國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

應用於IPv6位址搜尋之高能源效應內容可定址記憶體電路設計

Energy-Efficient Content-Addressable Memory Design For IPv6 Addressing Lookup Application

研究生：張書瑋

指導教授：黃 威 教授

中華民國九十五年七月

應用於 IPv6 位址搜尋之高能源效應內容可定址記憶體電路設計

Energy-Efficient Content-Addressable Memory Design For IPv6 Addressing Lookup Application

研究生：張書瑋　　Student：Shu-Wei Chang

指導教授：黃　威 教授　　Advisor：Prof. Wei Hwang

國立交通大學

電子工程學系電子研究所

碩士論文

A Thesis

Submitted to Department of Electronics Engineering & Institute of Electronics

College of Electrical Engineering and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Electronics Engineering

July 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

應用於 IPv6 位址搜尋之高能源效應內容可定址記憶體電路設計

學生：張書瑋　　　　　　　　　指導教授：黃　威　教授

國立交通大學電子工程學系電子研究所

摘　　要

本論文利用平行交錯式比對與基於無關項的低功率技術，提出了一個全新的高速低功率抗雜訊的三元內容可定址記憶體。利用高度的平行化搜尋，butterfly 結構使得比較線上的搜尋延遲得以降低。利用 IPv6 無關項的特性，可以阻斷比較線上預先充電電路的路徑。並利用搜尋時因無關項造成比較符合時與搜循線上的資料無關的特性，可以阻斷階層式搜尋線之資料傳遞。一個高速低功率 256 行×128 位元之三元內容可定址記憶體亦被提出，利用 TSMC 0.13μm CMOS 技術來實現電路設計與佈局。根據模擬結果顯示，此新的三元內容可定址記憶體在最大操作頻率 500Mhz 可以達到 0.22fJ/bit/search 的能源表現與 0.71ns 的搜尋時間。

Energy-Efficient Content-Addressable Memory Design For IPv6 Addressing Lookup Application

Student : Shu-Wei Chang Advisors : Prof. Wei Hwang

Department of Electronics Engineering & Institute of Electronics
National Chiao-Tung University

ABSTRACT

The new high speed, low-power and noise-tolerant ternary content-addressable memories (TCAMs) using butterfly match-line scheme and don't care based low power technique are realized in this thesis. Butterfly match-line scheme reduces search delay by high parallel search. Furthermore, for IPv6 addressing lookup application, butterfly match-line scheme reduces switching activity also. By use of XOR-based conditional keepers, power consumption and search delay further reduced. The potential charge-sharing problem in the AND match-line could be solved also. Don't-care based low power technique takes the advantage of application future of TCAM. Don't-care based power-gating eliminate unnecessary precharege. Don't-care based hierarchical search-line scheme decrease switching capacitance. A 256-word x 128-bit energy-efficient ternary CAM is also proposed and simulations and layout are implemented in TSMC 0.13μm CMOS technology. Simulation results show that 0.29fJ/bit/search energy performance and 0.65ns search time operate in maximum frequency 500Hhz.

致謝

感謝許多人的幫助，讓我完成了這一篇論文。

首先，我要感謝我的指導教授黃威，在他的指導下讓我對自己研究的領域有更深入的瞭解，建立了研究的興趣與自信心。黃教授提供了一個常優良的研究環境與充足的研究資源，讓我能夠充分發揮自己的能力完成這一篇論文。接著，我要感謝溫瓌岸教授，對我在碩一時的指導線，讓我學到很多關於通訊領域的知識。

感謝實驗室的成員，在過去的兩年當中在生活上與研究上對我的幫助。感謝黃柏蒼、張明宏和楊浩義對於我在研究上的幫忙，讓我能夠有更加優良的研究成果。

最後我要感謝我的家人對我在生活上的關心與幫助，讓我能夠順利的完成碩士的論文研究。

Contents

Chapter 1 Introduction 1

1.1 Background 1

1.2 Motivation..2

1.3 Organization...3

Chapter 2 An Overview of Content Addressable Memory 5

2.1 Conventional CAM Architecture 5

2.2 Conventional CAM Cell 6

2.2.1 Binary CAM Cell...7

2.2.1.1 NOR-type CAM Cell 7

2.2.1.2 AND-type CAM Cell 8

2.2.2 Ternary CAM Cells 9

2.2.2.1 NOR-type TCAM Cell...10

2.2.2.2 AND-type TCAM Cell...11

2.3 Match-line Structure 12

2.3.1 NOR-type Match-line 13

2.3.2 AND-type Match-line 13

2.4 Applications of CAM 15

2.4.1 Cache Memory..15

2.4.2 Translation Look-aside Buffer 16

2.4.3 Packet Forwarding Using CAM..18

2.4.4 ATM Switches...19

2.5 Low power Content-addressable Memory Designs.........................20

2.5.1 Method of reducing Match-line power consumption....................20

2.5.1.1 Low-Swing Scheme 21

2.5.1.2 Current-Race Scheme 21

2.5.1.3 Selective-Precharge Scheme 23

2.5.1.4 Pipelining Scheme 24

2.5.1.5 Current-Saving Scheme 25

2.5.2 Method of reducing Search-line power consumption 26

2.5.2.1 Hierarchical Search-line Scheme 26

2.5.2.2 Charge-Recycling Search-line Driver..28

2.5.3 Power-Saving CAM architecture..30

Chapter 3 Noise Tolerant Butterfly Mach-line Scheme....................34

3.1 On-Chip Noise Sources...34

3.1.1 Cross-Couple Noises..34

3.1.2 Charge Sharing Noises...35

3.1.3 Leakage Noises 36
3.1.4 Power Rail Bouncing Noises 36
3.1.5 Substrate Noises 36
3.2 Noise-Tolerant Circuits Design 36
3.2.1 Employing Keepers 37
3.2.2 Pre-charge Internal Nodes 38
3.2.3 Raising Source Voltage 39
3.2.4 Creating Complementary P-Network 40
3.3 XOR-Based Conditional Keeper 41
3.4 Noise-Tolerant Match-line Schemes for CAM 43
3.4.1 NOR-type Noise-Tolerant Match-line Scheme 44
3.4.2 AND-type Noise-Tolerant Match-line Scheme 44
3.5 Noise-Tolerant Butterfly Match-line Scheme Design 45
3.5.1 Pseudo-Footless Clock-and-Data Pre-charge Dynamic (PF-CDPD) Circuits 45
3.5.2 Conventional Pipelined AND-type Match-line Scheme 47
3.5.3 Butterfly Match-line Scheme 48
3.5.3.1 Architecture 48
3.5.3.2 Circuit Implementation 50
3.5.4 Analysis of Power Consumption 51
3.5.4.1 Conventional AND-type Match-line 52
3.5.4.2 Tree-style AND-type Match-line 53
3.5.4.3 Butterfly AND-type Match-line 53
3.5.5 Performance Comparisons 56
3.5.5.1 Search Time 56
3.5.5.2 Power Consumption 58
3.6 Summary 59
Chapter 4 Don’t-care Based Low Power Technique 60
4.1 IPv6 Addressing Architecture 60
4.1.1 Unspecified Unicast Addresses 61
4.1.2 Loopback Unicast Addresses 61
4.1.3 Global Unicast Addresses 62
4.1.4 Local-Use IPv6 Unicast Addresses 62
4.2 IP Address Lookup 63
4.3 Don’t-care Based Power-gating Technique 65
4.4 Don’t-care based Hierarchical Search-line scheme 68
4.4.1 Architecture 68
4.4.2 Timing Analysis 70

4.5 Performance Comparisons ... 70

4.5.1 IPv6 prefix length distribution in the router ... 70

4.5.2 Simulation Result ... 72

4.5.2.1 Don’t-care Based Power-gating Technique ... 72

4.5.2.2 Don’t-care Based Hierarchical Search-line ... 76

4.6 Summary ... 79

Chapter 5 Energy-Efficient Ternary Content-Addressable Memory for IPv6 Addressing Lookup Application ... 80

5.1 Power Sources in Digital CMOS Circuits ... 81

5.1.1 Dynamic Power ... 81

5.1.2 Short-Circuit Power ... 81

5.1.3 Leakage Power ... 81

5.1.4 Low-Power CAM Design ... 82

5.2 MOSFET Structure Capacitances ... 82

5.2.1 Gate Capacitances ... 83

5.2.1.1 Overlap Capacitances ... 83

5.2.1.2 Gate-to-Channel Capacitances ... 84

5.2.2 Junction Capacitances ... 85

5.2.2.1 Bottom-Plate Junction Capacitances ... 86

5.2.2.2 Side-Wall Junction Capacitances ... 86

5.3 Proposed Ternary Content-Addressable Memory ... 86

5.3.1 Architecture ... 87

5.3.2 Match-line Scheme ... 88

5.3.2.1 Butterfly AND-type match-line scheme for IPv6 ... 88

5.3.2.2 Applying Don’t-care based Power-Gating Technique ... 90

5.3.3 Ternary CAM Cell ... 90

5.3.4 Don’t-care based hierarchical search-line scheme for IPv6 ... 92

5.4 Simulation Results ... 93

5.5 Layout and the Post-Simulations ... 96

5.6 Summary ... 99

Chapter 6 Conclusions and Future Work ... 100

6.1 Conclusions ... 100

6.2 Future Work ... 101

Bibliography ... 103

List of Figures

Fig. 2.1 Conventional CAM Architecture 6

Fig. 2.2 NOR-type binary CAM cell. (a) 9-transistor BCAM cell and (b) 10-transistor BCAM cell. 8

Fig. 2.3 AND-type 9-transistor binary CAM cell. 9

Fig. 2.4 Static NOR-type ternary CAM cell. 10

Fig. 2.5 Dynamic NOR-type ternary CAM cell. 11

Fig. 2.6 AND-type ternary CAM cell. 12

Fig. 2.7 Conventional CAM word circuits. (a) NOR-type and (b) AND-type 14

Fig. 2.8 A simple cache memory. 16

Fig. 2.9 A simple virtual memory system. 17

Fig. 2.10 CAM-based implementation of the routing table of Table 2.5 19

Fig. 2.11 ATM switch with CAM. 20

Fig. 2.12 For current-race match-line sensing [48]: (a) a schematic of the circuit implementation including precharge circuitry and (b) a timing diagram for a single search cycle 23

Fig. 2.13 Pulsed NAND-NOR ML architecture 24

Fig. 2.14 Pipelined match-lines reduce power by shutting down after a miss in a stage 25

Fig. 2.15 Current-saving matchline-sensing scheme [58], [59]. 26

Fig. 2.16 Schematic of the hierarchical search-line architecture. 28

Fig. 2.17 Charge-recycling search-line driver 29

Fig. 2.18 Block diagram of hybrid-type TCAM architecture. 30

Fig. 2.19 Block diagram of main bank. 31

Fig. 2.20 Sub/extra bank structure. 32

Fig. 2.21 Schematic of match-line repeater. 32

Fig. 2.22 (a) Partial match (PM) block. (b) Main match (MM) block and local priority encoding (LPE). 33

Fig. 3.1 High fan-in dynamic multiplexer circuit and the presence of noise sources. 34

Fig. 3.2 Diagram of cross-couple noises. (a) Negative voltage excursion. (b) Positive voltage excursion. 35

Fig. 3.3 Applying keeper techniques to improve the noise immunity. (a) Weak always on keeper. (b) Conventional feedback keeper. (c) Weak keeper. (d) Smart keeper. (e) HS feedback keeper. (f) STHS feedback keeper. (g) Conditional keeper. 38

Fig. 3.5 Applying rising source voltage techniques to improve the noise immunity.

(a) PMOS pull-up. (b) NMOS pull-up. (c) Mirror techniques. (d) Twin transistors techniques. 39

Fig. 3.8 CMOS inverter techniques with direct DC path. (a) A 3-input OR-AND gate. (b) Direct conducting path. 40

Fig. 3.9 The XOR gate implementation. 42

Fig. 3.10 Timing diagram for XOR-based conditional keeper. 42

Fig. 3.11 Match-line schemes for CAM. (a) NOR-type. (b) AND-type. 43

Fig. 3.12 NOR-type match-line scheme with XOR-based keeper. 44

Fig. 3.13 AND-type match-line scheme with XOR-based keeper. 44

Fig. 3.14 Transfer four-input AND gate logic into clock-and-data pre-charge dynamic (CDPD) circuits. 45

Fig. 3.15 Pseudo-footless clock-and-data pre-charge dynamic (PF-CDPD) circuits. 46

Fig. 3.16 Two types of pipelined AND-type match-line scheme. (a) CAML I (b) CAML II 47

Fig. 3.17 Waveforms of two types of CAML in the search operation. 48

Fig. 3.18 Power comparison for two types of CAML. 48

Fig. 3.19 Butterfly match-line scheme. (a) Type I (a) Type II (a) Type III 49

Fig. 3.20 Circuit implementation of Type I butterfly match-line scheme with conventional keeper 51

Fig. 3.21 Circuit implementation of Type I butterfly match-line scheme with XOR-based conditional keeper 51

Fig. 3.22 2-level tree-style match-line scheme 53

Fig. 3.23 A mismatch example of Type I butterfly match-line scheme 54

Fig. 3.24 A mismatch example of Type II butterfly match-line scheme 54

Fig. 3.25 A mismatch example of Type III butterfly match-line scheme 55

Fig. 3.26 Timing diagram of CAM during the search operation 57

Fig. 3.27 Search time comparisons for three types butterfly match-line scheme 57

Fig. 3.28 Power consumptions for three types butterfly match-line scheme 58

Fig. 3.29 Normalized power consumptions for three types butterfly match-line scheme 58

Fig. 4.1 The general format for IPv6 global unicast addresses 62

Fig. 4.2 The format of local-use IPv6 unicast addresses 63

Fig. 4.3 Packet routing based on longest prefix matching mechanism 65

Fig. 4.4 AND-type TCAM match-line scheme with XOR-based conditional keeper 66

Fig. 4.5 Timing diagram of CAM segment when all TCAM cells are don’t-care. 66

Fig. 4.6 Don’t-care based power-gating match-line scheme. 67

Fig. 4.7 Timing diagram of CAM segment with don’t-care based power-gating technique 67
Fig. 4.8 A simplified architecture of don’t-care based hierarchical search-line scheme 69
Fig. 4.9 Circuit implementation of don’t-care based hierarchical search-line scheme 69
Fig. 4.10 Timing analysis of don’t-care based hierarchical search-line scheme...... 70
Fig. 4.11 Network address prefix distribution of IPv6 71
Fig. 4.12 Performance of Type I butterfly match-line scheme with don’t-care based power-gating technique 72
Fig. 4.13 Performance of Type II butterfly match-line scheme with don’t-care based power-gating technique 73
Fig. 4.14 Performance of Type III butterfly match-line scheme with don’t-care based power-gating technique 74
Fig. 4.15 Performance analysis of butterfly match-line scheme without don’t-care based power-gating technique 75
Fig. 4.16 Performance analysis of butterfly match-line scheme with don’t-care based power-gating technique 75
Fig. 4.17 Performance of Type I butterfly match-line scheme with don’t-care based hierarchical search-line 76
Fig. 4.18 Performance of Type II butterfly match-line scheme with don’t-care based hierarchical search-line 77
Fig. 4.19 Performance of Type III butterfly match-line scheme with don’t-care based hierarchical search-line 77
Fig. 4.20 Performance analysis of butterfly match-line scheme without don’t-care based hierarchical search-line 78
Fig. 4.21 Performance analysis of butterfly match-line scheme with don’t-care based hierarchical search-line 78
Fig. 5.1 MOSFET overlap capacitances. 83
Fig. 5.2 The gate-to-channel capacitance and how the operation region influences its distribution over the three other terminals. (a) Cut-off. (b) Resistive. (c) Saturation. 84
Fig. 5.3 Structure of source junction. 85
Fig. 5.4 CAM array divided into two sub-arrays. 88
Fig. 5.5 IP addressing lookup table based on CAM. 89
Fig. 5.6 Butterfly match-line scheme for IPv6 addressing lookup application 89
Fig. 5.7 Circuit implementation of proposed match-line scheme. 90
Fig. 5.8 16-transistor AND-type TCAM cell. (a) Bit-line and Search-line are

shared. (b) Bit-line and Search-line are separated. 91
Fig. 5.9 Block diagram of proposed hierarchical search-line scheme 92
Fig. 5.10 Performance of Type I butterfly match-line scheme with don't-care based low power technique. 93
Fig. 5.11 Performance of Type II butterfly match-line scheme with don't-care based low power technique. 94
Fig. 5.12 Performance of Type III butterfly match-line scheme with don't-care based low power technique. 95
Fig. 5.13 Performance Analysis of proposed TCAM 95
Fig. 5.14 Search time comparisons for three types butterfly match-line scheme 96

List of Tables

Table 2.1 Truth table of NOR-type binary CAM cell. 8
Table 2.2 Truth table of AND-type binary CAM cell. 9
Table 2.3 State assignments and truth table for static TCAM cell. 10
Table 2.4 State assignments for TCAM cell. 12
Table 2.5 Example Routing Table 19
Table 2.6 Power consumption of CRSLD compare to other SL schemes 30
Table 3.1 Control organism of XOR-based conditional keeper. 41
Table 3.2 Performance comparison for butterfly match-line scheme and conventional AND-type match-line scheme 59
Table 4.1 IPv6 address type identification 61
Table 4.2 Prefix length distribution in the router of 6Bone 71
Table 5.1 Average distribution of channel capacitance of MOS transistor for different operation regions. 85
Table 5.2 State assignments for TCAM cell. 91
Table 5.3 Summary of the proposed TCAM scheme. 99

Chapter 1
Introduction

1.1 Background

Content-addressable memory (CAM), also called associative memory, is a memory that implements the lookup-table function in a single clock cycle using dedicated comparison circuitry. In past decades, CAM is used in a wide variety of applications requiring high speed search ability. These applications are parametric curve extraction [1], Hough transformation [2], Huffman coding/decoding [3], [4], Lempel–Ziv compression [5]–[8], and image coding [9]. At present, CAM is popular in network routers for packet forwarding, packet classification, asynchronous transfer mode (ATM) switches, and so on.

A CAM cell contains 1-bit storage memory and 1-bit comparison circuit. Based on different forms of storages, it could be divided into two type of CAM cells. The first one is called static CAM cell which is implemented by SRAM, and the other one is called dynamic CAM cell which consists of capacitance. A conventional CAM cell, also called binary CAM (BCAM) cell, has two states: “one” state and “zero” state. Therefore, BCAM cell just needs one storage memory to store data. On the other hand, there exists another kind of CAM cell, called ternary CAM (TCAM) cell. Different from BCAM cell, the TCAM has an extra state, don’t-care, which is suitable to be used in network applications. Because TCAM cell has this extra state, TCAM cell needs an additional storage memory to store data.

