundancy implementation	Post test Yield estimate		Die overhead	Cost-per-bit Ratio at case-2
	Case-1 Pre-test Y=30%	Case-2 Pre-test Y=15%		
0set	30%	15%	0%	1.00
1set for 1K-rows	**74%**	59%	0.9%	0.26
1set for 512-rows	**80%**	68%	1.9%	0.23
1set for 256-rows	**81%**	**71%**	3.8%	0.22
1set for 128-rows	**82%**	**71%**	7.6%	0.23

References

[47] J. A. Cunningham, "The Use and Evaluation of Yield Models in Integrated Circuit Manufacturing," IEEE Trans. on Semiconductor Manufacturing, Vol.3, No.2, pp.60-71 (May 1990).

[48] C. H. Stapper and R. J. Rosner, "Integrated Circuit Yield Management and Yield Analysis: Development and Implementation," IEEE Trans. on Semiconductor Manufacturing, Vol.8, No.2, pp.95-101 (May 1995).

[49] C. H. Stapper, "Improved Yield Models for Fault-Tolerant Memory Chips," IEEE Trans. on computers, Vol. 42, No.7, pp.872-880 (July 1993).

[50] C. Neil Berglund, "A unified Yield Model Incorporating Both Defect and Parametric Effects," IEEE Trans. on Semiconductor Manufacturing, Vol.9, No.3, pp.447-454 (Aug. 1996).

[51] H. Sato, T. Yamagata, K. Fujita, Y. Nishimura, and K. Anami, "A distributed globally replaceable redundancy scheme for sub-half micron ULSI memories and beyond," Symp. VLSI Circuits, Dig. Tech. Paper, pp.195-201 (May 1993).

[52] S. Kikuda, H. Miyamoto, S. Mori, M. Niiro, and M. Yamada, "Optimized redundancy selection based on failure-related yield model for 64-Mb DRAM and beyond," IEEE Journal of Solid-State Circuits, Vol. 26, pp.1550-1555 (Nov. 1991).

Chapter 8
Fabricated examples of CAM-based application specific VLSI circuits

8.1 Introduction

I will return to the major concern of bandwidth with content addressability. Works described in chapter 6 and 7, targeted directly at cost reduction in terms of power and in terms of bit-cell have successfully established reliable scalability in memory with content addressability. Whereas the bandwidth capability provided by CAM has acknowledged, there seems to be little argument and works for that scalability. Now I develop the discussion of CAM based application specific VLSI for the network application, as the basis of provided two reliable scaling efforts.

There are two CAM based VLSI models described in this chapter. One is high-density TCAM, which proposes a dynamic flexible partitioning for the purpose of various complicated table lookup, which is today's major application in switches and routers. The other model is a signature-matching co-processor, which is applicable to real-time filtering of misused packets without disturbing the forwarding operation of the network. The benefit of CAM based performance in comparison with the conventional tree algorithm is examined. Beyond that, there has been a renewal of interest in intelligent functionality in content addressability. A sufficient number of bandwidth derived from CAM can replace the complicated task of regular guard related in network security into the memory with intelligence.

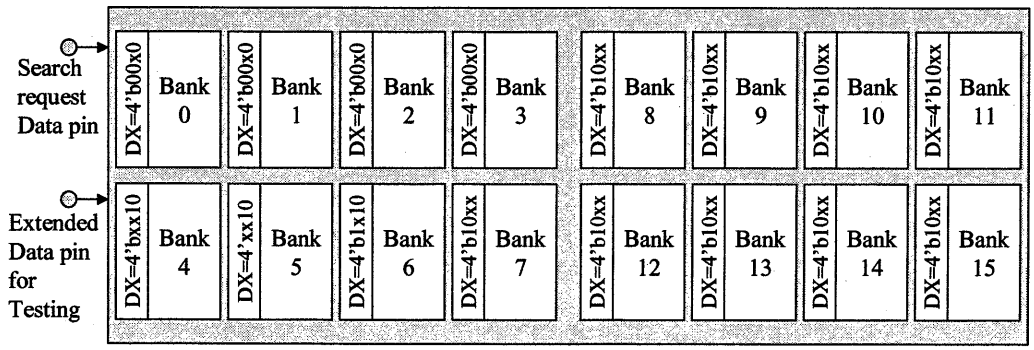
8.2 High-density 18M-bit full ternary CAM VLSI

The direction of high-density integration is a standard technical trend for semiconductor memory. Note that it is not carried out unless reliable scalability is primarily established. Now, this technical trend can also be applied to CAM by described verification works, one is cost of power and the other is cost of bit-cell. 18M-bit full ternary CAM VLSI has been developed to adopt today's network application. There are two key technologies, one is the flexible partitioning and the other is the intelligent aging function.

In routers, great numbers of packet go around the network traffic, hence multiple purposes of table lookup is required per single packet, that is also described in Chapter 6. Again, the difficult table lookup manner in network application is that each table size is quite unique, and some tables are often overlapped by multiple lookup purposes. On the contrary, the configuration provided by conventional memories simply partitions the entire memory into the same space size. For example, an 18M-bit device consists of Bank A through D, each comprises of same 4M-bit density.

8.2.1 Dynamically-configurable flexible partitioning

As shown in Figure 8.1, proposed flexible partitioning applied to the experimental 18M-bit TCAM VLSI consists of sixteen banks, e.g. Bank 0 through 15 where each bank contains extended data bits (DX) for the purpose of primary search. Each DX comprises of four ternary CAM bits searched at the first pipelining stage, and the next search applied to corresponding Bank executed only when the result of primary DX search is passed. Consequently, this architecture provides fully programmable functional partitioning, which each lookup size is unique, regardless of the physical partitioning by Bank. Figure 8.1 also illustrates the example of functional partitioning including Forwarding, Filtering and other purposes.



Forwarding	Bank 0,1,2 and 3 DX = 4'b0000	Filtering	Bank 4,5 and 6 DX = 4'b1110
Classifying	Bank 0,1,2,3,4 and 5 DX = 4'b0010	Signature-matching	Bank 7,8,9,10,11,12,13,14 and 15 DX = 4'b1000

Figure 8.1 Proposed flexible partitioning with extended DX bit

While each bank consists of 16K-entry times 72-bit lookup size, DX per bank comprises just a single entry times 4-bit. That is because the major purpose of DX is to eliminate extra bank in a search, in other words it can restrict the extra active power, which participates in the search. Single entry per bank is appropriate number for DX to function with the advantage of cost saving.