According to match-line scheme of CAM, there are two different types of match-line scheme. While stored data is not exactly the same as search data in every bits, the match-line would be discharged to ground. This match-line is called

NOR-type match-line scheme. The other one is AND-type match-line scheme.[1] With AND-type match-line scheme, the match-line would be discharged to ground only when all bits of stored data match with all bits of search data. Generally, NOR-type match-line scheme has shorter search time but consumes more search power. On the contrary, AND-type match-line scheme consumes less search power but needs longer search time to complete each search operation.

1.2 Motivation

Energy-efficient design is essentially important for the future System-on-Chip (SoC) design. CAM compares input search data against a table of stored information, and returns the address of the matching data [10]-[14]. CAMs have a single clock cycle throughput making them faster than other hardware- and software-based search systems. However, the speed of a CAM comes at the cost of increased silicon area and power consumption. As CAM applications grow, the larger CAM size demanding, the further exacerbated power problem. Reducing power consumption, without sacrificing speed or area, is the main thread of recent research topic in large-capacity CAMs.

Internet Protocol Version 6 (IPv6) [15] is the "next generation" protocol designed by the IETF to replace the current version Internet Protocol, IP Version 4 ("IPv4"). IPv6 addresses are 128-bit identifiers for interfaces and sets of interfaces [16]. Each address needs 128 TCAM cells to store data results a long search delay path in network routers for packet forwarding application. As mentioned before, AND-type TCAM has more search power saving but longer search time is the price. Thus, reducing the search time by modifying TCAM architecture becomes the inevitable issue. Furthermore, large amounts of TCAM cells result in large capacitance on search-line which could cause large power consumptions.

With the advance of technology and SoC design, coupling noise, charge sharing and power/ground fluctuation noises are increasing the soft-error rate of dynamic circuits. Since AND-type match-line scheme belongs to dynamic circuits, noise immunity is a serious issue for CAM design. Recently, several techniques have been

[1] Because the principles and operations of AND-type match-line schemes like ones of NAND-type match-line schemes, we call NAND-type match-line scheme as AND-type match-line scheme in this thesis.

developed to improve the noise immunity in dynamic circuits. However, all of them have extra price, like more power consumptions, longer propagation delays, and larger area overheads.

According to above issues, this thesis focuses on the techniques of search time and dynamic power reduction in an energy-efficient TCAM. Moreover, we also care about the noise immunity of match-line schemes. A noise-tolerant butterfly match-line scheme and don’t-care based low power technique which can reduce search time and dynamic power would be applied to TCAM design for IPv6 application.

1.3 Organization

The organization of this thesis is as follows. An overview of CAM is introduced in Chapter 2. Here, a conventional CAM architecture including CAM cells and CAM word schemes would be presented. Besides, the application and prior arts of CAM would be described in this chapter as well.

The noise-tolerant butterfly match-line scheme is realized in Chapter 3. The proposed butterfly match-line scheme with XOR-based conditional keeper [17] achieves the search time reduction and search power at the same unity noise gain margin compared to conventional AND-type match-line scheme. Furthermore, at the same search time criteria, the AND-type match-line scheme not only has better performance but also saves area compared to conventional AND-type match-line scheme.

Don’t-care based low power technique is proposed in Chapter 4. Although there does have area overhead, with increasing bits of match-line scheme, the overhead is becoming acceptable. By use the characteristic of addresses of IPv6, match-line precharge circuit can be turned off according to numbers of don’t-care bits in one search stage scheme on match-line. Furthermore, hierarchical search-line scheme controlled by don’t-care bits are proposed to reduce switching capacitances on search-line. As a result, power consumption can be reduced.

An energy-efficient ternary CAM array is implemented in Chapter 5. In this chapter, we reduce the switching capacitances and switching activities to decrease dynamic power consumption. Butterfly match-line scheme which decrease search delay is also utilized. A novel AND-type match-line scheme which combines butterfly match-line scheme, don’t-care based low power technique, XOR-based conditional keeper, pseudo-footless clock and data pre-charge dynamic (PF-CDPD) circuit [18] is proposed. TCAM array composed of the proposed AND-type match-line scheme has shorter search time and less search power consumption. Moreover, there is only a little area overhead about the proposed TCAM scheme. Finally, the overall investigation results and conclusions will be discussed in Chapter 6.

Chapter 2
An Overview of Content Addressable Memory

In this chapter, it introduces the overview of CAM. The conventional CAM architecture, CAM cell circuits, and CAM word schemes would be described in Section 2.1, Section 2.2, and Section 2.3, respectively. In addition, the applications of CAM would be detailed in Section 2.4. Finally, Section 2.5 would give an introduction for low power content-addressable memory design.

2.1 Conventional CAM Architecture

A conventional CAM architecture is usually composed of the data memories, address decoders, bit-lines pre-charge circuits, word match schemes, read sense amplifiers, address priority encoders and so on [19]-[21]. Fig. 2.1 shows a simplified block diagram of a CAM. Generally, CAM has three operation modes: write, read, and search. In write and read operation, CAM plays just like an ordinary memory. That is to say, data is manipulated in the CAM array as the same way in SRAM array. Different from SRAM, CAM has a special mode: search mode. The input in Fig. 2.1 called search word that is broadcast onto the search-lines to the table of stored data. The number of bits in a CAM word is usually large, with existing implementations ranging from 36 to 144 bits. A typical CAM employs a table size ranging between a few hundred entries to 32K entries, corresponding to an address space ranging from 7 bits to 15 bits. Each stored word has a match-line that indicates whether the search word and stored word are identical (the match case) or are different (a mismatch case, or miss). The match-lines are fed to an encoder that generates a binary match location corresponding to the match-line that is in the match state. An encoder is used in systems where only a single match is expected. In CAM applications where more than one word may match, a priority encoder is used instead of a simple encoder. A priority encoder selects the highest priority matching location to map to the match result, with words in lower address locations receiving higher priority. The overall function of a CAM is to take a search word and return the matching memory location. One can think of this operation as a fully programmable arbitrary mapping of the large space of the input search word to the smaller space of the output match location.

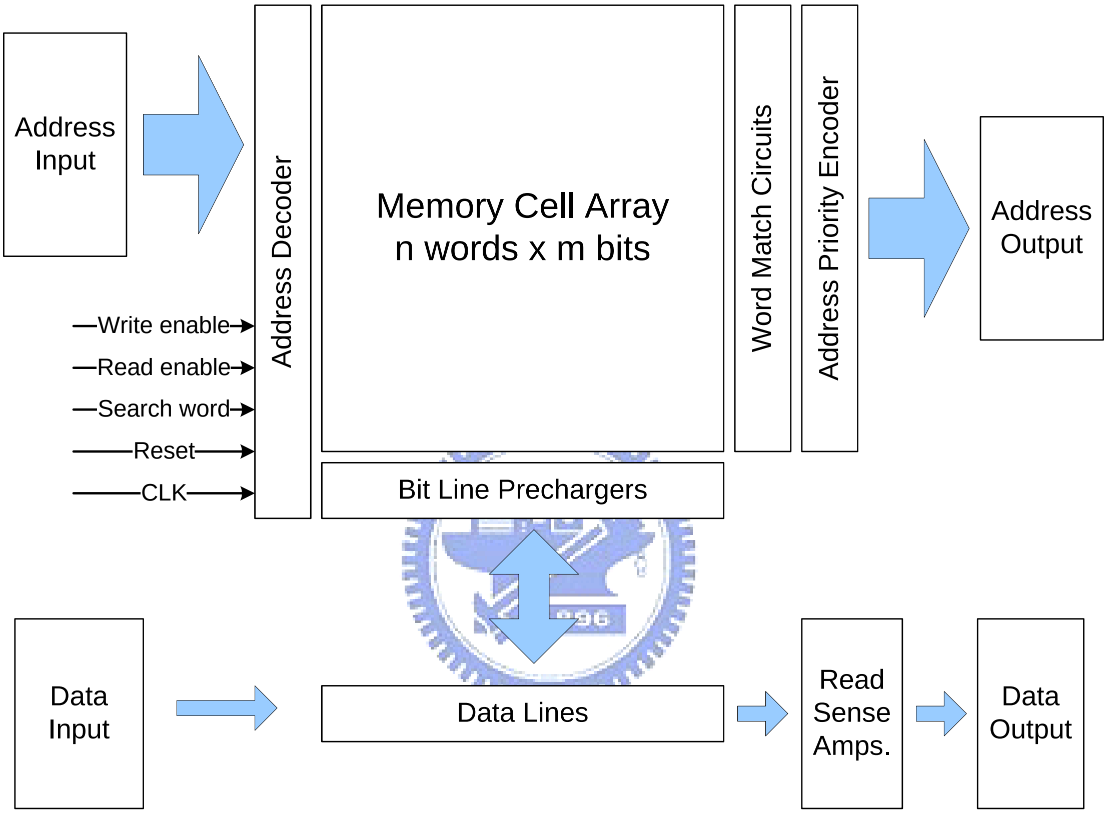


Fig. 2.1 Conventional CAM Architecture

2.2 Conventional CAM Cell

In this section, a conventional CAM cell would be introduced. A CAM cell serves two basic functions: bit storage (as in RAM) and bit comparison (unique to CAM). There are two types of CAM cells would be introduced as following: one is binary CAM (BCAM) cell and the other is ternary CAM (TCAM) cell.

2.2.1 Binary CAM Cell

Depending upon working different methods in search mode, CAM[2] cells are classified into two kinds: NOR-type CAM cell and AND-type[3] CAM cell [22]. The differences of them would be described as follows.

2.2.1.1 NOR-type CAM Cell

Fig. 2.2 depicts the NOR-type CAM cells which are widely used for CAM scheme design in past years. Fig. 2.2 (a) is constructed by 9-transistor structure and Fig. 2.2 (b) is composed of 10-transistor structure. Table 2.1 shows the truth table of a NOR-type CAM cell. The 9-transistor CAM cell consists of a traditional 6-transistor SRAM and a PTL-type compare circuit; the 10-transistor CAM cell is composed of an ordinary 6-transistor SRAM and the pull down XOR comparison circuits. As the CAM cell is to be written, not only 9-transistor CAM cell but also 10-transistor CAM cell work same as a SRAM cell. While word-line is active, the complementary data is forced onto the bit-lines to be stored in the D-latch which is composed of two inverters. In read operation, bit-lines will be pre-charged to high first and whether the bit-lines discharge to ground or not depends on stored data. After passing the read sense amplifier, the correct data is sent to the output stage. About the 9-transistor, the match-line will be charged to high first in the search operation. If search data is equal to the stored data, the node X becomes low. Furthermore, the NMOS, Mn, is turned off, and the match-line is still floating. On the other hand, if search data doesn't match with stored data, the node X would become high and result in the NMOS, Mn, being turned on. Therefore, the match-line would be discharged to ground. Regarding 10-transistor CAM cells, the principle is same as 9-transistor CAM cells. During searching operation, the match-line would be pre-charged to high first. If searching data is equal to the stored data, the match-line is still floating. Contrarily, if searching data is not equal to the stored data, there is a path from match-line to ground and match-line would be discharged to ground through this path.

[2] In this thesis, we call binary CAM as CAM simply. If we mean the ternary CAM, we will call TCAM specially.

[3] Because the principles and operations of AND-type cells like ones of NAND-type cells, we all call NAND-type match-line scheme as AND-type match-line scheme in this thesis.

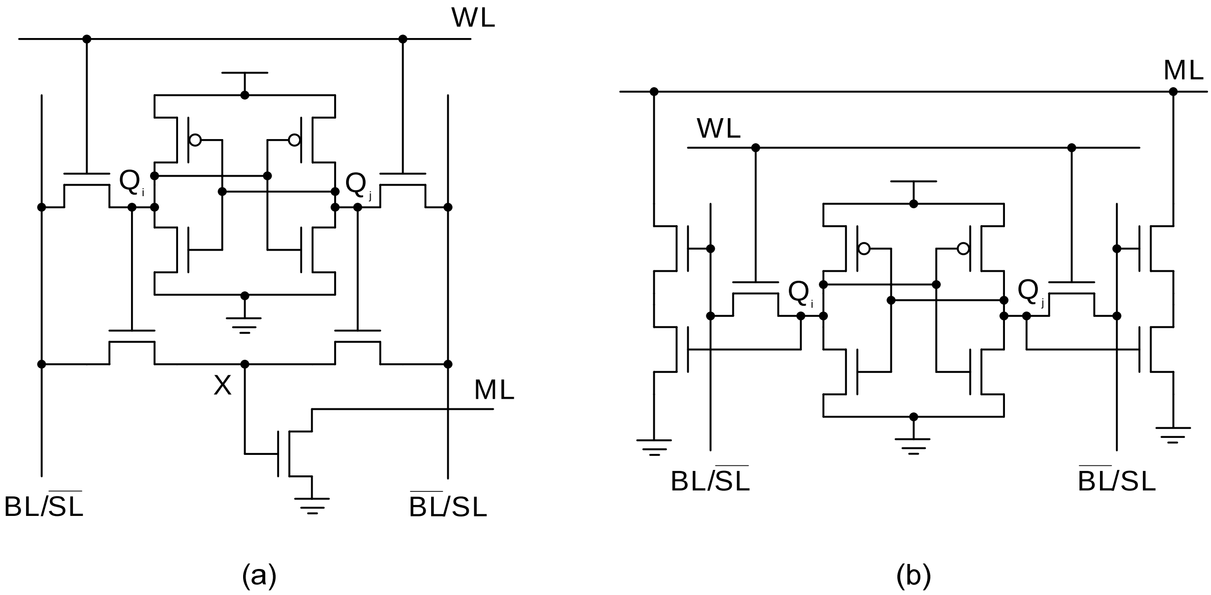


Fig. 2.2 NOR-type binary CAM cell. (a) 9-transistor BCAM cell and (b) 10-transistor BCAM cell.

Table 2.1 Truth table of NOR-type binary CAM cell.

State	Q_i	SL	ML
Zero (0)	0	0	floating
	0	1	0
One (1)	1	0	0
	1	1	floating

2.2.1.2 AND-type CAM Cell

An AND-type CAM cell is similar to 9-transistor CAM cell whatever it works in write or read operation. The only one difference from 9-transistor CAM cell is the match-line scheme. Fig. 2.3 depicts an AND-type CAM cell and Table 2.2 describes the truth table of AND-type CAM cell. As an AND-type CAM cell works in search operation, the match-line would be pre-charged to high first. However, contrary to 9-tansistor NOR-type CAM cell, the match-line hold floating when the search data doesn’t match with stored data and the match-line is discharged to ground only while the search data and stored data are match.

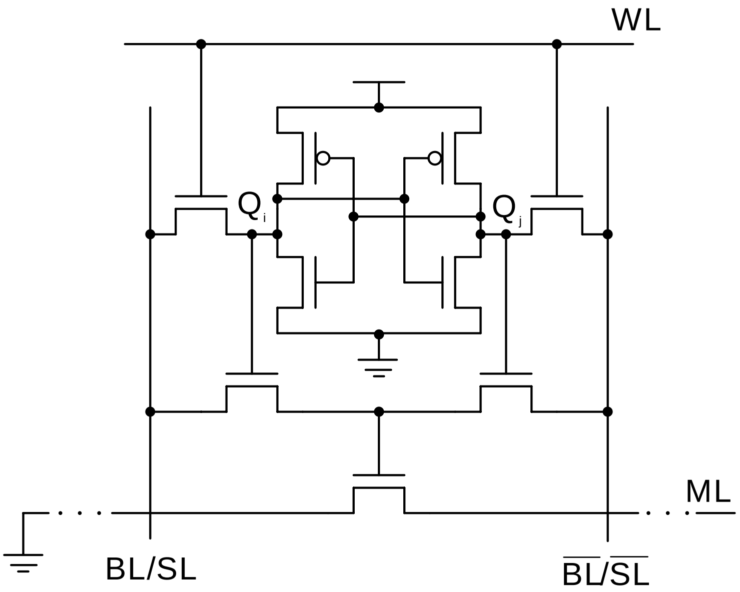


Fig. 2.3 AND-type 9-transistor binary CAM cell.

Table 2.2 Truth table of AND-type binary CAM cell.

State	Q_i	SL	ML
Zero (0)	0	0	0
	0	1	floating
One (1)	1	0	floating
	1	1	0

2.2.2 Ternary CAM Cells

For the CAM circuit design, the ternary CAM (TCAM) performs a more powerful data search function [23]-[24]. Different from binary CAM which has two states: one (1) and zero (0) state, the ternary CAM (TCAM) cell has an additional state: don't care (X) state. Alike binary CAM cell, TCAM would be classified into two kinds: NOR-type TCAM cell and AND-type TCAM cell. Both of them would be introduced in following sections.

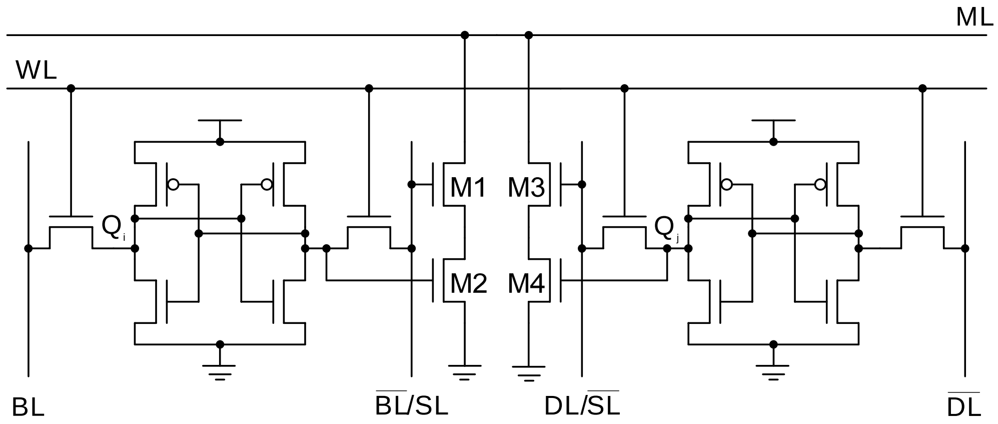


Fig. 2.4 Static NOR-type ternary CAM cell.

Table 2.3 State assignments and truth table for static TCAM cell.

State	**Q$_i$**	**Q$_j$**	**SL**	**ML**
Zero (0)	0	1	0	floating
	0	1	1	0
One (1)	1	0	0	0
	1	0	1	floating
Don’t care (X)	0	0	0	floating
	0	0	1	floating
Not allowed	1	1	0	—
	1	1	1	—

2.2.2.1 NOR-type TCAM Cell

Fig. 2.4 shows a static NOR-type TCAM cell. It consists of 2-SRAM and 4-transistor comparison circuits. This TCAM cell is designed to store three states, namely zero (0), one (1) and don’ care (X). These three states are set by Q$_i$ and Q$_j$. Table 2.3 illustrates how the three states are stored in this TCAM cell and the truth table of the static NOR-type TCAM cell. When Q$_i$ is low and Q$_j$ is high, the TCAM cell is in the “zero” state. In the searching operation, the same as BCAM cell, match-line will be charged to high first. If search data is low, the NMOS M1 and M4 would not be turned on, such that the ML will still be floating. On the other hand, while search data is high, the NMOS M1 and M2 are turned on at the same time result in the match-line being discharged to ground. However, the TCAM cell is in the “one” state, while search data is high, the match-line would keep high. While search data is low, the match-line would be discharge to the ground. Particularly, while Q_i and Q_j are both low, the TCAM cell is in “don’t care” state. No matter search data is high or is low, the NMOS M1 and M3 are not turned on result in the match-line keeping floating. Note that Q_i and Q_j cannot be high simultaneously, this state are not be allowed.

There is an additional dynamic NOR-type TCAM cell is called dynamic TCAM cell [25]-[29], as shown in Fig. 2.5. The major difference between static TCAM cell and dynamic TCAM cell is that the storage memories composed of 2 SRAM cells in static TCAM cell are replaced by 2 capacitances in dynamic TCAM cell. The dynamic TCAM cell works like static TCAM and Table 2.3 also shows how these three states are stored in this dynamic TCAM cell and the truth table of the dynamic TCAM cell.

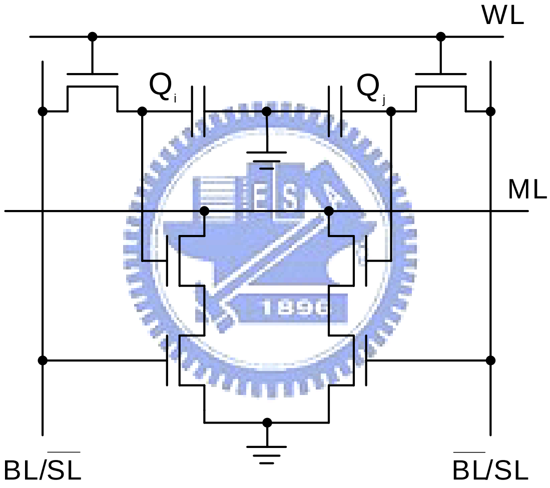


Fig. 2.5 Dynamic NOR-type ternary CAM cell.

2.2.2.2 AND-type TCAM Cell

Fig. 2.6 illustrates a 16-transistor AND-type TCAM cell which includes 2-SRAM and comparison circuits composed of three NMOS. The state assignments and truth table of this TCAM cell is described in Table 2.4. The AND-type TCAM cell is alike a 9-transistor AND-type BCAM cell when TCAM cell works in zero (0) and one (1) states. However, while this AND-type TCAM cell is in don’t care (X) state (Q_j is high), no matter the search data is high or low, the match-line would be discharged.

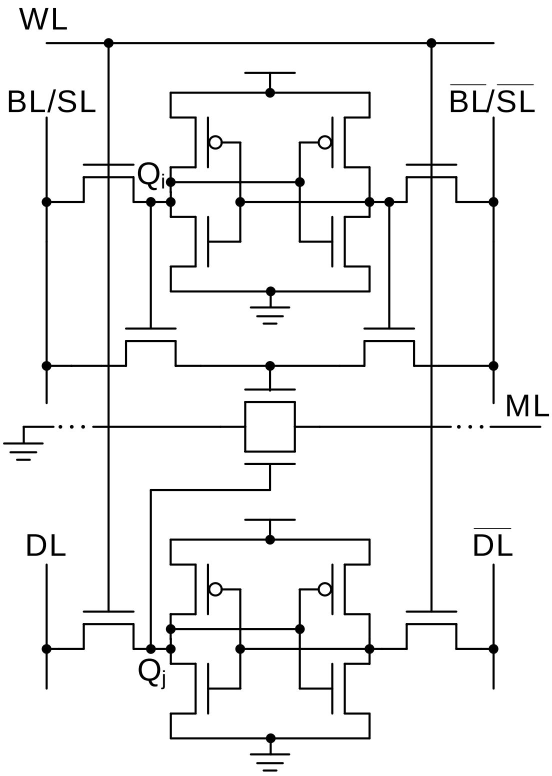


Fig. 2.6 AND-type ternary CAM cell.

Table 2.4 State assignments for TCAM cell.

State	**Q_i**	**Q_j**	**SL**	**ML**
Zero (0)	0	0	0	0
	0	0	1	floating
One (1)	1	0	0	floating
	1	0	1	0
Don't Care (X)	0	1	0	0
	0	1	1	0
	1	1	0	0
	1	1	1	0

2.3 Match-line Structure

In the conventional CAM architecture, the circuit design of CAM word circuits adopts dynamic CMOS circuits to improve data matching performance and hardware cost. Applying the dynamic CMOS circuits designs, the conventional NOR-type CAM word schemes and AND-type match-line schemes are shown in Fig. 2.7 (a) and Fig. 2.7 (b), respectively [30].

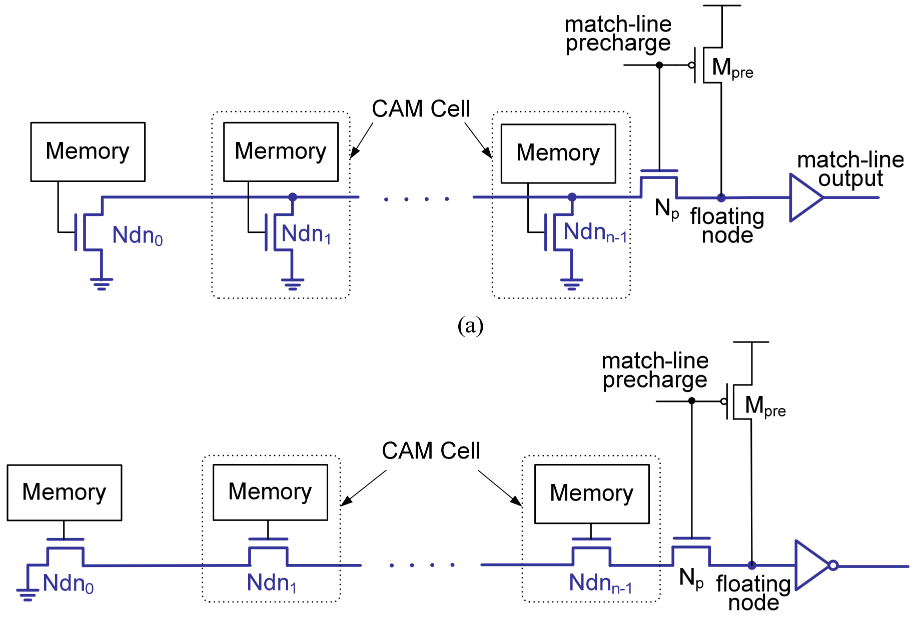
2.3.1 NOR-type Match-line

Fig. 2.7 (a) depicts, in schematic form, how NOR cells are connected in parallel to form a NOR match-line, ML. While we show binary cells in the figure, the description of match-line operation applies to both binary and ternary CAM. A typical NOR search cycle operates in three phases: search-line precharge, match-line precharge, and match-line evaluation. First, the search-lines are precharged low to disconnect the match-lines from ground by disabling the pull down paths in each CAM cell. Second, with the pull down paths disconnected, the M_{pre} transistor precharges the match-lines high. Finally, the search-lines are driven to the search word values, triggering the match-line evaluation phase. In the case of a match, the ML voltage, V_{ML} stays high as there is no discharge path to ground. In the case of a miss, there is at least one path to ground that discharges the match-line. The match-line sense amplifier (MLSA) senses the voltage on ML, and generates a corresponding full-rail output match result. We will see several variations of this scheme for evaluating the state of NOR match-lines in Section III. The main feature of the NOR match-line is its high speed of operation. In the slowest case of a one-bit miss in a word, the critical evaluation path is through the two series transistors in the cell that form the pull down path. Even in this worst case, NOR-cell evaluation is faster than the NAND case, where between 8 and 16 transistors form the evaluation path.

2.3.2 AND-type Match-line

Fig. 2.7 (b) shows the structure of the AND match-line. A number of AND CAM cells are cascaded to form the ML (this is, in fact, a floating node, but for consistency we will refer to it as ML). For the purpose of explanation, we use the binary version of the AND cell, but the same description applies to the case of a ternary cell. On the right of the figure, the precharge pMOS transistor, M_{pre} sets the initial voltage of the ML to the supply voltage. Next, the evaluation nMOS transistor, $N_{p,}$ turns ON. In the case of a match, all nMOS transistors are ON, effectively creating a path to ground from the ML, hence discharging ML to ground. In the case of a miss match, at least one of the series nMOS transistors is OFF, leaving the ML voltage high. The AND match-line has an explicit evaluation transistor, N_p, unlike the NOR match-line, where the CAM cells themselves perform the evaluation.

There is a potential charge-sharing problem in the AND match-line. Charge sharing can occur between the ML and the intermediate nodes. For example, in the case where all bits match except for the leftmost bit in Fig. 2.7 (b), during evaluation there is charge sharing between the ML and nodes Ndnn-1 through Ndn1 . This charge sharing may cause the ML voltage to drop sufficiently low such that the output inveter detects a false match. A technique that eliminates charge sharing is to precharge high, in addition to ML, the intermediate match nodes. This procedure eliminates charge sharing, since the intermediate match nodes and the ML node are initially shorted. However, there is an increase in the power consumption. A feature of the AND match-line is that a miss stops signal propagation such that there is no consumption of power past the final matching transistor in the serial nMOS chain. Typically, only one match-line is in the match state, consequently most match-lines have only a small number of transistors in the chain that are ON and thus only a small amount of power is consumed. Two drawbacks of the AND match-line are a quadratic delay dependence on the number of cells, and a low noise margin.



(a)

(b)

Fig. 2.7 Conventional CAM word circuits. (a) NOR-type and (b) AND-type

2.4 Applications of CAM

CAMs are widely used in cache memory system and translation look-aside buffer (TLB) in virtual memory system in past years. The primary commercial application of CAMs today is to classify and forward Internet protocol (IP) packets in network routers [31]-[36]. In networks like the Internet, a message such an as e-mail or a Web page is transferred by first breaking up the message into small data packets of a few hundred bytes, and, then, sending each data packet individually through the network. These packets are routed from the source, through the intermediate nodes of the network (called routers), and reassembled at the destination to reproduce the original message. The function of a router is to compare the destination address of a packet to all possible routes, in order to choose the appropriate one. A CAM is a good choice for implementing this lookup operation due to its fast search capability.

2.4.1 Cache Memory

In the memory hierarchy system, cache plays an important role [37]-[38]. Cache is the name given to the first level of the memory hierarchy encountered once the address leaves the CPU. Its function is used to refer to any storage managed to take advantage of locality of access. Cache serves as a method for providing fast reference to recently used portion of instruction or data. When CPU finds a wanted data item in the cache, it is called cache hit. On the contrary, if CPU does not find a data item that is needed in the cache, it is called cache miss. Temporal locality means that the requested data item is likely needs it again in the near future, so it is useful to place the requested data item in the cache where it can be accessed quickly. A fixed-size collection of data items which contains the requested data item is called block. There is high possibility that the other data items in the block will be needed soon for spatial locality.

An example for direct data mapping cache is illustrated in Fig. 2.8. The address has 32 bits, and it is divided into three parts. First one part is byte offset which occupies two bits. Second part is Index, and third part is Tag. The numbers of Index can tell us the capacity of cache. If there are N bits for Index, the cache has 2^N entries which can be stored data items. The action is first to find the corresponding position of index. When the corresponding position is found out, the tag stored in the corresponding position would be taken out. This tag would be compared to the third part of tag. If they are the same, and valid bit is one, a hit signal and the corresponding data would be sent out. Of course, the tag entries are composed of CAM array. The valid bit is used to indicate whether an entry contains a valid address or not. If they are not the same, a miss occurs. It means that no requested data in the cache. The wanted data may be stored in the lower level memory. When the wanted data is found in the lower level, it would be written back to the cache.

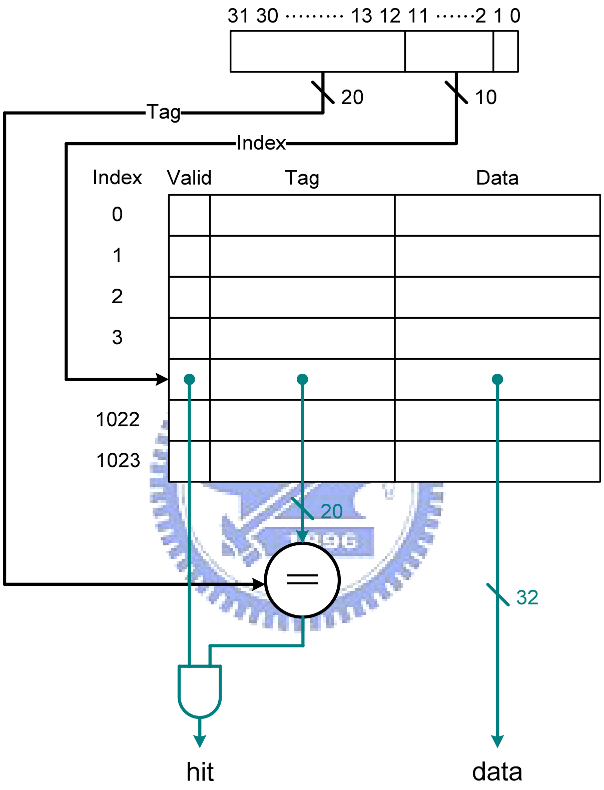


Fig. 2.8 A simple cache memory.

2.4.2 Translation Look-aside Buffer

Translation look-aside buffer (TLB) is widely used to virtual memory system. A TLB is like a cache that hold only page table mapping [37]-[38]. Its function is to provide fast translation from the virtual address to the physical address. When we get the physical address, we can use this physical address to access the data which are stored in the memory (such as cache or DRAM or DISK). Because TLB can speed up address translation in processor with virtual memory and it also can cut down access time and lowering the miss rates.

Fig. 2.9 shows a simple virtual memory system. The TLB contains a subset of the virtual-to-physical page mappings that are in the page table. Because the TLB is a cache, it must have a tag field which consists of CAMs. As a virtual page number (VPN) is sent to the TLB, this VPN would be compared with all valid tags in TLB. If the VPN can find a corresponding tag in the TLB, the corresponding physical page address would find in the corresponding tag. However, if there is no matching entry in the TLB for a page, the page table must be examined. The page table either supplies a physical page number for the page or indicates that the page resides on disk, in which case a page fault occurs. Since the page table has an entry for every virtual page, no tag field is needed.

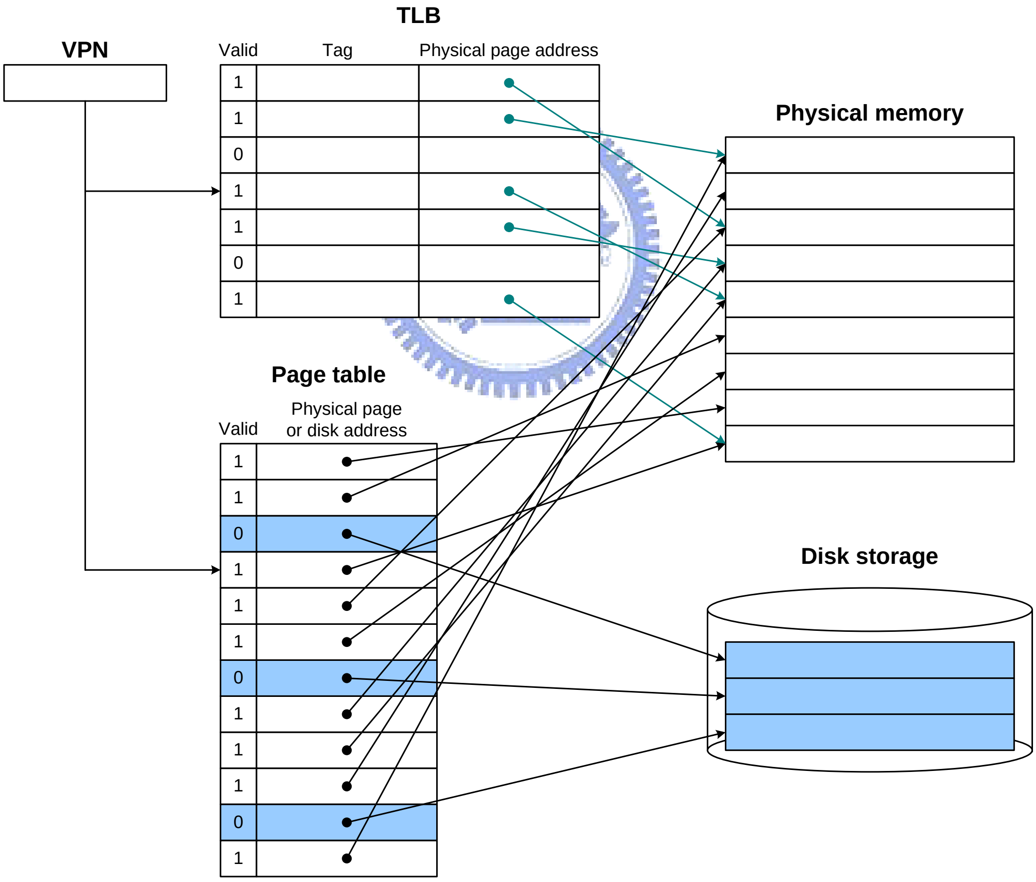


Fig. 2.9 A simple virtual memory system.

2.4.3 Packet Forwarding Using CAM

We describe the application of CAMs to packet forwarding in network routers. First, we briefly summarize packet forwarding and then show how a CAM implements the required operations. Further examples of approaches that use CAMs for the purposes of sorting and searching are provided in [39]-[40]. Network routers forward data packets from an incoming port to an outgoing port, using an address-lookup function. The address-lookup function examines the destination address of the packet and selects the output port associated with that address. The router maintains a list, called the routing table, that contains destination addresses and their corresponding output ports. An example of a simplified routing table is displayed in Table 2.5. All four entries in the table are 5-bit words, with the don' care bit, "X", matching both a 0 and a 1 in that position. Because of the "X" bits, the first three entries in the Table represent a range of input addresses, i.e., entry 1 maps all addresses in the range 10100 to 10111 to port A. The router searches this table for the destination address of each incoming packet, and selects the appropriate output port. For example, if the router receives a packet with the destination address 10100, the packet is forwarded to port A. In the case of the incoming address 01101, the address. lookup matches both entry 2 and entry 3 in the table. Entry 2 is selected since it has the fewest "X" bits, or, alternatively, it has the longest prefix, indicating that it is the most direct route to the destination. This lookup method is called longest-prefix matching. Fig. 2.10 illustrates how a CAM accomplishes address lookup by implementing the routing table shown in Table 2.5. On the left of Fig. 2.10, the packet destination-address of 01101 is the input to the CAM. As in the table, two locations match, with the (priority) encoder choosing the upper entry and generating the match location 01, which corresponds to the most-direct route. This match location is the input address to a RAM that contains a list of output ports, as depicted in Fig. 2.10. A RAM read operation outputs the port designation; port B, to which the incoming packet is forwarded. We can view the match location output of the CAM as a pointer that retrieves the associated word from the RAM. In the particular case of packet forwarding the associated word is the designation of the output port. This CAM/RAM system is a complete implementation of an address-lookup engine for packet forwarding.