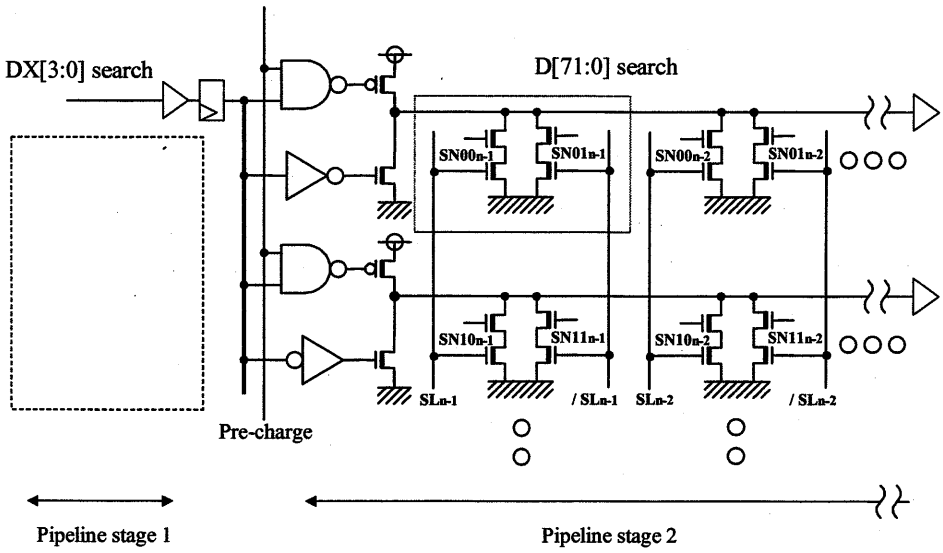


Figure 8.2 Hierarchical Search scheme with DX

8.2.2 Intelligent aging function for CAM entries

We often examine the functionality of memory hardware in comparison with our human brain. This objective is to evaluate the memory from the aspect of intelligence. Traditionally, aging is one of the difficult tasks for memory hardware, while our brain relatively manages it easily. The human brain memorizes various data carried out from information outside, and also we commonly forget and lose part of this data from memory. In fact, the task "forget" is very important to memory maintenance, because "forget" creates an open space in memory and that space can be refreshed for incoming data from outside again. Since memory has a limited capacity, we cannot stack data forever unless some part is lost. This high-level control management in our brain determines which part should be memorizing and which

part should be lost. Priority management is typically qualified by access count. Higher priority means that the data access is executed more, while lower priority means less number of data access. I conducted the integration of this aging operation into experimental 18M-bit CAM described as follows.

Proposed technology enabling aging operation is not provided by expensive large-scale hardware, but by cost effective simple modification. For example known architecture to achieve aging function is to integrate SRAM comprising same number of entry-address times 1-bit depth, which is placed in conjunction with corresponding CAM entry. 1-bit depth SRAM can identify if hit happens or not per entry of CAM. This is like a table with the function of hit flag, however, this approach involves several problems. One is the difficult write operation for SRAM. When multiple CAM entries match as a result of search, corresponding addresses of depth bits in SRAM have to be written simultaneously. In fact, it is difficult to write into multiple addresses simultaneously, therefore it can be executed by serially separated multiple write cycles. Consequently, it takes longer clock cycle than the case of single entry match. In addition, it needs great many read operations for depth bit in SRAM to know the hit flag entirely. Additional problem in this SRAM table is the cost of hardware. SRAM with 256K-address space is a serious overhead in addition to 18M-bit CAM VLSI with 256K-entry times 72-bit.

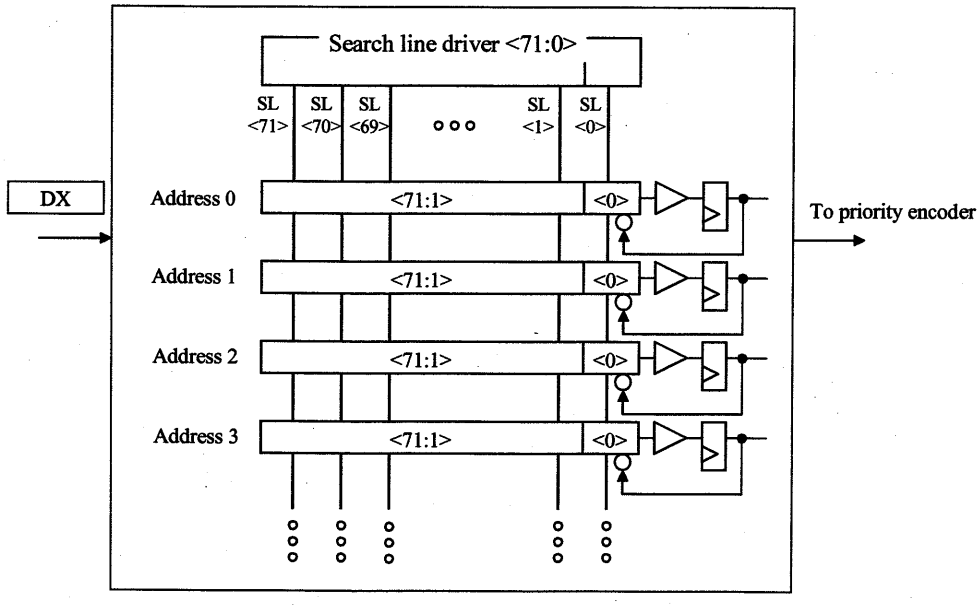


Figure 8.3 Extended function of Empty/Occupied indicator and Aging

Figure 8.3 shows the concept of aging applied to the experimental 18M-bit TCAM. The special bit to identify if "hit" happens or not is provided by raw TCAM cell bit <0> instead of additional SRAM bit. In other words, it does not create any specific memory hardware, but provided by one of modified CAM cell. The modification is that bit<0> in this figure can manage the aging function as well as the standard CAM function. First, bit <0> for all entries are written with data "1" for the purpose of initial reset. The primary definition, bit <0> with data "1" means that entry is "vacant" state. Second, data "0" is written into bit <0> when actual write operation is executed to the corresponding entry. Therefore, bit <0> keeps "1" if that entry is not accessed yet, while bit <0> turns to "0" if that entry is surely accessed. Third, user can provide the search operation with that the search request data bit <0> is data "0". Since the entries involving data "1" at bit <0> never indicate hit, this application can removes the unexpected hit occurred by vacant entry.