Table 2.5 Example Routing Table

Entry NO.	Address (Binary)	Output Port
1	101XX	A
2	0110X	B
3	011XX	C
4	10011	D

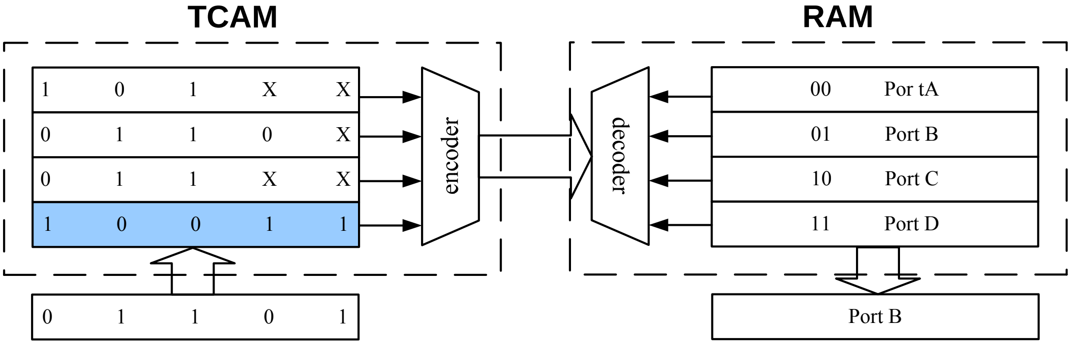
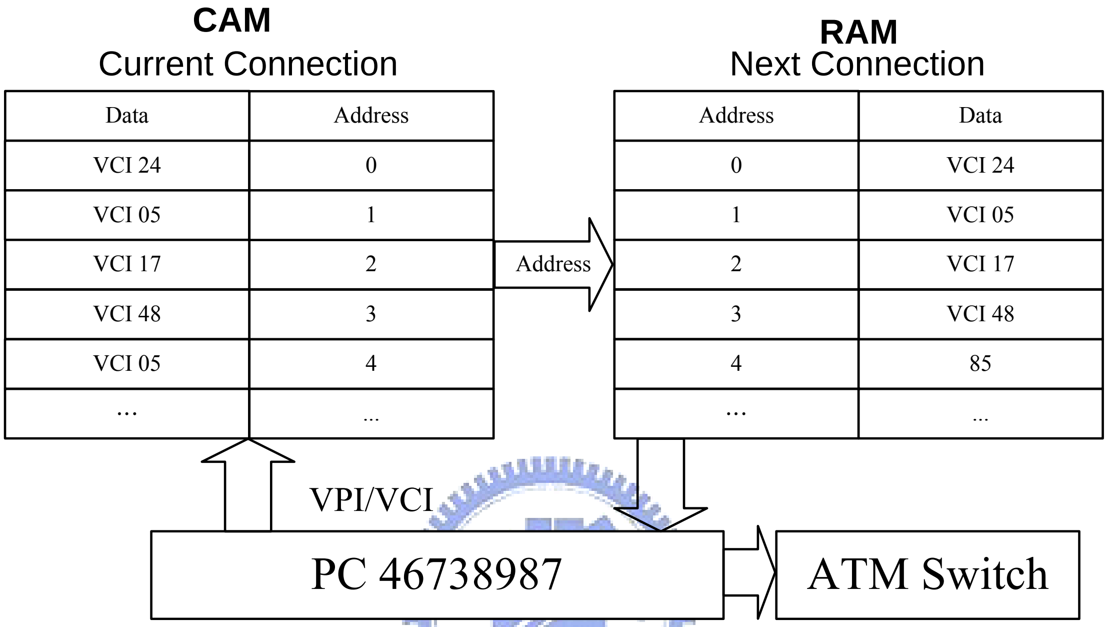


Fig. 2.10 CAM-based implementation of the routing table of Table 2.5

2.4.4 ATM Switches

For ATM switching network application, CAM can be adopted as a translation table. Virtual circuits are important parts to ATM networks, and they need to be set up across ATM networks before any data transfer because ATM networks are connection-oriented. There are two types of ATM virtual circuits, Virtual Path (identified by a virtual path identifier [VPI]) and Channel Path (identified by a channel path identifier [VCI]). Each segment of the total connection has unique VPI/VCI combinations, and the VPI/VCI value of ATM cells would be changed into the value for the next segment of connection while ATM cell go through a switch [41]-[42].

CAM is applied to an ATM switch as an address translator and can quickly perform the VPI/VCI translation. In the translation process, the CAM causes address which access data in RAM and uses incoming VPI/VCI values in ATM cell headers. A CAM/RAM combination realizes the multi-megabit translation tables with full parallel search capability. Take VPI/VCI fields from the TM cell header and the list of current connections stored in the CAM array for comparison, as a result, CAM originates and address which is used to access an external RAM where VPI/VCI mapping data and other connection information is stored. The ATM controller uses the VPI/VCI data from the RAM for modifying the cell header, and the cell is sent to the switch, depicted in Fig. 2.11.



CAM
Current Connection

Data	Address
VCI 24	0
VCI 05	1
VCI 17	2
VCI 48	3
VCI 05	4
…	…

RAM
Next Connection

Address	Data
0	VCI 24
1	VCI 05
2	VCI 17
3	VCI 48
4	85
…	…

Fig. 2.11 ATM switch with CAM.

2.5 Low power Content-addressable Memory Designs

In the past, there are several low-power and high-speed CAMs were proposed [43]-[46]. Many techniques were developed to improve the performance of CAM. On match-line scheme, there are low-swing scheme, current-race scheme, selective-precharge scheme, pipelining scheme and current-saving scheme. On search-line scheme, there are eliminating search-line precharge, charge recycling search-line driver and hierarchical search-line scheme. These designs will be introduced as follow.

2.5.1 Method of reducing Match-line power consumption

The dynamic power consumed by a single match-line that misses is due to the rising edge during precharge and the falling edge during evaluation, and is given by the equation

$$P_{miss} = C_{ML}V_{DD}^{2} f \tag{2.1}$$

where *f* is the frequency of search operations. In the case of a match, the power consumption associated with a single match-line depends on the previous state of the match-line; however, since typically there is only a small number of match we can neglect this power consumption. Accordingly, the overall match-line power consumption of a CAM block with ω match-lines is

$$P_{ML} = \omega P_{miss} = \omega C_{ML}V_{DD}^{2} f \tag{2.2}$$

2.5.1.1 Low-Swing Scheme

One method of reducing the ML power consumption, and potentially increasing its speed, is to reduce the ML voltage swing [25], [47]. The reduction of power consumption is linearly proportional to the reduction of the voltage swing, resulting in the modified power equation

$$P_{ML} = \omega \times C_{ML}V_{DD}V_{MLswing} f \tag{2.3}$$

The match-line swing is reduced from V_{DD} to $V_{MLswing}$ where $V_{MLswing} < V_{DD}$. We can obvious that P_{ML} reduced.

2.5.1.2 Current-Race Scheme

Fig. 2.12 (a) shows a simplified schematic of the current-race scheme [48]. This scheme precharges the match-line low and evaluates the match-line state by charging the match-line with a current IML supplied by a current source. The signal timing is shown in Fig. 2.12 (b). The precharge signal, mlpre, starts the search cycle by precharging the match-line low. Since the match-line is precharged low, the scheme concurrently charges the search-lines to their search data values, eliminating the need for a separate SL precharge phase required by the precharge-high scheme. Instead, there is a single SL/ML precharge phase, as indicated in Fig. 2.12 (b). After the SL/ML precharge phase completes, the enable signal, $\overline{en}$, connects the current source to the match-line. A match-line in the match state charges linearly to a high voltage, while a match-line in the miss state charges to to a voltage of only $I_{ML} \times R_{ML} / m$, where m denotes the number of misses in cells connected to the match-line. By setting the maximum voltage of a miss to be small, a simple match-line sense amplifier easily differentiates between a match state and a miss state and generates the signal MLso. As shown in Fig. 2.12 (a), the amplifier is the nMOS transistor, M_{sense}, whose output is stored by a half-latch. The nMOS sense transistor trips the latch with a threshold of V_{tn}. After some delay, match-lines in the match state will charge to slightly above V_{tn} tripping their latch, whereas match-lines in the miss state will remain at a much smaller voltage, leaving their latch in the initial state. A simple replica match-line (not shown) controls the shutoff of the current source and the latching of the match signal.

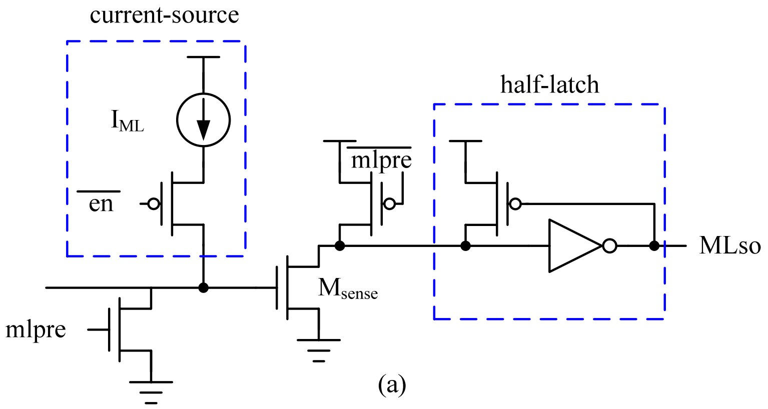
We derive the power consumption of this scheme by first noting that the same amount of current is discharged into every match-line, regardless of the state of the match-line. Looking at the match case for convenience, the power consumed to charge a match-line to slightly above V_{tn} is

$$P_{match} = C_{ML} V_{DD} V_{tn} f \tag{2.4}$$

Since the power consumption of a match and a miss are identical, the overall power consumption for all ω match-lines is

$$P_{ML} = \omega \times C_{ML} V_{DD} V_{tn} f \tag{2.5}$$

This equation is identical to the low-swing scheme (2.3), with $V_{MLswing} = V_{tn}$



(a)

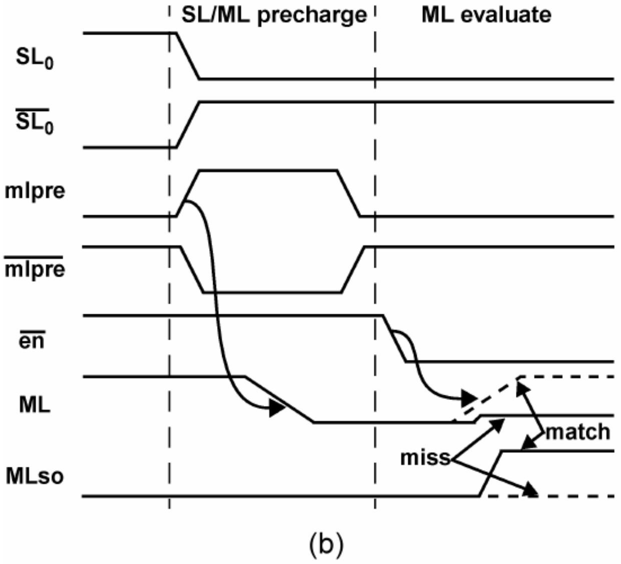


Fig. 2.12 For current-race match-line sensing [48]: (a) a schematic of the circuit implementation including precharge circuitry and (b) a timing diagram for a single search cycle

2.5.1.3 Selective-Precharge Scheme

Selective-precharge, performs a match operation on the first few bits of a word before activating the search of the remaining bits [49]. For example, in a 144-bit word, selective precharge initially searches only the first 3 bits and then searches the remaining 141 bits only for words that matched in the first 3 bits. Assuming a uniform random data distribution, the initial 3-bit search should allow only 1/2 words to survive to the second stage saving about 88% of the match-line power. In practice, there are two sources of overhead that limit the power saving. First, to maintain speed, the initial match implementation may draw a higher power per bit than the search operation on the remaining bits. Second, an application may have a data distribution that is not uniform, and, in the worst-case scenario, the initial match bits are identical among all words in the CAM, eliminating any power saving. Fig. 2.13 is an example of selective precharge in the original paper [49]. The example uses the first k bits for the initial search and the remaining m-k bits for the remaining search. The RML is precharged through the PMOS which is controlled by the NAND CAM cell and turned on only if there is a match in the first k CAM bits. The remaining cells are NOR cells. Thus, one implementation of selective precharge is to use this mixed NAND/NOR match-line structure. Selective precharge is perhaps the most common method used to save power on match-lines [24], [50]-[54] since it is both simple to implement and can reduce power by a large amount in many CAM applications.

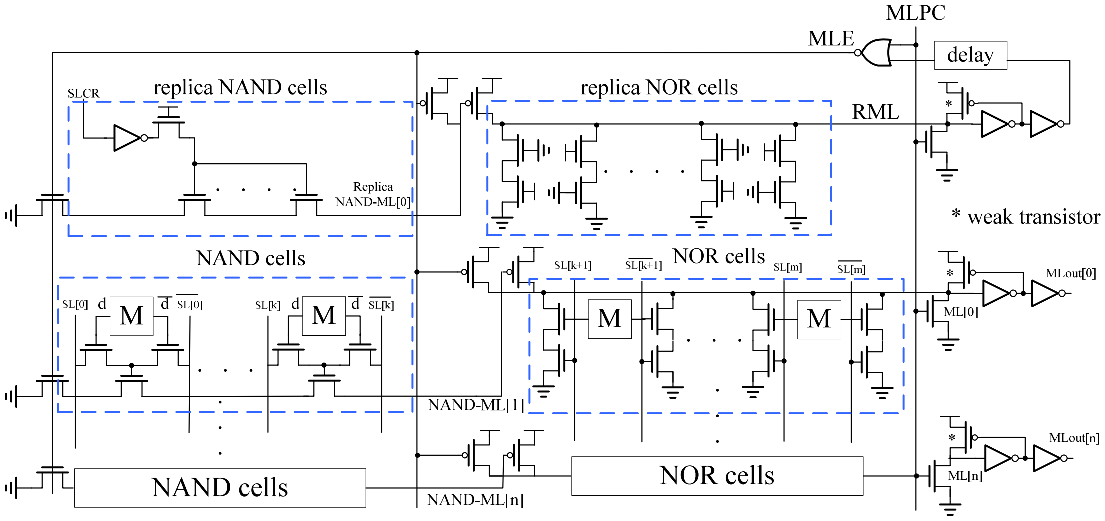


Fig. 2.13 Pulsed NAND-NOR ML architecture

2.5.1.4 Pipelining Scheme

In selective precharge, the match-line is divided into two segments. More generally, an implementation may divide the match-line into any number of segments, where a match in a given segment results in a search operation in the next segment but a miss terminates the match operation for that word. A design that uses multiple match-line segments in a pipelined fashion is the pipelined match-lines scheme [55],[56]. Fig. 2.14 (a) shows a simplified schematic of a conventional NOR-type match-line structure where all cells are connected in parallel. Fig. 2.14 (b) shows the same set of cells as in Fig. 2.14 (a), but with the match-line broken into four match-line segments that are serially evaluated. If any stage misses, the subsequent stages are shut off, resulting in power saving. The drawbacks of this scheme are the increased latency and the area overhead due to the pipeline stages. By itself, a pipelined match-line scheme is not as compelling as basic selective precharge; however, pipelining enables the use of hierarchical search-lines, thus saving power. Section IV discusses hierarchical search-lines in detail. Another approach is to segment the match-line so that each individual bit forms a segment [57]. Thus, selective precharge operates on a bit-by-bit basis. In this design, the CAM cell is modified so that the match evaluation ripples through each CAM cell. If at any cell there is a miss, the subsequent cells do not activate, as there is no need for a comparison operation. The drawback of this scheme is the extra circuitry required at each cell to gate the comparison with the result from the previous cell.

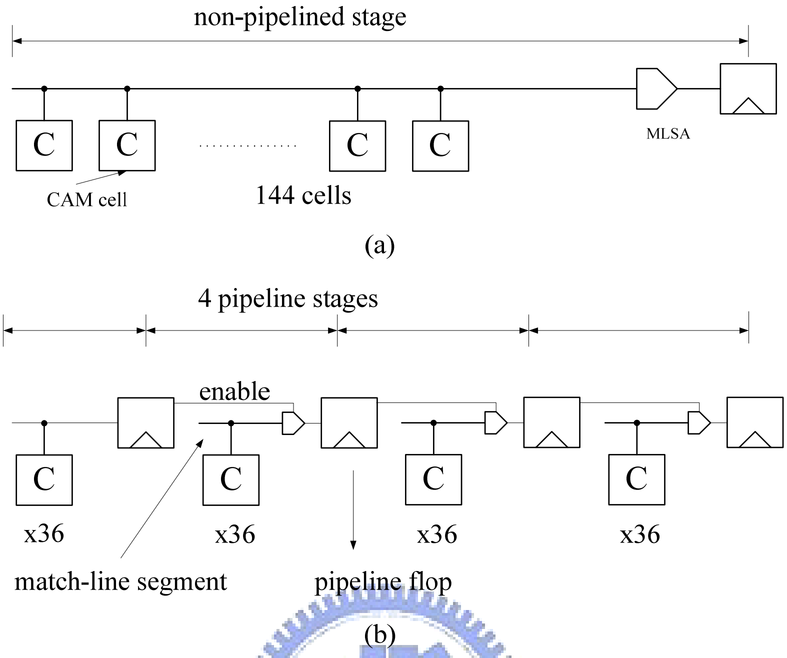


Fig. 2.14 Pipelined match-lines reduce power by shutting down after a miss in a stage

2.5.1.5 Current-Saving Scheme

The current-saving scheme [58], [59] is another data-dependent match-line-sensing scheme which is a modified form of the current-race sensing scheme. The key improvement of the current-saving scheme is to allocate a different amount of current for a match than for a miss. In the current-saving scheme, matches are allocated a larger current and misses are allocated a lower current. Since almost every match-line has a miss, overall the scheme saves power. Fig. 2.15 shows a simplified schematic of the current-saving scheme. The main difference from the current-race scheme as depicted in Fig. 2.12 (a) is the addition of the current-control block. This block is the mechanism by which a different amount of current is allocated, based on a match or a miss. The input to this current-control block is the match-line voltage, V_{ML}, and the output is a control voltage that determines the current, I_{ML}, which charges the match-line. The current-control block provides positive feedback since higher V_{ML} results in higher I_{ML}, which, in turn, results in higher V_{ML}. In this scheme, the match-line is precharged low, just as in the current-race scheme. In the current-race scheme, the current source provides a constant amount of current regardless of the state of the match-line. In the current-saving scheme, the amount of current is initially the same for all match-lines, but the current control reduces the current provided to match-lines that miss (have a resistance to ground), but maintains the current to match-lines that match (with no resistance to ground). The current-control block increases the current as the voltage on the ML_{rises} and the voltage on the ML rises faster for large ML resistance. Since the amount of current decreases with the number of misses, it follows that the power dissipated on the match-line also depends on the number of bits that miss. References [58] and [59] show that this scheme saves over 50% of match-line power compared to the current-race scheme.