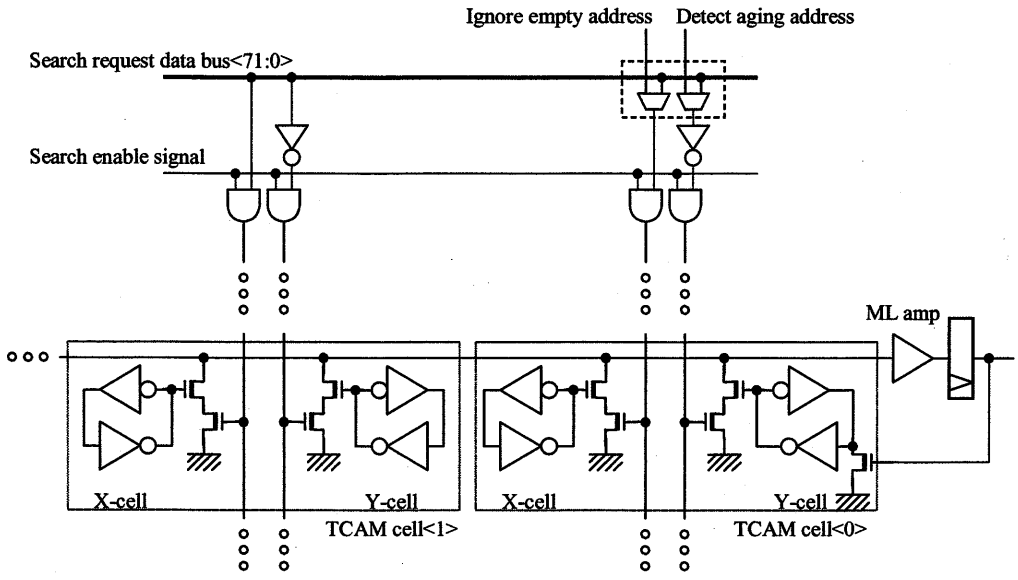


Figure 8.4 Proposed Aging scheme in 18M-bit CAM

Figure 8.4 shows a more detailed explanation in the modification of address bit <0>. TCAM cell consists of X-cell and Y-cell, and the special usage of described vacant/occupied indicator is provided by X-cell only. In this figure, X-cell is written with data "1" for the purpose of initialization. When write operation to that entry happens, X-cell is written with

data "0" to state that entry is occupied. The MUX of search-line controls either the usage of vacant/occupied indicator or just one of CAM X-cell as bit <0>. When it is used as the vacant/occupied indicator, the search-line for X-cell should keep physical data "0" to eliminate extra hit to occur from empty entries. In other words, empty entries never state hit regardless of data <71:1>. Equally important, the usage of Y-cell is the proposal of aging, which the output of match-line amp returns to Y-cell to flip the data for the purpose of memorizing a hit status. Y-cell should be initialized to data "0" to show hit does not happen yet. It is only turned to data "1" by the actual event of hit during search.

The special search operation with that search request data bit <71:1> are all masked and bit <0> is data "0" for both X-cell and Y-cell, tells us the entry which is occupied but does not state hit yet. User and application easily know the entry for aging, and this entry can be erased as lower priority. MUX for Y-cell search line also controls either aging function or typical usage. Hence intelligent aging function is successfully combined to CAM array without any serious overhead of hardware.

8.2.3 Remaining issue of power-supply noise due to large *di/dt*

The second half of this chapter describes the AC power dissipation, which is an additional problem experienced in fabricated 18M-bit TCAM VLSI. In today's VLSI design work, the concern of internal power supply is commonly evaluated by the pre-silicon verification, which is named I-R drop. The internal resistance is extracted from the layout-drawing data, and is combined into the simulation as back-annotated data. The power simulations are repeatedly executed with and without the extracted internal resistance, and this comparison result represents the I-R drop.

Figure 8.5 illustrates the example of I-R drop in fabricated TCAM VLSI, which is drawn by computer graphics. In this analysis, power-supply voltage is provided by wire-bonds located around the CAM array, therefore the center of the chip seems to indicate the worse I-R drop. It is however better at the actual center of the chip marked by "+". That is because the peripheral control circuitries, which do not consume tremendous power are placed there. In this development, it was examined at pre-silicon evaluation that the peak voltage drop level was less than 200mV, that is not significantly worse number in comparison with other VLSI design.

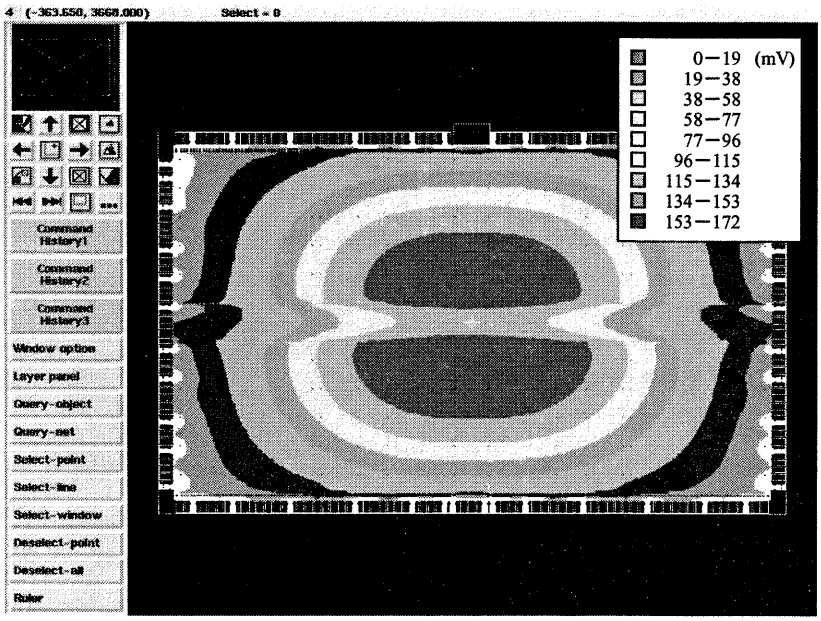


Figure 8.5 Example of I-R drop simulation

On the contrary, actual measured data at post-silicon evaluation are shown in Figure 8.6 through Figure 8.11. These figures demonstrate the search performance in addition to the measured I-R drop. The fact that search performance depends on the active area during search is made clear in these figures, that is the measured search speed is getting slower with the increase of active-area due to the voltage drop of power supply.

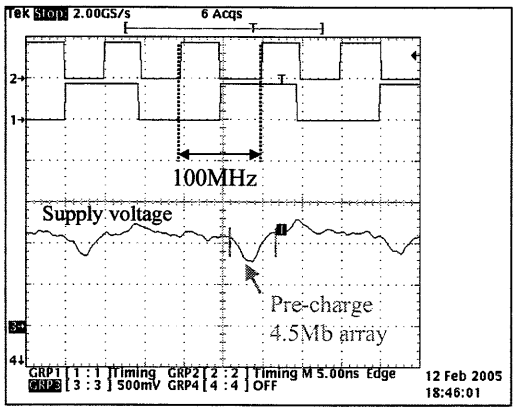


Figure 8.6 VDD waveform with 4.5Mb CAM array

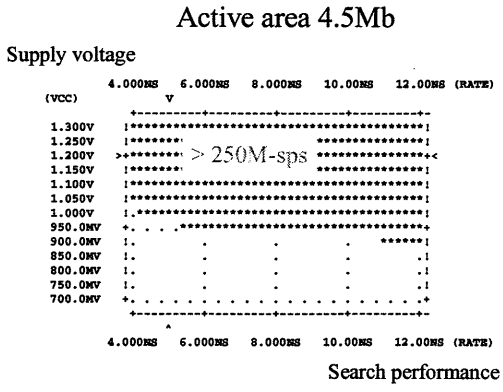


Figure 8.7 Schmoo plot with 4.5Mb CAM array

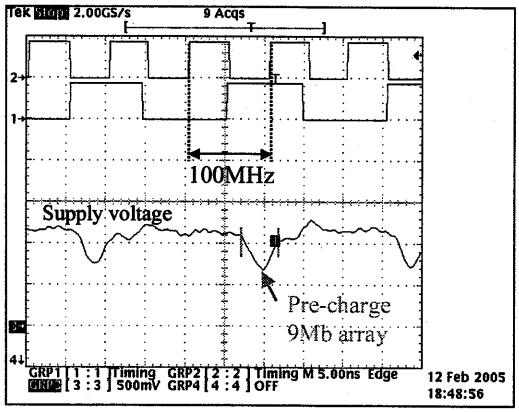


Figure 8.8 VDD waveform with 9Mb CAM array

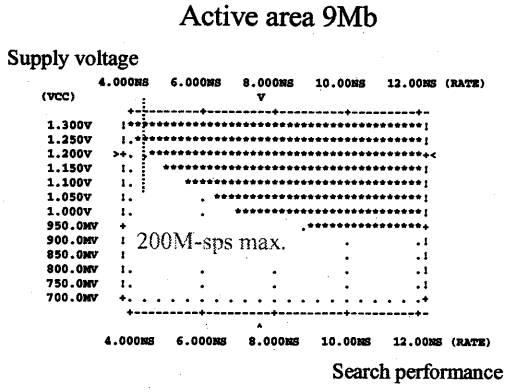


Figure 8.9 Schmoo plot with 9Mb CAM array

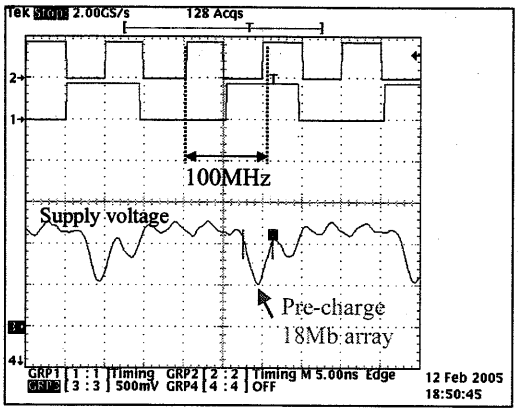


Figure 8.10 VDD waveform with 18Mb CAM array

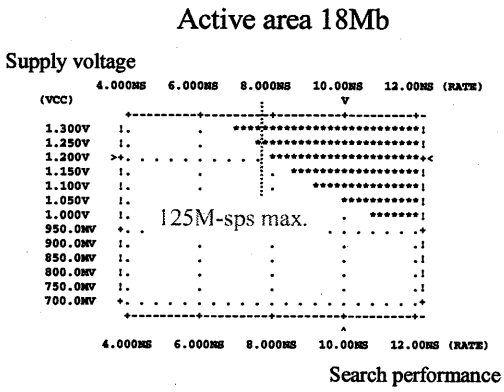


Figure 8.11 Schmoo plot with 18Mb CAM array

In fact, this problem is not caused by the mismatch between pre-silicon simulation and actual measurement at post-silicon. Apparently, the root cause is not internal resistance, but the inductance (L) contained in the package. The inductance accelerates the noise with the factor of *di/dt*. In typical VLSI, which comprises large number of transistors such as today's high performance CPU, most of the transistors are continuously functioning therefore it appears as almost constant power consumption. In contrast, CAM consumes tremendous power but only during the pre-charge of match-line and the assertion of search-line. That is completed in narrow timing space in entire search timing. As a result, the factor of *di/dt* indicates unique characteristic in CAM, which cannot be simply represented by gate-count alone. Even though both high performance processor and CAM consume huge power, large *di/dt* is a serious concern for CAM. Also, the technical trend of various scaling factors cannot compromise the growth of search performance, therefore the management of narrowing

timing space will make this situation more difficult.

The problem of supply voltage reduction, caused by L and C contained in the package, is generally calculated by following equations.

$$\Delta v = L\frac{di}{dt} \qquad 8\text{-}1$$

$$\Delta v = \frac{1}{C}\int idt \qquad 8\text{-}2$$

Actual L and C in the LSI package are not simply represented by the one-dimensional equations 8-1 and 8-2, but are performed by a complicated networked structure as shown in Figure 8.12. Hence an integral computer analysis 8-3 becomes necessary. [53]

$$\Delta V(f) = \sum Z(f)\cdot I(f) \qquad 8\text{-}3$$

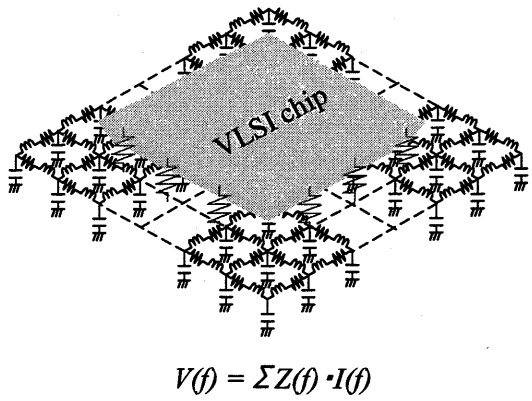


Figure 8.12 VLSI and PKG. modeling

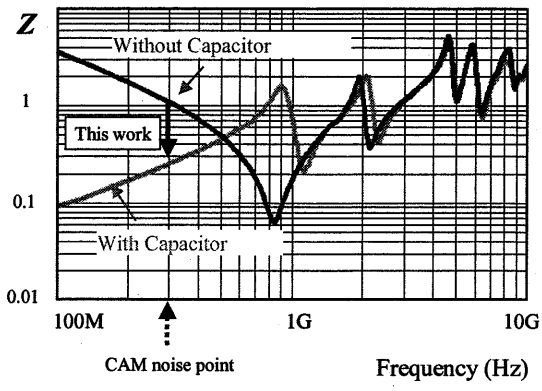


Figure 8.13 Effect of on-package capacitor

It is not easy to reduce the inductance contained in the package, however the integration of capacitors into the package can effectively reduce the impedance $Z(f)$. Figure 8.13 is an example of comparison study in case of without capacitor and with capacitor. [54] As actual operating frequency in the duration of the pre-charge and SL assertion is less than 300MHz, a decouple capacitor performed on package can effectively reduce $Z(f)$ in the experimental 18M-bit TCAM. Nevertheless, the application and the scaling factors do not stay in 300MHz

but surely moves towards ~GHz or beyond, therefore, described solution with capacitor integration does not profit in near future. The problem of *di/dt* remains as an issue to be discussed.