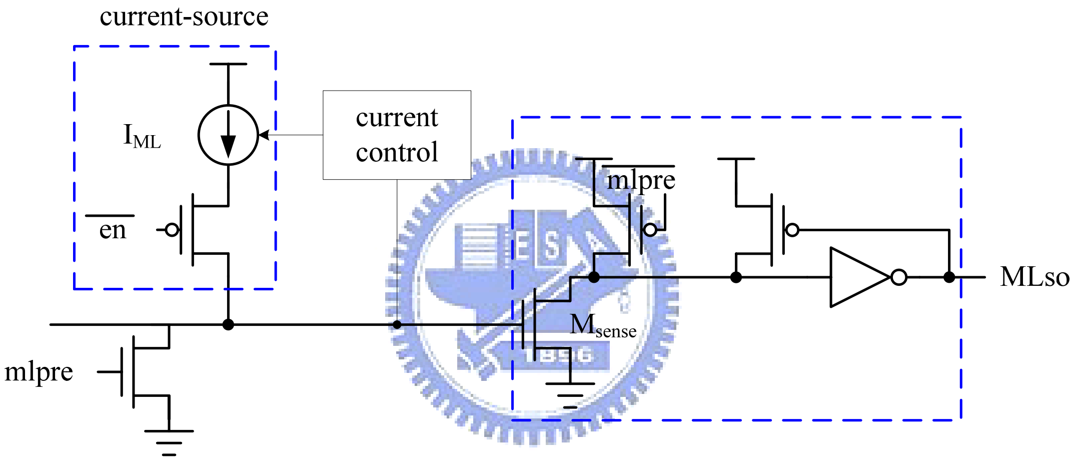


Fig. 2.15 Current-saving matchline-sensing scheme [58], [59].

2.5.2 Method of reducing Search-line power consumption

2.5.2.1 Hierarchical Search-line Scheme

Hierarchical search-lines are built on top of pipelined match-lines [26], [55], [56], [60]. The basic idea of hierarchical search-lines is to exploit the fact that few match-lines survive the first segment of the pipelined match-lines.

With the conventional search-line approach, even though only a small number of match-lines survive the first segment, all search-lines are still driven. Instead of this, the hierarchical search-line scheme divides the search-lines into a two-level hierarchy of global search-lines (GSLs) and local search-lines (LSLs). Fig. 2.16 shows a simplified hierarchical search-line scheme, where the match-lines are pipelined into two segments, and the search-lines are divided into four LSLs per GSL. In the figure, each LSL feeds only a single match-line (for simplicity), but the number of match-lines per LSL can be 64 to 256. The GSLs are active every cycle, but the LSLs are active only when necessary. Activating LSLs is necessary when at least one of the match-lines fed by the LSL is active. In many cases, an LSL will have no active match-lines in a given cycle, hence there is no need to activate the LSL, saving power. Thus, the overall power consumption on the search-lines is

$$P_{SL} = \left(C_{GSL}V_{DD}^{2} + \alpha C_{LSL}V_{DD}^{2}\right)f \tag{2.6}$$

Where C_{GSL} is the GSL capacitance, C_{LSL} is the LSL capacitance (of all LSLs connected to a GSL) and α is the activity rate of the LSLs. C_{GSL} primarily consists of wiring capacitance, whereas C_{LSL} consists of wiring capacitance and the gate capacitance of the SL inputs of the CAM cells. The factor α, which can be as low as 25% in some cases, is determined by the search data and the data stored in the CAM. We see from (2.6) that determines how much power is saved on the LSLs, but the cost of this savings is the power dissipated by the GSLs. Thus, the power dissipated by the GSLs must be sufficiently small so that overall search-line power is lower than that using the conventional approach.

If wiring capacitance is small compared to the parasitic transistor capacitance [60], then the scheme saves power. However, as transistor dimensions scale down, it is expected that wiring capacitance will increase relative to transistor parasitic capacitance. In the situation where wiring capacitance is comparable or larger than the parasitic transistor capacitance, C_{GSL} and C_{LSL} will be similar in size, resulting in no power savings. In this case, small-swing signaling on the GSLs can reduce the power of the GSLs compared to that of the full-swing LSLs [55], [56]. This results in the modified search-line power of

$$P_{SL} = 2n\left(C_{GSL}V_{LOW}^{2} + \alpha C_{LSL}V_{DD}^{2}\right)f \tag{2.7}$$

where V_{LOW} is the low-swing voltage on the GSLs (assuming an externally available power supply V_{LOW}). This scheme requires an amplifier to convert the low-swing GSL signal to the full-swing signals on the LSLs. Fortunately, there is only a small number of these amplifiers per search-line, so that the area and power overhead of this extra circuitry is small.

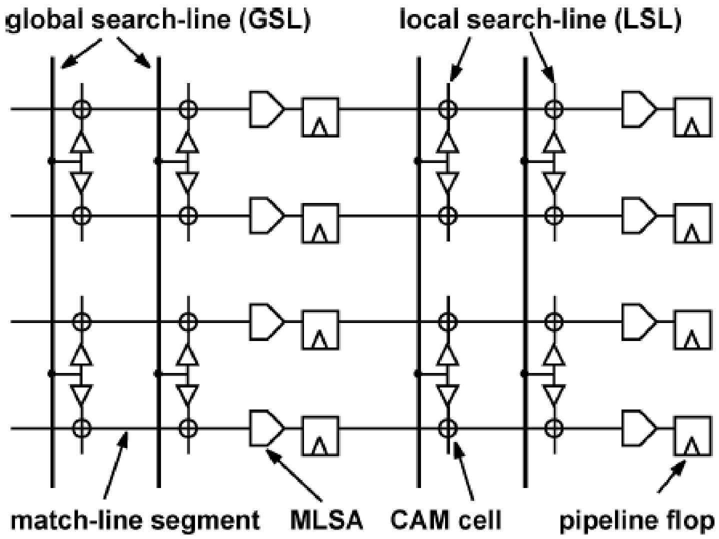


Fig. 2.16 Schematic of the hierarchical search-line architecture.

2.5.2.2 Charge-Recycling Search-line Driver

The SLs consume power only when the search data change. In general, the transition probability of SL is smaller than one half. The nonprecharged SLs consume less power than half of that of the precharged SLs [61]. The CRSLD further saves the SL power by recycling the charge of SLs. Fig. 2.17 shows the CRSLD architecture.

Table 2.6 shows the power comparison of SLs.

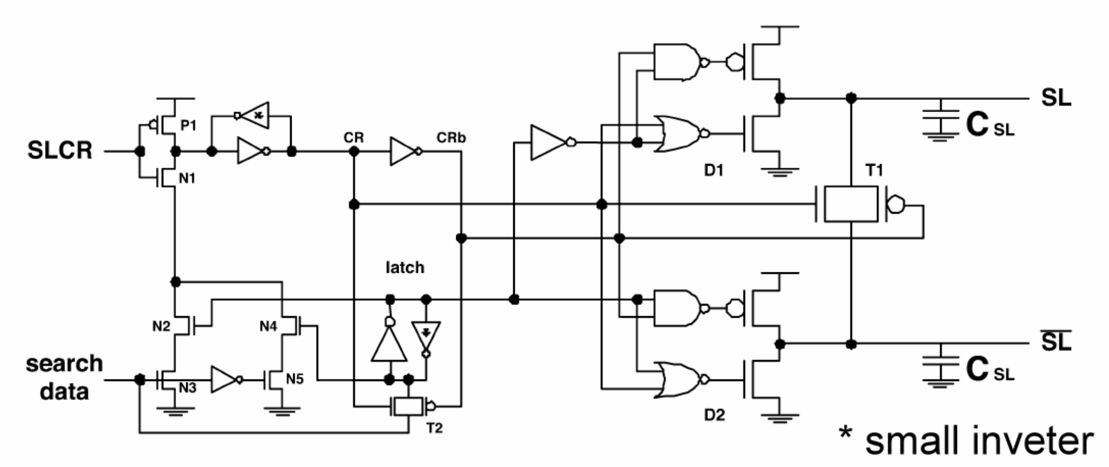


Fig. 2.17 Charge-recycling search-line driver

Table 2.6 Power consumption of CRSLD compare to other SL schemes

	Precharge	Charge recycling	Power of SL
Precharge SL	Yes	No	$f \times C_{SL} \times V_{DD}^2$
Non-precharge SL	No	No	$\alpha \times f \times C_{SL} \times V_{DD}^2$
CRSLD	No	Yes	$\frac{1}{2} \alpha \times f \times C_{SL} \times V_{DD}^2$

2.5.3 Power-Saving CAM architecture

A 32-bit IPv4 destination IP address could be divided into two parts. The first 8-bit is called merged field (MF) and remaining 24-bit is called individual field (IF). Therefore, the hybrid-type TCAM architecture has three different types of banks: the main bank, constructed with high-speed NOR-type TCAM, the sub bank, and the extra bank constructed with low-power AND-type TCAM [62], as shown in Fig. 2.18. In this case, most of the data are divided into common data portion which store in MF and its unique data portion which store IF. The main bank stores k-bit MF, and the number of its word lines is equal to the number of sub bank. The match result of each word line of main bank will activate the corresponding sub bank as illustrated in Fig. 2.19. If some data don’t have any common MF, the extra bank stores them without dividing them into MF and IF.

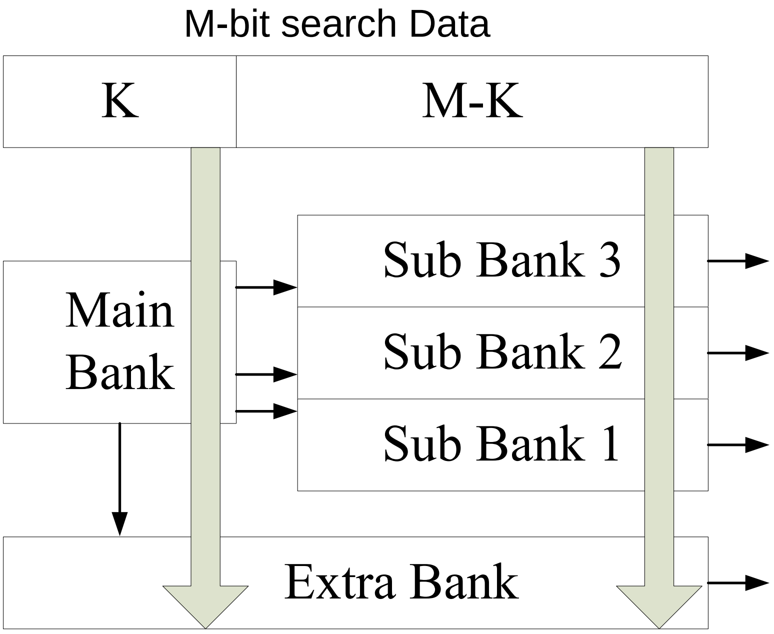


Fig. 2.18 Block diagram of hybrid-type TCAM architecture.

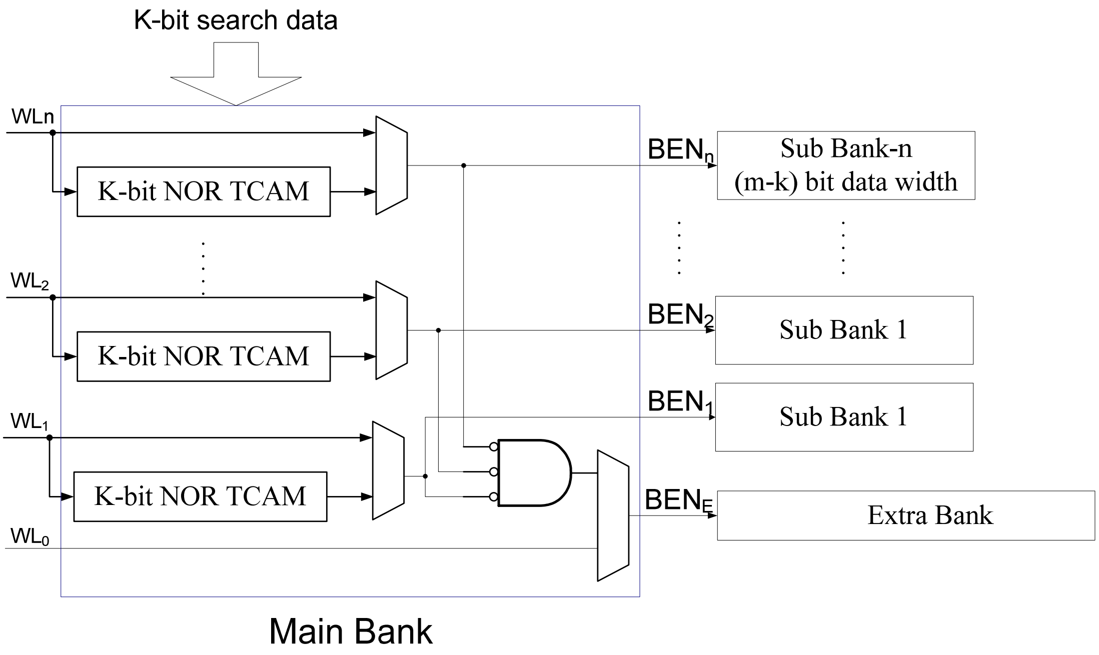


Fig. 2.19 Block diagram of main bank.

Applying this hybrid-type of TCAM architecture, area saving can be expected. Assuming that a conventional CAM of $(B \times m \times W)$ where B is the number of banks, m is the bit width of a bank, and W is the number of words in a bank. However, if applying the hybrid architecture, the total number of the cells is given by:

$$m \times W + (B-1) \times [(m-k) \times W] + (B-1) \times k \tag{2.8}$$

Therefore, the area is approximately $(m-k)/m$ of the conventional architecture.

In order to reduce the search time and redundant search power consumption, the match-line repeater (MLRPT), 1-bit column decoding, and local priority encoding (LPE) schemes are applied. As described in Fig. 2.20, a word block consists of four cell blocks, two partial match blocks (PM), and a main match block. Each cell block includes 72 AND-type TCAM cells and 3 MLRPT to construct a search engine with 144-bit search data width. 288 cells are attached to each word line and they are column-decodes using LSB address when read or write operation is performed.

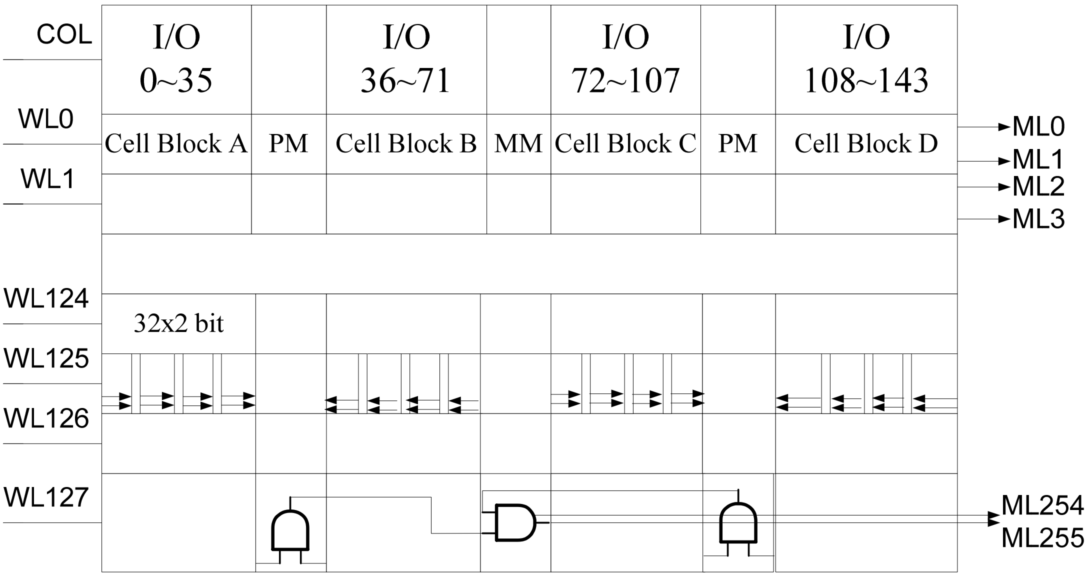


Fig. 2.20 Sub/extra bank structure.

Because the major drawback of AND-type TCAM cell is the slow search time, the match-line repeater is proposed to improve pre-charge and evaluation time. As shown in Fig. 2.21, one MLRPT is composed of pass transistors to speed up the propagation of the match-line.

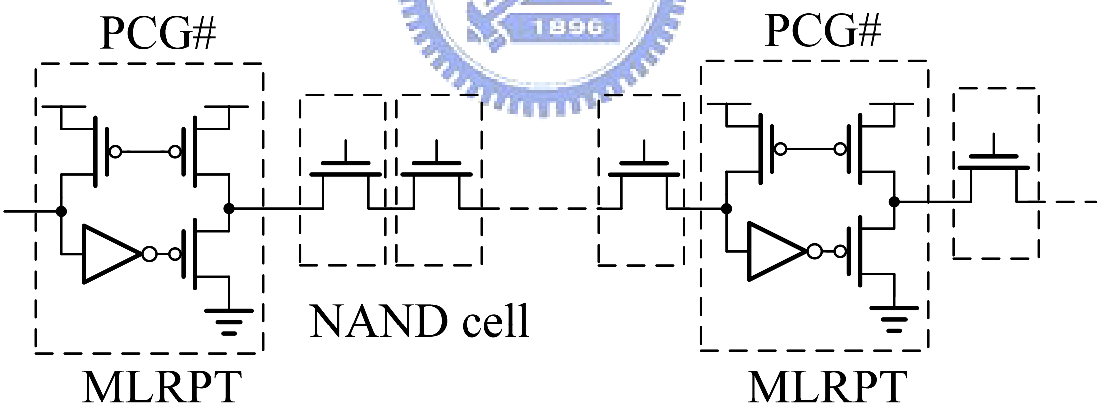


Fig. 2.21 Schematic of match-line repeater.