The die photo and major characteristics of fabricated 18M-bit TCAM VLSI are described in Figure 8.14 and Table 8.1, respectively.

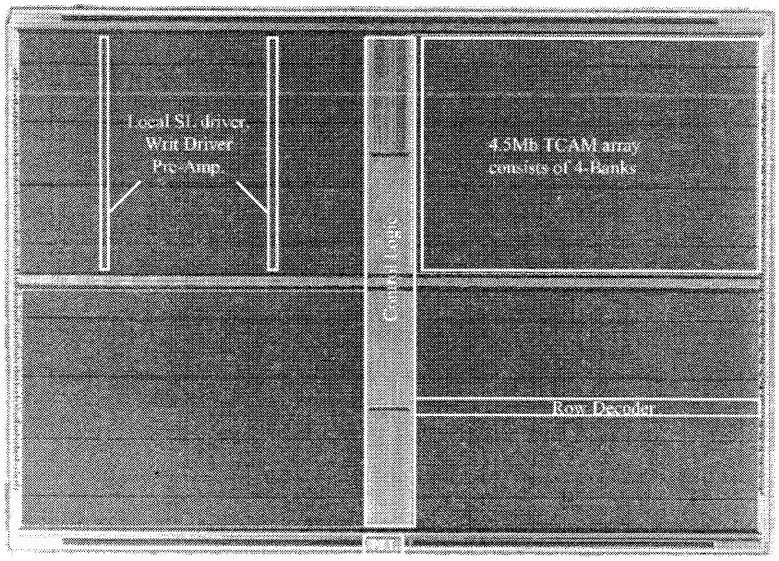


Figure 8.14 Experimental 18M-bit TCAM VLSI

Table 8.1 Summary of features and performance data for 18Mb full ternary CAM

Technology	6AL +1Poly 0.13um Generic technology
Power supply	1.2V for TCAM core 2.5V for IO
Configuration	256K x 72b / 128K x 144b / 64K x288b / 32K x576b Dynamically configurable 18Mb full ternary CAM
Performance	125M-sps max.
Extended Features	Flexible Partitioning within 16-Banks vacant/occupied indicator for Aging function
Power consumption	12W at 125M-sps worst

8.3 Signature-matching co-processor VLSI

Established scalability for the cost of power described in chapter 6 and cost of bit-cell described in chapter 7 are also applied in TCAM-based design of the signature-matching co-processor VLSI. As the proposed power reduction for encoded SL presents, focusing on the specific application properties can often provide much more effective solutions than the restricted discussion of a general-purpose hardware core. On the other hand, such application properties sometimes limit the value of a supposedly optimum concept, as in the pipelined search case where the desired effect of reduced power dissipation is less seen in the case of larger numbers of multiple matches.

Figure 8.15 shows the block diagram of the signature-matching co-processor VLSI, which is designed in 130nm CMOS technology. The maximized performance 125M-sps per byte (per character) has been targeted to enable real-time filtering of packets in 1G-bps Ethernet, without disturbing the normal operation in the network traffic. A further important point in addition to the application-driven power-reduction is described as follows.

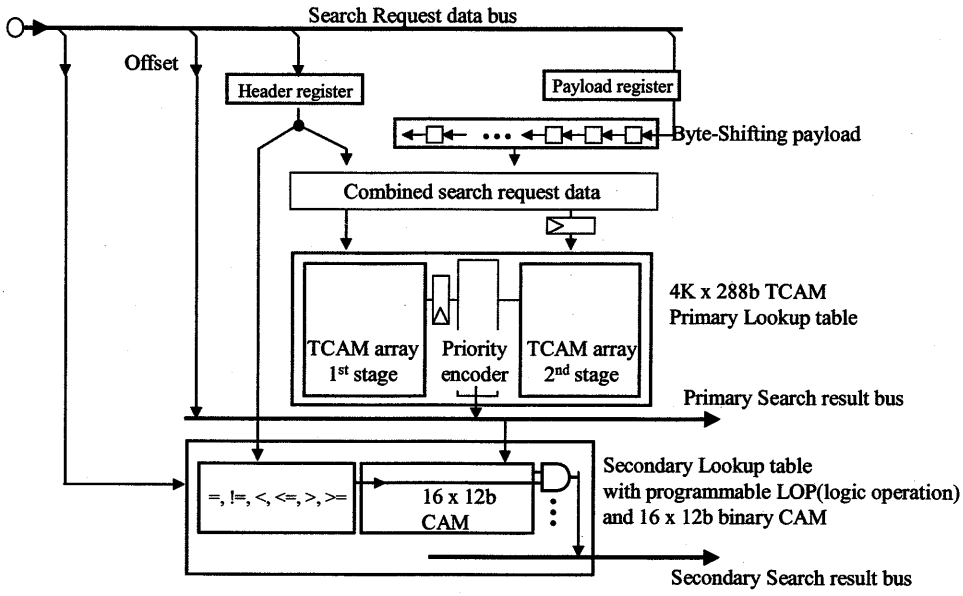


Figure 8.15 Block diagram of signature-matching co-processor

8.3.1 Search request data for thorough byte-shifted payload

A length of 288-bit is chosen for search request data as well as the reference data stored in the TCAM. Header and payload length are programmable and a typical configuration is the division into a 32-bit header and a 256-bit payload. An important concept is the reduction of bandwidth demanded by I/O pin. This bandwidth would amount to 4.5G-Bps at a search speed of 125M-sps, if the search request data is supplied in the conventional way via 288-pins of the VLSI chip. In the design reported here, the header is loaded only once via the input pins and is stored in an internal header register for constant use. Also, the payload part of the search request data is generated by an integrated byte-shifting register, which provides a sequential character-by-character shifting after every search cycle. The combined search request data (header plus byte-shifted payload) is then fully compared with the signature database in a single clock cycle. The designed storage capacity is for 4K signatures each with 288 bits. Additionally the shift offset within the packet is recorded by incrementing a counter after every clock cycle during a packet check. Due to this offset recording, the signature-matching co-processor knows and can output the position of a detected signature within the packet. The header register and the byte-shifting register lead to a reduction of the external bandwidth demanded. Since only one byte per clock cycle has to be loaded from the input pins, the required bandwidth is reduced by nearly a factor 40 to only 125M-Bps at the operating speed of 125M-sps.