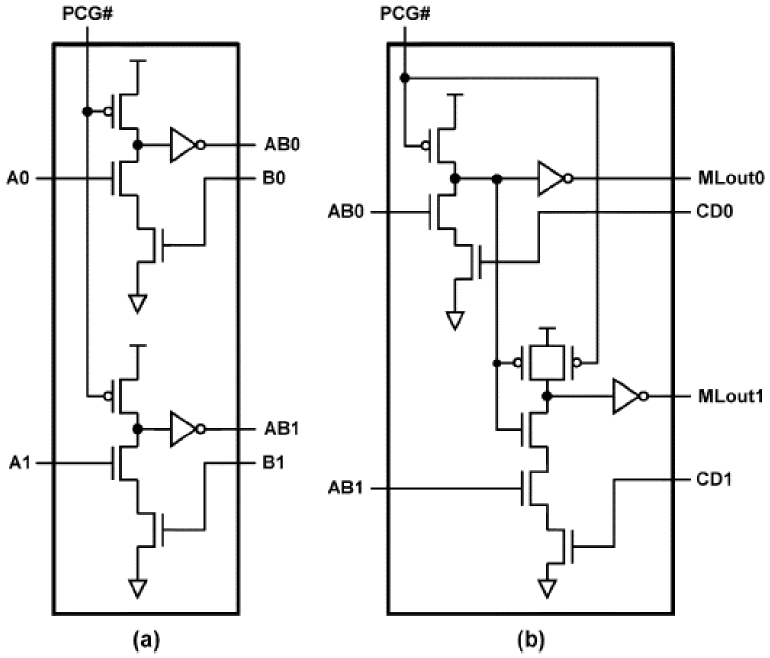


Fig. 2.22 (a) Partial match (PM) block. (b) Main match (MM) block and local priority encoding (LPE).

The Fig. 2.22 shows the schematics of partial match block and main match block which also integrates with LPE circuits. AB0, AB1, CD0, and CD1 are evaluated by PMs. (AB0, CD0) and (AB1, CD1) represent the partial match results of column0 and column1, respectively. They are merged to generate the final match results MLout0 and MLout1 by MM. When both column0 and column1 are matched, only the match-line of column0 is valid because of LPE.

Chapter 3
Noise Tolerant Butterfly Mach-line Scheme

3.1 On-Chip Noise Sources

Fig. 3.1 depicts an N to 1 multiplexer circuit with the presence of vital noise sources. Generally, noise sources include gate internal noises, external noises, process variation, alpha particle radiation, and so on [63], [64]. This section focus on cross-couple noises, charge sharing noise, leakage noises, power bouncing noises, ground bouncing noises, and substrate noises. The detail descriptions are introduced as follows.

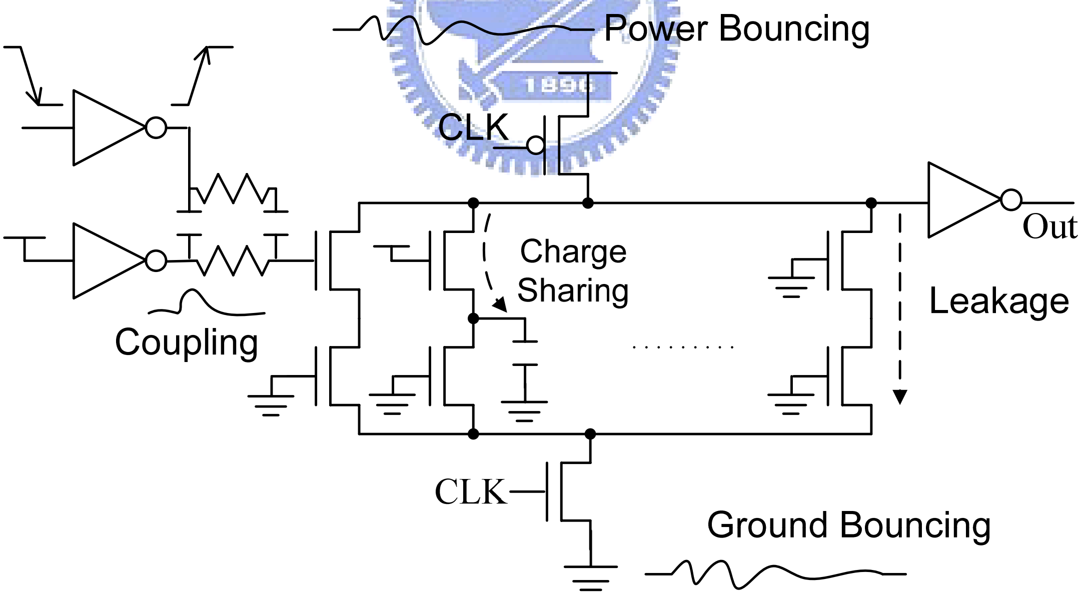


Fig. 3.1 High fan-in dynamic multiplexer circuit and the presence of noise sources.

3.1.1 Cross-Couple Noises

The coupling noises occur because of signal transitions being coupled capacitive to a net from its neighbors. This form of noise, which is also referred to as cross-talk noise, is recognized as a dominant source of noise in scaled VLSI technologies. Fig. 3.2 shows the topology of the circuit under the influence of coupled noise.

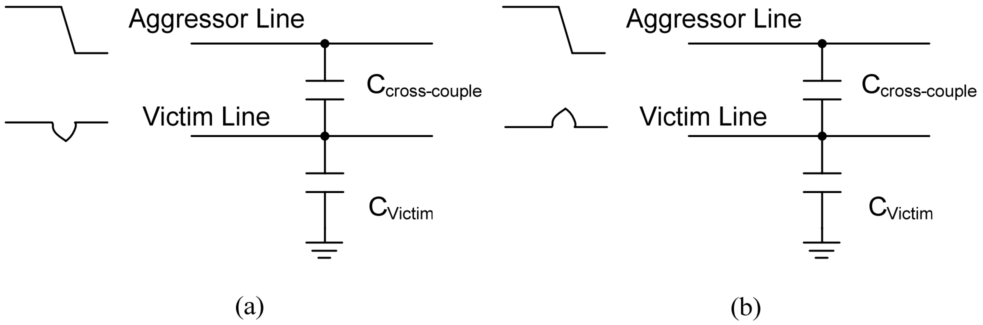


Fig. 3.2 Diagram of cross-couple noises. (a) Negative voltage excursion. (b) Positive voltage excursion.

Cross-couple noises appearing on input terminals and floating nodes will cause the dynamic circuits to malfunction. Because of the nature of dynamic CMOS, a noise pulse does not always produce a logical error at the circuit output. The response of the output depends on two factors: 1) The amplitude and width of the noise pulse; 2) The state of the circuit as well as the clock. Assume a noise pulse with sufficient amplitude and width occurs at one of the circuit inputs or at the clock input. The possible combinations of the inputs push the output to respond erroneously.

Input noises refer to noises presented at the inputs of a logic gate. They are primarily caused by the coupling effect, also known as crosstalk, among adjacent signal wires. This type of noise has become a prominent source of failures for nano-scale VLSI circuits because of the aggressive interconnect scaling in the lateral dimensions with relatively unchanged vertical dimensions.

3.1.2 Charge Sharing Noises

Charging sharing noise is caused by charge redistribution between the dynamic node and the internal nodes of the pull-down network. Charge sharing reduces the voltage level at the dynamic node causing potential false switching of a dynamic logic gate.

3.1.3 Leakage Noises

Leakage noises refer to the possible charge loss in the evaluation phase due to leakage currents. Sub-threshold leakage and gate tunneling leakage currents are getting even worse in nano-scale technologies. Leakage currents increase exponentially in relation to transistor threshold voltage, which is continuously being down-scaled while the power-supply voltage and threshold voltage reduce. Therefore, leakages in transistors can be significant sources of noise in wide fan-in dynamic logic gates designed under nano-scale process technology.

3.1.4 Power Rail Bouncing Noises

Power and ground noises are mainly caused by the parasitic resistance and inductance, which are at the power and ground networks and at the chip package. Power and ground networks can also be contaminated by external noises from chip pins. Besides obviously reducing gate noise margin due to possibly lowered supply voltage, the power and ground voltage mismatch between a driver gate and a receiver gate can be translated to a DC noise at the input of the receiver.

3.1.5 Substrate Noises

Substrate noise can affect the signal integrity of a logic gate through substrate coupling. Furthermore, since transistor threshold voltage is a function of the substrate voltage, noise in the substrate can momentarily lower the threshold voltage of the transistors in the pull-down network rendering them more susceptible to other noises.

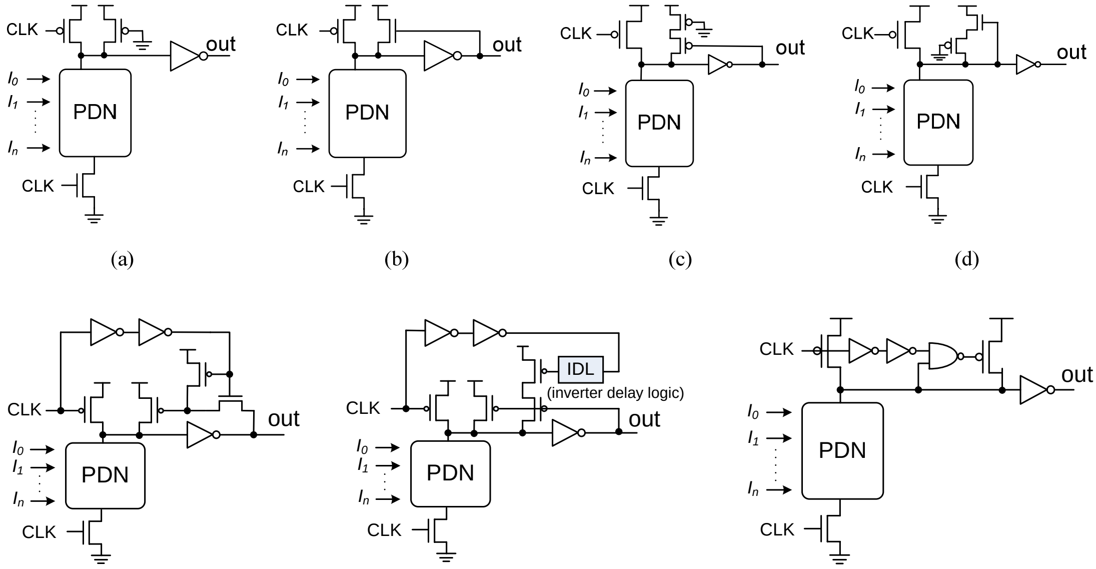
3.2 Noise-Tolerant Circuits Design

The noise immunity is a notable issue for SOC design. Several techniques have been developed to improve the noise immunity in dynamic circuits. According to the functionality and operation, they could be classified into four distinct types, employing keepers, pre-charge internal nodes, raising source voltage, and constructing complementary p-network. The detail descriptions are introduced as follows.

3.2.1 Employing Keepers

Dynamic circuits are sensitive to noise in consequence of the easily interfered floating nodes. Conventionally, upsizing the keepers is the simple method to enhance the noise tolerance of dynamic circuits [66], [67]. As floating nodes is going to be discharged caused by cross-couple, charge sharing, leakage current, and so on, keepers supply a small amount of current from the power-supply network to the floating node sp that the charge stored in the floating node is maintained [68].

The existing keeper techniques are schematized as follow. To begin with weak always on keeper shown in Fig. 3.3 (a), we note that gate of the keeper PMOS is tied to the ground in the original dynamic logic. Conventional feedback keepers or half latch became more widely used because they eliminate the potential DC power consumption problem using the always-on keeper in the evaluation phase of domino dynamic logic, as Fig. 3.3 (b) illustrates. Furthermore, weak feedback keepers and smart keepers [65] are shown in Fig. 3.3 (c) and Fig. 3.3 (d), each. Both of them have feeble ability of noise immunity but can enhance delay time for the same size of keeper. Finally, high-speed domino (HS-domino) [68], [69], skew-tolerant high-speed domino (STHS-domino) [70], and conditional keepers [71]-[73] are illustrated in Fig. 3.3 (e), Fig. 3.3 (f), and Fig. 3.3 (g), respectively. All the last three circuit techniques share the same principle, that is, to temporarily disable the keeper during the small time window when the dynamic gate switches. Although these circuit techniques have been developed for noise immunity improvement in high fan-in dynamic circuits, they also cause the propagation delay and dynamic power consumption penalties.



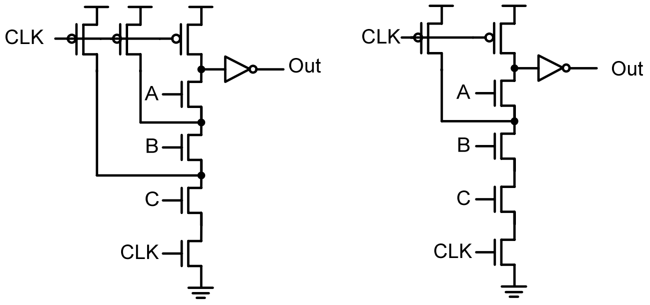
(a) (b) (c) (d)

(e) (f) (g)

Fig. 3.3 Applying keeper techniques to improve the noise immunity. (a) Weak always on keeper. (b) Conventional feedback keeper. (c) Weak keeper. (d) Smart keeper. (e) HS feedback keeper. (f) STHS feedback keeper. (g) Conditional keeper.

3.2.2 Pre-charge Internal Nodes

Charge sharing between floating node and internal nodes often results in false gate switching, especially for deep fain-in dynamic circuits. Pre-charging the internal nodes is an effective way to prevent the charge sharing problems. Pre-charge all internal nodes [74], as depicted in Fig. 3.4 (a), and partial pre-charge [75], as shown in Fig. 3.4 (b), have been developed to maintain the robustness of the dynamic circuits. It should be noted that pre-charge internal node techniques are useful for resisting the internal noises but are not very effective against external noises.



(a) (b)

Fig. 3.4 Applying pre-charge internal nodes techniques to improve the noise immunity. (a) Pre-chare all internal nodes. (b) Partial pre-charge.

3.2.3 Raising Source Voltage

Techniques regarding raising source voltage include PMOS pull-up [76], NMOS pull-up [77], mirror technique [78], and twin transistor technique [79], [80]. The principles of these four techniques are raising the voltages on source nodes to enhance noise immunity. The detailed configurations of the foregoing techniques are shown in Fig. 3.5. However, these techniques might conduct a direct DC path between V_{DD} and ground as shown in Fig. 3.6. Thus these techniques must be used with care in constructing large complex dynamic gates.

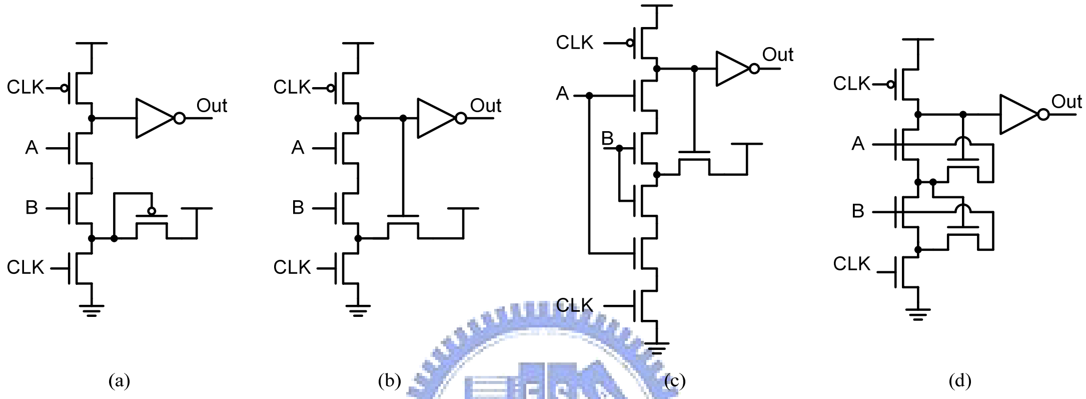


Fig. 3.5 Applying rising source voltage techniques to improve the noise immunity. (a) PMOS pull-up. (b) NMOS pull-up. (c) Mirror techniques. (d) Twin transistors techniques.

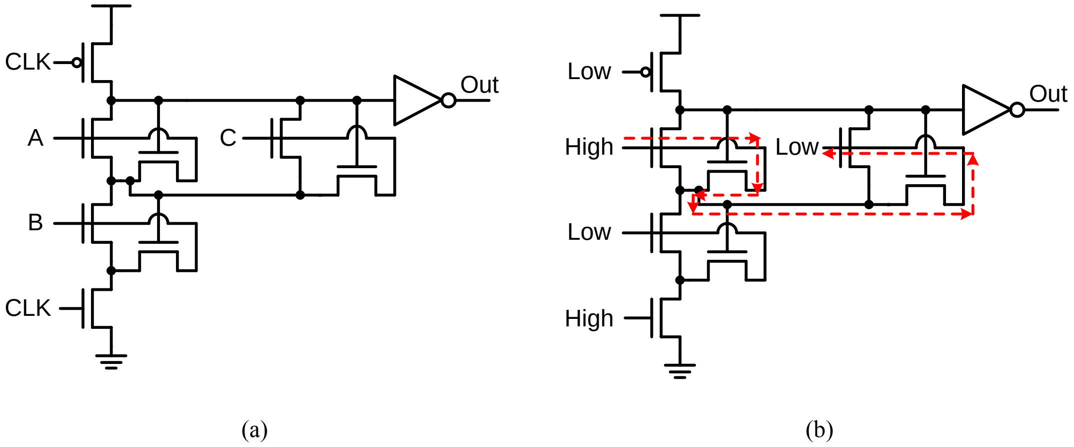


Fig. 3.6 Raising source voltage techniques with direct DC path. (a) A 3-input OR-AND gate with twin transistors. (b) Direct conducting path.

3.2.4 Creating Complementary P-Network

The basic principle of constructing complementary PMOS network is to build a weak complementary PMOS network to prevent the dynamic node from floating in the evaluation phase. This class of techniques includes complementary p-network technique [81], CMOS inverter technique [82], gated CMOS inverter technique [83], and triple transistor technique [84]. Fig. 3.7 depicts the detailed structures of the aforementioned techniques. However, these techniques might also bring out a direct DC path between VDD and ground as shown in Fig. 3.8. Likewise, these techniques must be used with caution in constructing large complicated dynamic gates.

Whereas the raising source voltage techniques and constructing complementary PMOS network techniques may exist direct conducting paths, these two methods require careful design to avoid the existence of direct conducting paths.