8.3.2 Secondary lookup table with programmable logic operations

In recent years, useless packet detection is getting increasingly complicated. The header check, in particular, often requires logic operations for some types of useless packets. A typical example is the application of magnitude comparisons for a range test instead of match test. The detection of such complicated useless packets is achieved with the second key concept of the signature-matching co-processor, which uses a secondary lookup table operating in combination with programmable logic operations (LOPs) on the header. Several programmable LOP tests, implemented in a hierarchical structure, can be carried out for the header as indicated in Figure 8.15. The header-register data is therefore not only transferred to the primary lookup but also concurrently to the secondary lookup. The throughput of the secondary lookup is four times slower than that of primary lookup because of the LOP task and because it is not necessary for the secondary lookup to produce a result every clock

cycle. While the purpose of the primary lookup is to check every byte in a shifting manner to detect signatures hidden in the payload, the secondary lookup is just a single check per packet to detect certain ranges in the header. When the LOP check is finished, the result is transferred with four clocks latency to another small binary CAM which is also integrated into the secondary lookup. Actual data size of this binary CAM is 12-bit width, which corresponds to the address size of the entries in the primary lookup table, i.e. $2^{12} = 4K$. For example, if a packet-detection test consists of a range test 'LESS than 80' for the header and the signature test 'Apple' for the payload, first that 'LESS than 80' would be pre-programmed in the LOP table. Then the binary CAM starts a matching operation after the LOP test with packet header is passed. However, the binary CAM is not capable of direct signature matching but just to check for an entry address of the primary lookup table. That is to say, when 'Apple' is written in the primary lookup table with an entry address 4, the binary CAM will have this address stored as the 12-bit binary 12'b0000 0000 0100 in its secondary lookup table. The 12-bit search request data of the binary CAM is driven by the search result bus of the primary lookup, that is 12'b0000 0000 0100 if 'Apple' is detected in the primary lookup with entry address = 4. In this way, the secondary lookup identifies both a particular logical condition in the header and a specific signature in the payload.

The decision of handling LOP in a secondary lookup and not in the primary lookup was made because the LOP would make the primary look-up too complex and there are less useless packets which require these additional checks of the header. A further positive result is the small size of the required binary CAM integrated in the secondary lookup with only 16 entries each having a length of 12 bits.

The die photo Figure 8.16 shows the signature-matching co-processor, as designed and fabricated in 130nm low-leakage CMOS technology, and Table 8.2 summarizes the main design and performance features.

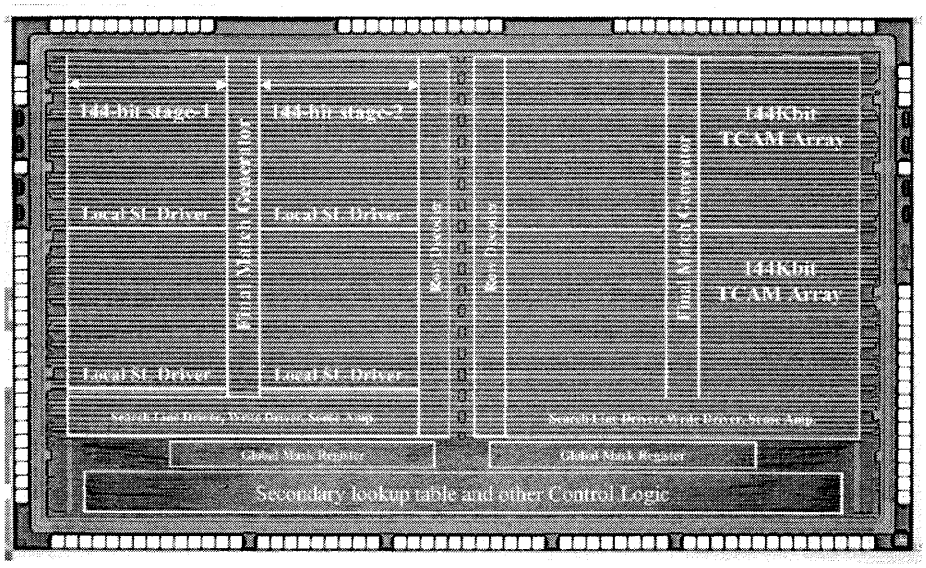


Figure 8.16 Die photo of fabricated signature-matching co-processor in 130nm low-leakage CMOS

Table 8.2 Summary of features and performance data for the signature-matching co-processor

Technology	6AL +1Poly 0.13um Low-Leak technology
Power supply	1.2V~1.5V for TCAM core 2.5V for IO
Primary lookup Features Performance limit	4K-entry x 288b 1.125Mb full Ternary CAM 0b ~32b Header + 256b max. Payload with byte-shifting 125M-sps max.
Secondary lookup Features Performance limit	=, !=, >, >=, <, <=, Programmable Logic Operation 16-entry x 12b Binary CAM 31.25M-sps max.
Power dissipation (worst case at 125M-sps)	1.4W without Encoded search-line 1.1W with Encoded search-line

Power dissipation measurements of this signature-matching co-processor were made for the best case and worst case search conditions configured with and without SL encoding. Figure 8.17 shows the plot of the measured results. The best case scenario is measured under the condition of 100% matches, where the power related to ML is 0% because all MLs stay

at high-level. In the conventional pipelined search, this benefit unfortunately disappears due to the overall ML-reset. The worst case measurement is done under the condition of 0% matches, where the power related to ML is 100% because all MLs are repeatedly charged and discharged in the non-pipelined search. This power dissipation is reduced in the same way by both the conventional and proposed improved pipelined search. The measurements verify that proposed improved pipelined search maintains both advantages, namely the benefit of the non-pipelined search in case of 100% matches and the benefit of the pipelined search in case of 0% matches. The contribution of encoded SLs further reduces the power dissipation of SL by 50%.

Figure 8.17 also shows two power-supply cases of 1.2V and 1.5V, corresponding to the performance targets of 100M-bps and G-bps Ethernet, respectively. Regardless of the number of multiple matches, measured maximum power dissipation with SL encoding is below 1.1W at the search performance of 125M-sps. The usage or non-usage of SL encoding is kept programmable to enlarge the application range of the designed signature-matching co-processor.

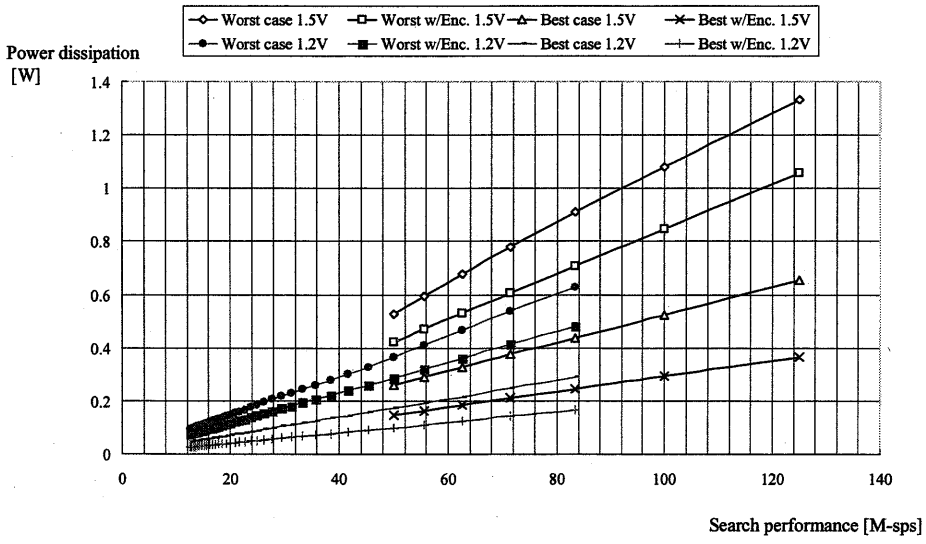


Figure 8.17 Measured power dissipation of fabricated signature-matching co-processor

Figure 8.18 shows the analysis of the described power saving proposals for the worst case search operation. The other power saving factor provided by core voltage reduction is

eliminated in this figure to make the evaluation clear. Since the search performance of this paper's signature-matching co-processor is boosted by 2.5 times as fast, power dissipation should be also 2.5 times higher, which is the starting point in this analysis. Note that the worst case used in this Figure is not exactly equal to, but close to 0% match. Because SLs generation at the 2nd pipeline stage is completely deactivated only in case of 0% matches in the improved pipelined search, I used the condition where single ML's match remained per sub-array to examine the power of SL generation at the 2nd stage as the worst case in this analysis. Power saving with factor of 15%, 29%, and 11% are achieved due to IO bandwidth savings by the byte-shifter, the ML effects of the improved pipelined search, and encoded SL generation, respectively. [55]

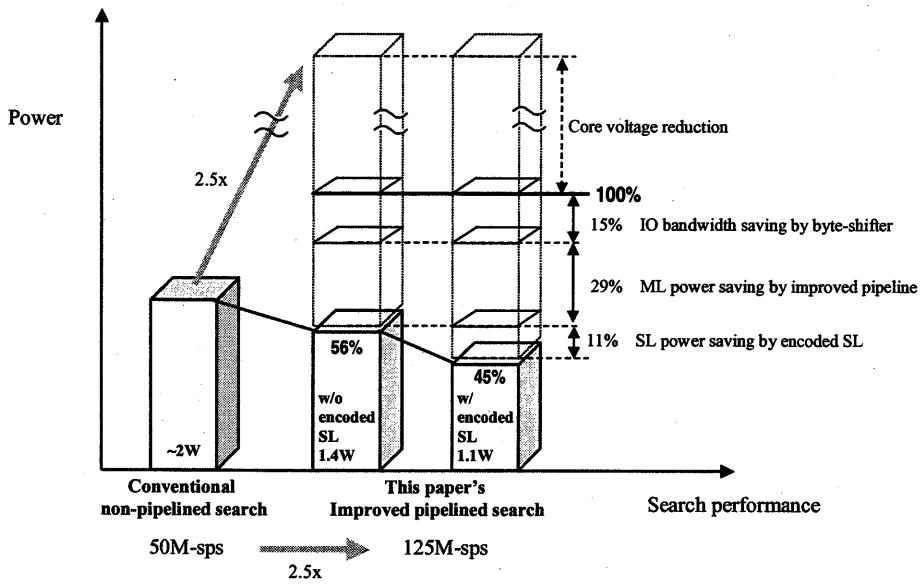


Figure 8.18 Analysis of the relative contributions of the different power saving proposals

References

[53] N. Na, T. Budell, C. Chiu, E. Tremble, and I. Wemple, "The Effects of On-chip and Package Decoupling Capacitors and an Efficient Decoupling Methodology", IEEE electronic Components and Technology Conference, Dig., pp.556-567 (2004).

[54] K. Nakagawa, M. Watanabe, S. Baba, K. Yamagishi, Y. Sasaki, M. Namatame and M. Kimura, "High Speed Transmission and Power Ground Characteristics of Flip-chip BGA Package Over 2,000pin Counts", Technical Report of IEICE, CPM2002-160 ICD2002-205, Tokyo, Japan, pp. 47-51, (2003).

[55] K. Inoue, H. Noda, K. Arimoto, H. J. Mattausch, and T. Koide, "A CAM-based signature-matching co-processor with application-driven power-reduction features," IEICE Trans. Electron, Vol.E88-C, No.6, pp.1332-1342 (2005).

Chapter 9
Conclusion

In this thesis, I have reported new methods and technologies for functional memories with a special focus on content addressability, and have closely discussed the relations between memory access bandwidth and scalability. The main achievements of my research work can be summarized in three major groups as described in the following.

1. The first objective and result of this research work is an examination of the true access bandwidth demanded by applications. The survey and the examination definitely indicate that the bandwidth desired by applications and that provided by memory do not match. The main analyzed examples came from the fields of graphics applications and network applications. In particular, I have proposed that functional integration onto memory with the utilization of an internal bus can be a rather cost effective high bandwidth solution in comparison to the conventional external I/O bus, and have experimentally verified it with fabricated VLSI circuits. The proposed circuit technologies are capable of driving further high-bandwidth specific VLSI developments as well as exploiting the benefit of saved memory-external bandwidth. Following application-related results are derived from the actual design experiments conducted in this research work.

1) An integrated pixel-processing ALU, which is devoted to Z-compare and A-Blend functions in 3D graphics application, has successfully converted the conventional read-modify-write operation into a pure write operation. In other words, functional integration saves external memory-bandwidth demand and nevertheless doubles the performance delivered by the designed functional memory.

2) The utilization of an internal data bus is exploited for the data transferring capacity of the sense-amplifier. The application of a fast erase of the screen can be carried out by the proposed duplicate page operation with a bandwidth provided by the number of sense-amplifiers times their operating frequency.