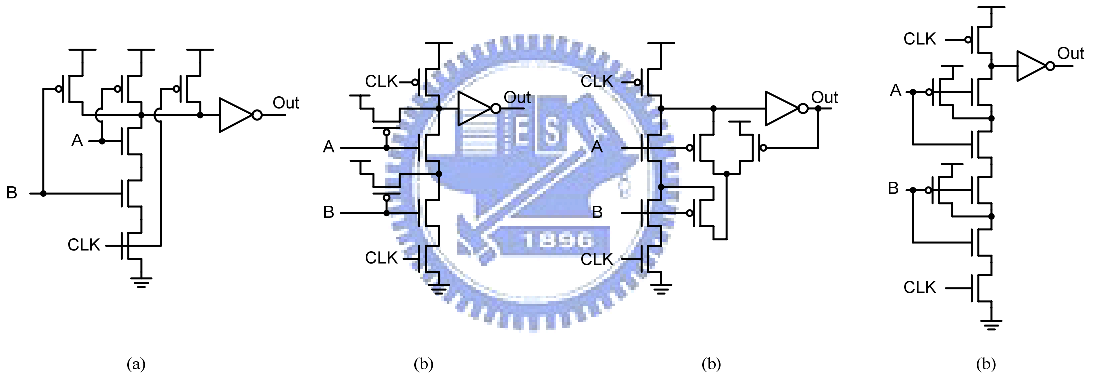


Fig. 3.7 Applying PMOS complementary techniques to increase noise immunity. (a) Complementary P-network techniques. (b) CMOS inverter technique. (c) Gated CMOS inverter technique. (d) Triple transistor technique.

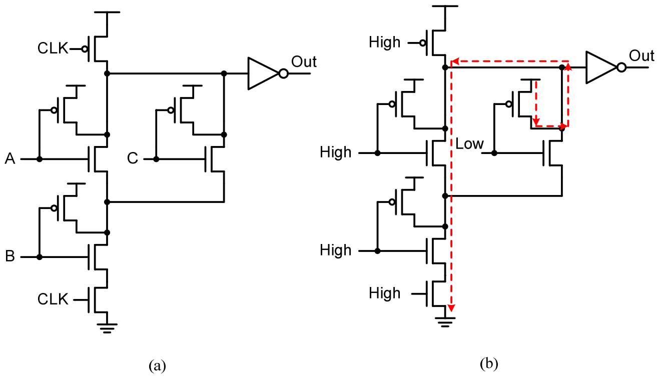


Fig. 3.8 CMOS inverter techniques with direct DC path. (a) A 3-input OR-AND gate. (b) Direct conducting path.

3.3 XOR-Based Conditional Keeper

Conventional keepers have worse performance in propagation delay and power consumption. Therefore, a novel keeper, XOR-based conditional keeper [85], would be presented in this section. The main idea of the proposed XOR-based conditional keeper is to avoid keeper turning on at the beginning of evaluation phase in dynamic circuit. The new control signals and their corresponding keeper status are shown in Table 3.1 and described as follows:

Table 3.1 Control organism of XOR-based conditional keeper.

CLK	**Floating Node**	**Control Signal on gate of keeper**
Low	Low	Low, to speed up the process of pre-charge
Low	High	High, to avoid the impact on performance at the very beginning of evaluation
High	Low	High, keeper should be off
High	High	Low, keeper should be activated to enhance the capability of noise immunity

When match-line pre-charge signal and the floating node are both low, the CAM circuit is at the beginning of the pre-charge period and we should turn on the conditional keeper to speed up the pre-charge procedure. As match-line pre-charge signal is low and the floating node is high, the pre-charge process is completed and the gate is ready for evaluation. Therefore, the conditional keeper should be turned off to avoid the impact on the delay and unnecessary power consumption.

While match-line pre-charge signal and floating node are both high, match-line is either at the beginning of the evaluation process or stores state HIGH on the floating node at the end of the evaluation process. If it is at the beginning of the evaluation process, the floating node will be eventually at the proper voltage level as long as the delay of the XOR gate is larger than the propagation delay of dynamic circuits. Since the delay time of dynamic circuits is shorter, the conditional keeper is slightly turned on at the beginning of the evaluation process and is fully turned on or off on the grounds of the final value stored on the floating node.

When match-line pre-charge signal is high and the floating node is low, the evaluation mode is finished and the final value stored on the floating node is low. Consequently, the conditional keeper should be fully turned off. According to the desired control signals, an XOR gate is what we need for generating these signals. Fig. 3.9 shows the XOR gates we used. Fig. 3.10 shows timing diagram for the XOR-based conditional keeper.

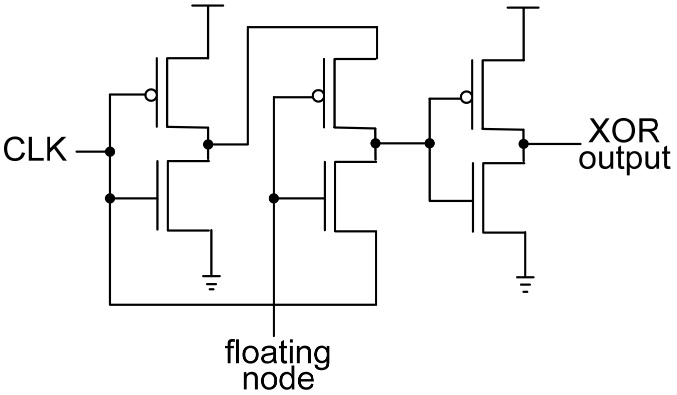


Fig. 3.9 The XOR gate implementation.

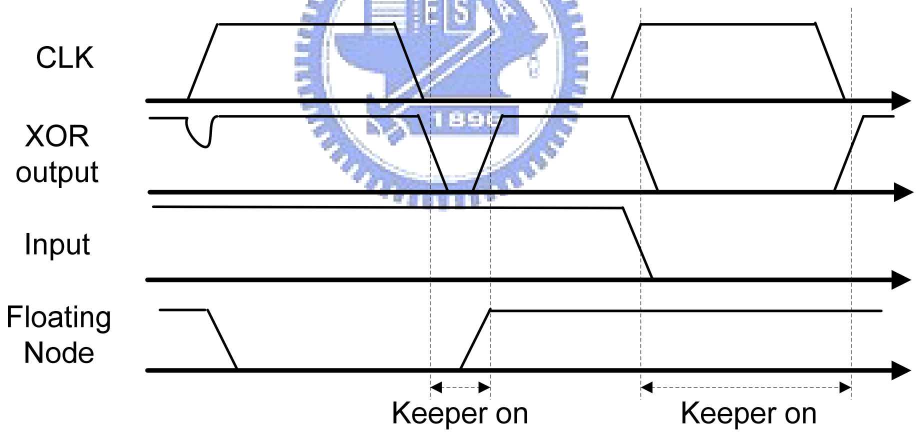
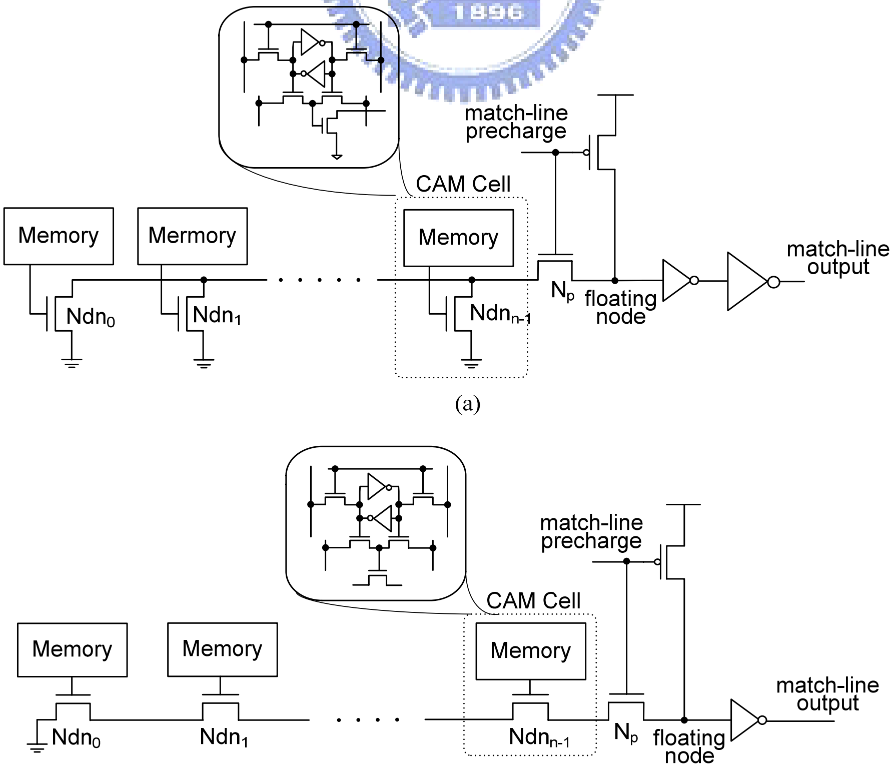


Fig. 3.10 Timing diagram for XOR-based conditional keeper.

The reason why we choose this XOR gate is that it requires fewer transistors and generates rail to rail output. There are still other 4 transistor XOR gates appeared in publications. However, most of those XOR gates are pass-transistor-based, thus the gates cannot have full swing output. This disadvantage degrades the noise immunity of conditional keepers.

3.4 Noise-Tolerant Match-line Schemes for CAM

Fig. 3.11 (a) and Fig. 3.11 (b) show NOR-type match-line scheme and AND-type match-line scheme respectively [56]. Use NOR-type match-line scheme for example, the floating node is charged to V_{DD}. N_p is turned off when the match-line precharge is low (pre-charge phase). As the match-line transfers to high (evaluation phase), whether the floating node is discharged to ground or keeps the floating voltage is in accordance with the search data matching stored data or not. If the search data are equal to the stored data, there is no path from floating node to ground and the floating node keeps it's previous voltage. Contrarily, if any bits of search data doesn't equal to the stored data, the floating node will be discharged to ground by NMOS, Nd_n. The AND-type match-line scheme is the same as NOR-type schemes in the pre-charge phase. However, the AND-type match-line scheme is different from NOR-type in evaluation phase. The floating node in AND-type scheme is discharged to ground when the search data and stored data are matched. No matter NOR-type match-line scheme or AND-type match-line scheme, they work like a high fan-in dynamic circuits [86]-[88]. For all high fan-in dynamic circuits are especially susceptible to all kinds of noises, they must guarantee the correctness of the function.



(a)

(b)

Fig. 3.11 Match-line schemes for CAM. (a) NOR-type. (b) AND-type.

3.4.1 NOR-type Noise-Tolerant Match-line Scheme

The NOR-type match-line scheme with XOR-based conditional keepers is shown in Fig. 3.12. At the beginning of pre-charge phase, the keeper enhances to charge the floating node. In the evaluation phase, if search data are equal to stored data, the floating node should be hold in high logic and the keeper will be woken up to defeat noises. However, if search data and stored data are mismatch, the floating node will be discharged through comparison circuits and the keeper will be turned off simultaneously. Furthermore, match-line scheme with XOR-gate keepers enhance the search time and reduce the power consumption.

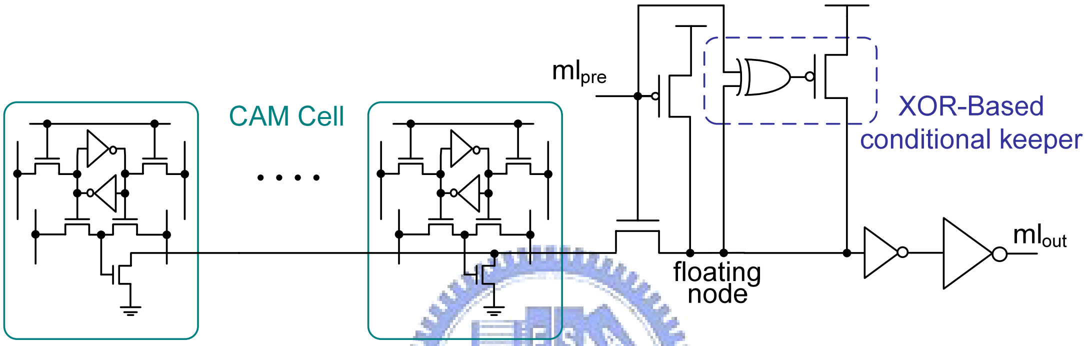


Fig. 3.12 NOR-type match-line scheme with XOR-based keeper.

3.4.2 AND-type Noise-Tolerant Match-line Scheme

As mentioned above, applying XOR-based conditional keeper in NOR-type match-line schemes has the advantage of reducing both search time and power consumption. The same is true of AND-type match-line schemes. The XOR-based conditional keeper is also adequate for AND-type ones, depicted in Fig. 3.13.

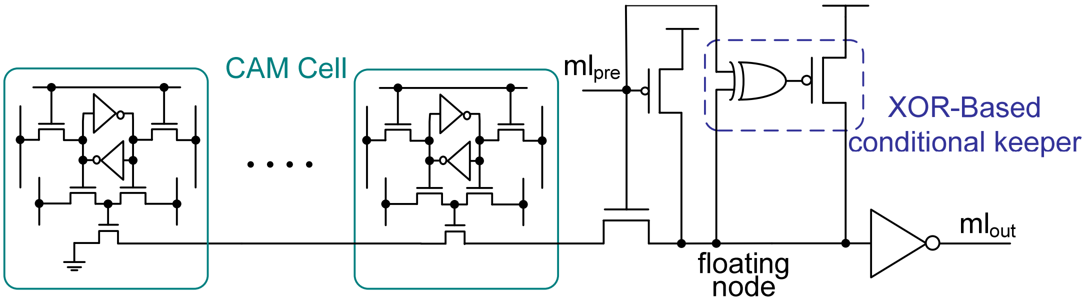


Fig. 3.13 AND-type match-line scheme with XOR-based keeper.

3.5 Noise-Tolerant Butterfly Match-line Scheme Design

As mentioned in section 2.3.2, AND-type match-line would be discharged to ground only when all bits of stored data are matched with all bits of search data. AND-type match-line scheme consumes less search power but needs longer search time to complete each search operation. In this section, a novel butterfly match-line scheme which is not only decrease search delay time but also power consumption is proposed. Combined with XOR-based conditional keeper, furthermore, a noise-tolerant butterfly match-line scheme is proposed.

3.5.1 Pseudo-Footless Clock-and-Data Pre-charge Dynamic (PF-CDPD) Circuits

AND-type match-line scheme connect CAM cells in serial causes long search time. Thus, in order to decrease the search time, it's necessary to separate the match-line schemes into several segments. Fig. 3.14 shows the four-input AND gate logic transform from dynamic circuits to clock-data pre-charge dynamic (CDPD) circuits [18]. Furthermore, a pseudo-footless clock-and-data pre-charge dynamic (PF-CDPD) circuit [85] which is combined with the AND-type match-line scheme and CDPD circuits are illustrated in Fig. 3.15.

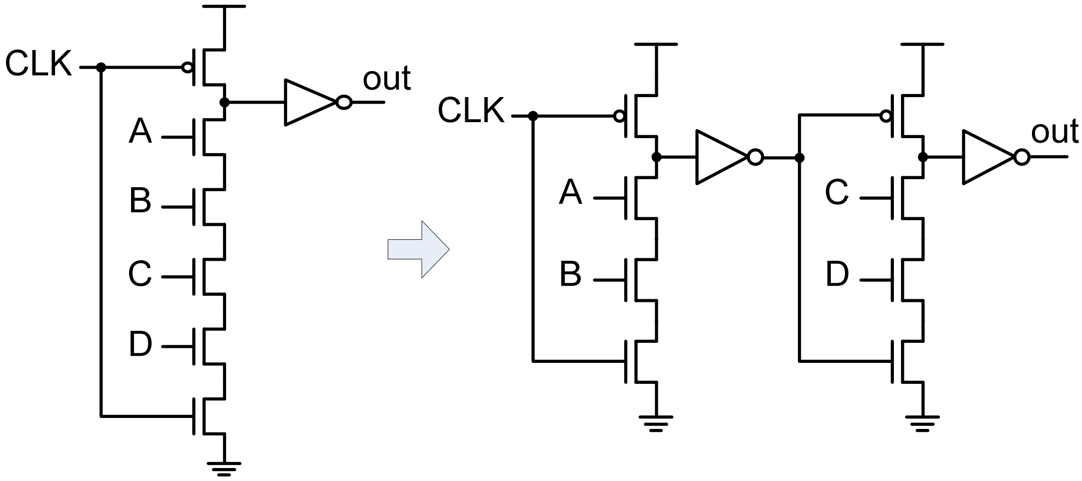


Fig. 3.14 Transfer four-input AND gate logic into clock-and-data pre-charge dynamic (CDPD) circuits.

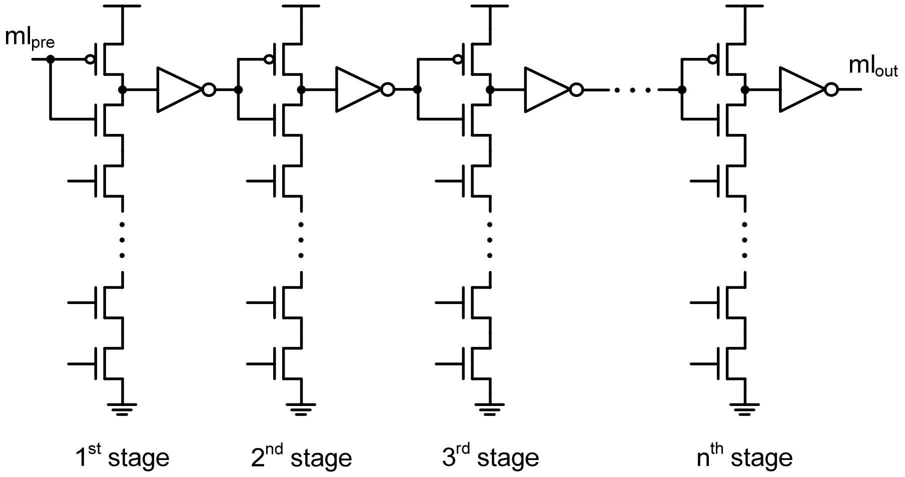


Fig. 3.15 Pseudo-footless clock-and-data pre-charge dynamic (PF-CDPD) circuits.

At the beginning of pre-charge phase, the floating node of first stage is charged to high and result of the first stage output becoming low to charge the floating node of next stage. Therefore, all floating nodes are charged to high during the pre-charge phase. At the evaluation phase, the comparison result of previous stage will influence the next stage. For example, if the comparisons result of first stage is match and thereby the first output becomes low to enable second stage comparison operation. In other words, if the comparison result of first stage is mismatch, the output of first stage still keeps low to disable the next stage comparison operation. The slowest evaluation happens when the input data matches with the stored data, and the PF-CDPD behaves much like a series of inverters. Accordingly, there are several advantages of CAM which adopts PF-CDPD circuits. First, the match-lines divided into many segments causes the size of comparison transistor being unnecessary too large in the same search time criteria. Second, because the size of comparison transistors becomes smaller, the switching capacitances are reduced effectively. Besides, separated match-line also causes the lower switching match-lines capacitances. Third, the evaluation operation enables or disables depending on output of previous stage. That is to say, if the stored data and search data are mismatch, the output will disable all comparison operations in after stages to avoid unnecessary switching. In consequence, PF-CDPD circuits not only enhance search time but also save power consumption.