3) Further utilization of the internal data bus is provided at the memory cell level. When all memory cells function simultaneously, the bandwidth is really maximized. I have verified the functional advantage of the use of all memory cells in the network

application, in the form of a content addressability function.

2. The second focus of attention is the content-addressability solution for high effective bandwidth. In this case every memory cell contributes to the maximized bandwidth capacity rather than an internal data bus or the sense-amplifiers as described above. I have examined the performance gain as well as the related cost of content addressability performed on memory in detail. During these examinations, I noticed that numerous scaling factors contribute to a negative effect on content addressability due to the technological trend, especially in terms of cost of power dissipation and cost per bit. I have therefore put a major effort towards establishing reliable scaling properties for content addressability, as a pre-requisite of successful application specific VLSI on the basis of content addressability.

1) I have proposed an improved hierarchical pipelined partial search, which in comparison to the basic conventional pipelined search, enables sufficient power reduction even in the case of many multiple matches, which are typical for network applications. The achieved solution can provide stable scalability in power dissipation, even in view of the standard technological trend driven by various scaling factors, which towards higher memory-cell density and faster operating frequency.

2) I have proposed a unique methodology for the defect analysis with respect to the physical structures performed on the memory cell, and it revealed that content addressability on the memory-cell level does severely affect the defect rate.

3) I have proposed an effective repair technology applied to the content addressable memory in order to repair defects, which appear more often than in other memories. In this way I could establish the second necessary scalability in terms of cost per bit, making sure that the negative effect of an increased defect rate with high-density integration can be overcome.

3. Established scalabilities with respect to power dissipation and cost-per-bit, enabled the development of content addressable memory (CAM) based application specific VLSI. I have verified experimentally several application cases.

1) The integration of multiple table lookups performed on a single CAM, which is

fabricated as an 18M-bit ternary CAM VLSI in 0.13μm CMOS technology. The proposed flexible partitioning has led to further power savings by eliminating extra lookup space, caused by the more complicated lookup manner in today's typical network applications.

2) As an intelligent function, the fabricated 18M-bit TCAM VLSI integrates an aging operation to identify either continued storing or possible replacement in the database. The verified aging technology is a cost effective proposal, since it is not provided by large scaled hardware, but by a simple modification of the existing CAM cell. I have thus verified the possibility of adding new values to content addressability, enabling an intelligent functional memory.

3) A signature-matching co-processor with 0.13μm low-leak CMOS technology for the application field of network security, which achieves real-time packet filtering without disturbing the packet flow at the performance of G-bps speed. The integrated byte-shift for the payload can carry out the difficult misused packet detection, which is also an evidence of exploiting content addressability for intelligent functional memories.

In addition, it seems appropriate to remark that the factor of *di/dt,* i.e. the time-change in the power dissipation, will be a further concern for continued technological scaling. It is unfortunately beyond the scope of this present work. Although the verified positive effect of on-package capacitors can be part of the solutions, the discussion of an effective solution is still open for operating frequencies in the G-Hz class.

This thesis has disclosed new technologies for the optimization of memory bandwidth with a main focus on the bandwidth effects of content addressability. Despite several problems, which I actually experienced, I could prove the efficiency and validity of content addressability as an intelligent high bandwidth solution. Achieved results lead to the conclusion that this research can drive further development opportunities of CAM based specific VLSI with respect to numerous expanded applications. I can say with fair certainty that continuous researches and experiments will strengthen the validity and the application range of my propositions made in this thesis.

Published Paper and Referenced Paper

Published Paper List

(1) Kazunari Inoue, Hideyuki Noda, Kazutami Arimoto, Hans Juergen Mattausch, and Tetsushi Koide, "A CAM-based signature-matching co-processor with application-driven power-reduction features," IEICE Trans. Electron, Vol. E88-C, No. 6, pp. 1332-1342 (2005).

(2) Kazunari Inoue, Hideaki Abe, Kaori Hayashi, and Shuji Fukagawa, "A low-voltage 42.4G-BPS read-modify-write bus and programmable page-size on a 3D frame-buffer," IEICE Trans. Electron, Vol.E83-C, No. 2, pp. 195-204 (2000).

(3) Kazunari Inoue, Hisashi Nakamura, and Hiroyuki Kawai, "A 10Mbit frame buffer memory with Z-compare and A-blend units," IEEE Journal of Solid-State Circuits, Vol. 30, No. 12, pp. 1563-1568 (1995).

Referenced Paper List

(1) Hideyuki Noda, Kazunari Inoue, Hans Juergen Mattausch, Tetsushi Koide, Katsumi Dosaka, Kazutami Arimoto, Kazuyasu Fujishima, Kenji Anami, and Tsutomu Yoshihara, "Embedded low-power dynamic TCAM architecture with transparently scheduled refresh," IEICE Trans. Electron, Vol. E88-C, pp. 622-629 (2005).

(2) Hideyuki Noda, Kazunari Inoue, Masayuki Kuroiwa, Futoshi Igaue, Kouji Yamamoto, Hans Juergen Mattausch, Tetsushi Koide, Atsushi Amo, Atsushi Hachisuka, Kazuyasu Fujishima, Kenji Anami, and Tsutomu Yoshihara, "A cost effective high-performance Dynamic TCAM with pipelined hierarchical searching and shift redundancy architecture," IEEE Journal of solid-state circuits, Vol. 40, No. 1, pp. 245-263 (2005).

(3) Akira Yamazaki, Takeshi Fujino, Kazunari Inoue, Isamu Hayashi, Hideyuki Noda,

Naoya Watanabe, Fukashi Morishita, Katsumi Dosaka, Kazutami Arimoto, Yoshikazu Morooka, Shinya Soeda, Setsuo Wake, Kazuyasu Fujishima, and Hideyuki Ozaki, "A 0.18um 32Mb embedded DRAM macro for 3D graphics controller," IEICE Trans. Electron, Vol.E85-C, No.9, pp. 1697-1708 (2002).

(4) Hiroyuki Kawai, Yoshitsugu Inoue, Junko Kobara, Robert Streitenberger, Masatoshi Kameyama, Kazunari Inoue, Yasutaka Horiba, and Kazuyasu Fujishima, "A Programmable geometry processor with enhanced four-parallel SIMD type processing core for PC-based 3D graphics," IEICE Trans. Electron, Vol.E85-C, No. 5, pp 1200-1210 (2002).

(5) Masaki Kumanoya, Toshiyuki Ogawa, and Kazunari Inoue, "Advances in DRAM interfaces," IEEE Micro, vol. 15, No. 6, pp. 30-36 (Dec. 1995).