3.5.2 Conventional Pipelined AND-type Match-line Scheme

Fig. 3.16 shows two type of pipelined AND-type match-line schemes. The first conventional AND-type match-line scheme (CAML I) is shown in Fig. 3.16 (a), 128-bit word is divided into two parts, each part contains of 11 CAM segment[4]. The first CAM segment consists of 4 CAM cells; others are constructed by 6 CAM cells. The second match-line scheme called CAML II is shown in Fig. 3.16 (b) which consists of 8 CAM cells for each CAM segment. The search delay path belongs to CAML I is 11 segments plus the delay caused by AND-gate. On the other hand, search delay path belongs to CAML II is 8 segments plus the delay caused by AND-gate. Fig. 3.17 describes the two type waveforms of CAMLs during the search operation when the stored data of the word matches with search data. CAML II has 6% search time reduction compare to CAML I. For power consumption, 7.6% power reduction of CAML I compare to CAML II displayed in Fig. 3.18. According to above simulation results, we make the conclusions as follows. First, match-line divided into less segments results in shorter search time. Second, more segments decrease the power consumption.

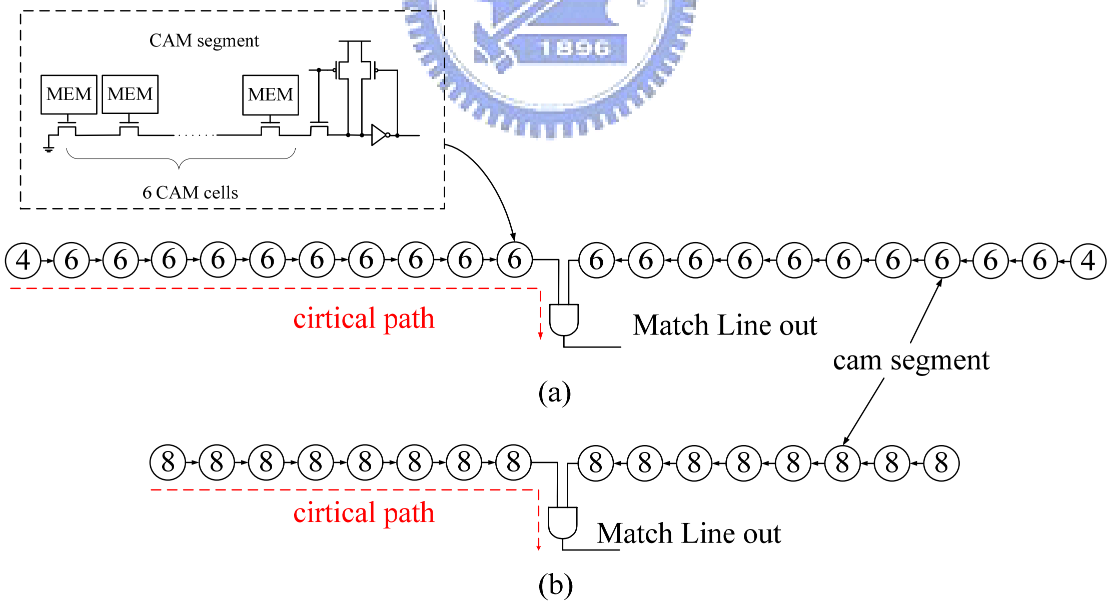


Fig. 3.16 Two types of pipelined AND-type match-line scheme. (a) CAML I (b) CAML II

[4] In this thesis, one segment of pipelined match-line scheme was called as CAM segment simply.

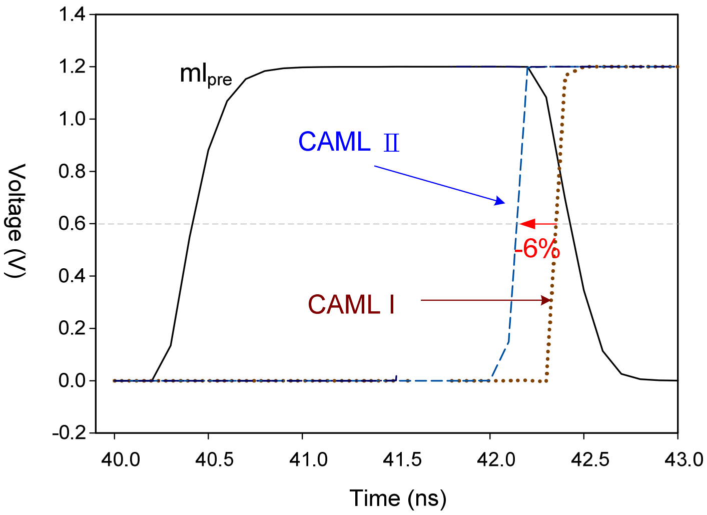


Fig. 3.17 Waveforms of two types of CAML in the search operation.

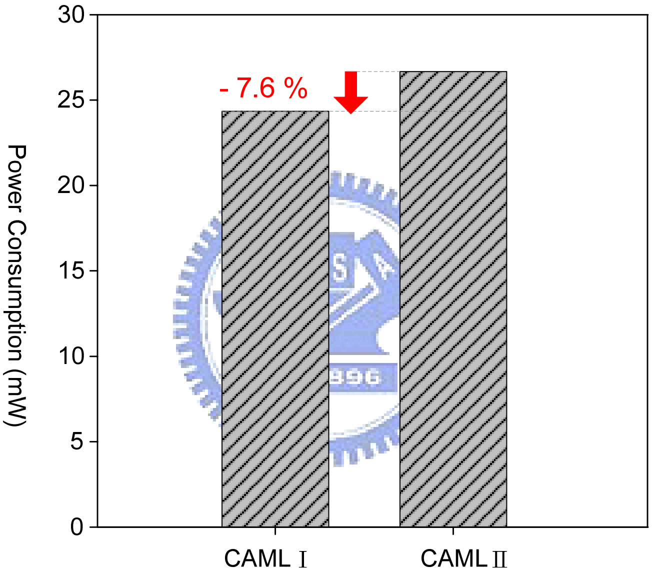


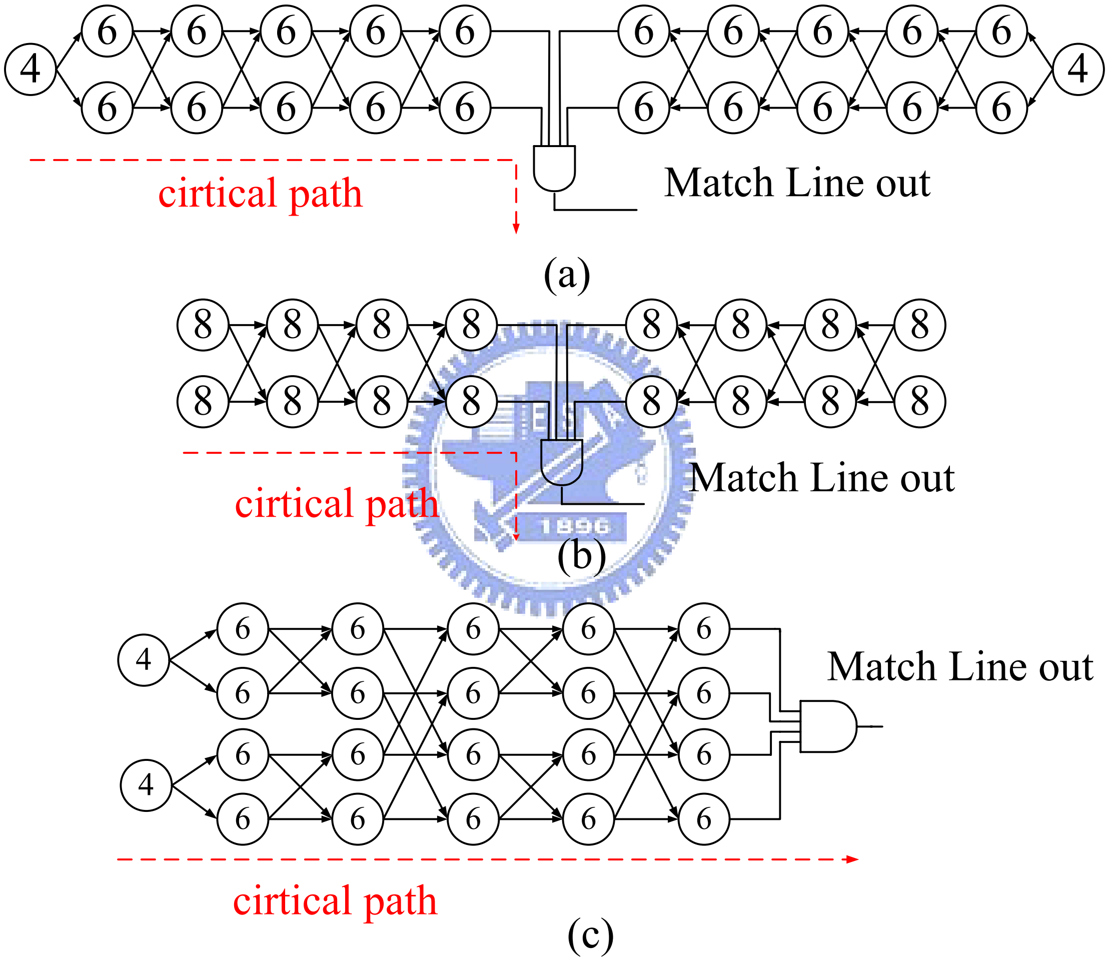
Fig. 3.18 Power comparison for two types of CAML.

3.5.3 Butterfly Match-line Scheme

3.5.3.1 Architecture

In order to reduce both search time and power consumption on match-line, butterfly match-line scheme is investigated. Three different butterfly schemes are proposed. Fig. 3.19 (a) shows the Type I butterfly match-line scheme. Unlike conventional AND-type match-line scheme shown in Fig. 3.16 (a), the relationship between each segment is not one by one connection but in interleaving style. The critical path is reduced from 11 segments to only 6 stages plus delay of an AND-gate.

If the propagation delay of one CAM stage is similar to one CAM segment. The proposed butterfly match-line scheme brings high search speed obviously. Fig. 3.19 (b) shows the Type II butterfly match-line scheme. Each CAM segment consists of 8 fan-in CAM cells. Although high fan-in property leads to poor propagation delay, Type II butterfly match-line scheme has only four stages on the critical path. Type III butterfly match-line scheme is shown in Fig. 3.19 (c). All the search operations at both sides of the match-line would be terminated when mismatch happened at any stage of the match-line.



(a)

(b)

(c)

Fig. 3.19 Butterfly match-line scheme. (a) Type I (a) Type II (a) Type III

In fact, the proposed butterfly match-line scheme is not only applying to AND-type match-line but also to NOR-type match-line. As mentioned in section 2.3.2, AND-type match-line scheme consumes less search power, but it takes longer search time to complete each search operation. Hence, applying butterfly match-line scheme to AND-type match-line to archive both high speed and low power CAM scheme is the main point in our design.

3.5.3.2 Circuit Implementation

For butterfly connection, there are additional circuits used to combine control signals came from different CAM segments which increase propagation delay on the critical path. Fig. 3.20 (a) shows the circuit implementation of Type I butterfly match-line scheme with conventional keeper. As mentioned in section 3.5.1, we assumed the propagation delay of CAM segment with conventional keeper is equal to two inverters. Take CAML I for example, delay of search operation equals to delay of 22 inverters plus one AND-gate. For Type I butterfly match-line scheme with conventional keeper, delay of search operation equals to delay of 12 inverters plus 5 AND-gates. However, an AND-gate composed of a NAND-gate and an inverter. It's obviously that search delay of Type I butterfly match-line scheme with conventional keeper is more than CAML I. In order to solve this problem, we have to do some logic optimization. The static inverter of CAM segment has two advantages. First, the fan-out of the gate is driven by a static inverter with a low impedance output which increases noise immunity. Also, the buffer reduces the capacitance of the dynamic output node by separating internal and load capacitances. Second, the inverter is used to drive a conventional keeper to combat leakage and charge re-distribution. Since the AND-gate also has the first advantage, we can try to combine these two inverters and one AND-gate to a NOR-gate. This can be realized by applying conditional keeper to butterfly match-line scheme. The XOR-based conditional keeper is turned off at some critical moments to reduce the delay and power consumption. A novel noise-tolerant butterfly match-line scheme is proposed. Fig. 3.21 shows the circuit implementation of Type I butterfly match-line scheme, and the delay of search operation equals to delay of 7 inverters plus 5 NOR-gates. Compare to CAML I, the critical path reduce from 23 to 12 logic gate delays.

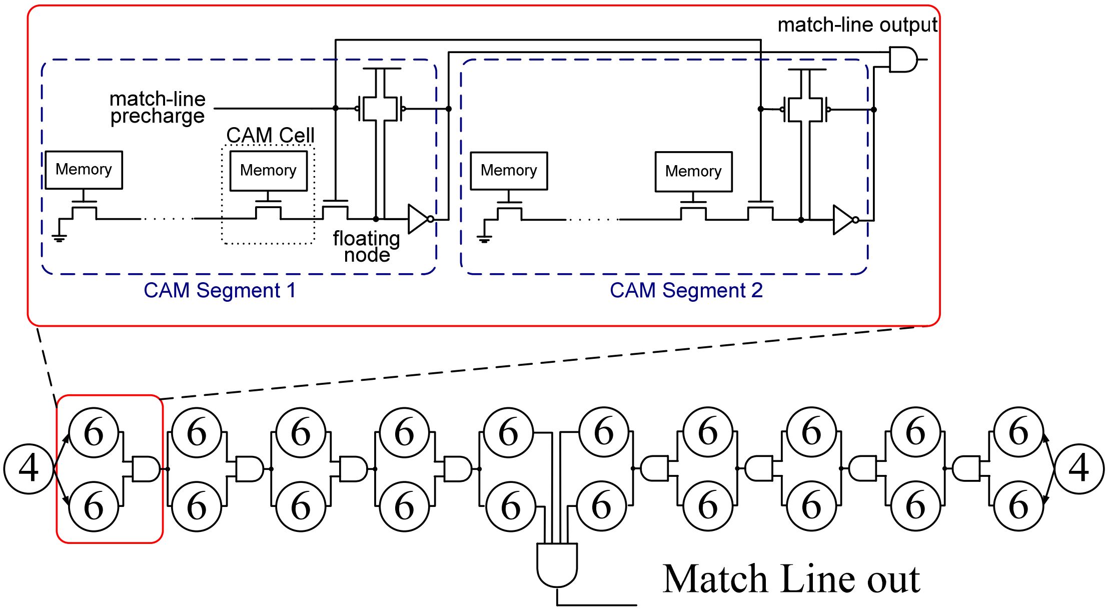


Fig. 3.20 Circuit implementation of Type I butterfly match-line scheme with conventional keeper

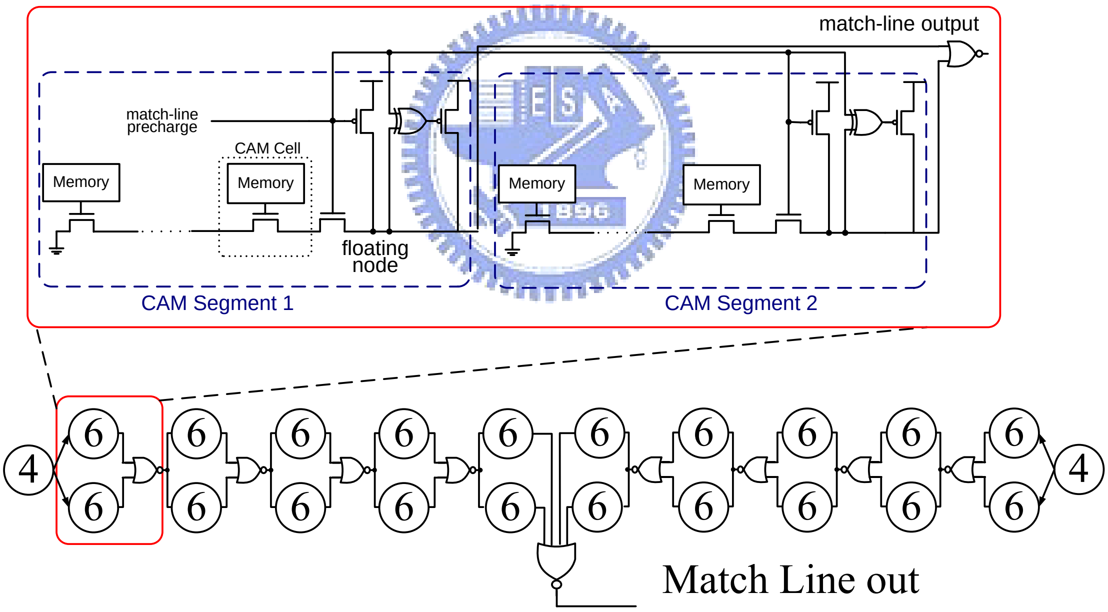


Fig. 3.21 Circuit implementation of Type I butterfly match-line scheme with XOR-based conditional keeper

3.5.4 Analysis of Power Consumption

Butterfly match-line scheme shortens the search time effectively. In the real, this scheme also reduces power consumption when search operation. The following section shows the analysis of power consumption on three different AND-type match-line schemes. We assume that each CAM cell with probability p (0 < p < 1) would match the search data.

3.5.4.1 Conventional AND-type Match-line

There are two types of conventional AND-type match-line presented in section 3.5.2. Consider of CAML I, the first stage consist of 4 CAM cells should allow only:

$$Pb_{S1} = p^{4} \tag{3.1}$$

CAM segments to survive to the second stage. For the second stage, there are only:

$$Pb_{S2} = Pb_{S1} \times p^{6} = p^{10} \tag{3.2}$$

CAM segments to survive to the third stage. We can get Pb$_{sn}$ as follows:

$$Pb_{S_n} = Pb_{S_{n-1}} \times p^{6} \qquad n = 2 \sim 11 \tag{3.3}$$

Assume each CAM cell consume the same power on match-line. The match-line power of CAML I can be written as follows:

$$\begin{aligned} P_{CAMLI} &= \frac{2\times 4}{128}P_{ML} + \frac{2\times 6}{128}P_{ML}Pb_{S1} + \frac{2\times 6}{128}P_{ML}Pb_{S2} + \ldots + \frac{2\times 6}{128}P_{ML}Pb_{S10} \\ &= \frac{2\times 4}{128}P_{ML} + \frac{2\times 6}{128}P_{ML}p^{4} + \frac{2\times 6}{128}P_{ML}p^{10} + \ldots + \frac{2\times 6}{128}P_{ML}p^{58} \end{aligned} \tag{3.4}$$

P$_{ML}$ is the non-pipelined match-line power consumption. For CAML II, the first stage consists of 8 CAM cells should allow only:

$$Pb_{S1} = p^{8} \tag{3.5}$$

CAM segments to survive to the second stage. Because each stage of CAML II consists of 2 eight-fan-in CAM segments, we can get Pb$_{sn}$ as follows:

$$Pb_{S_n} = Pb_{S_{n-1}} \times p^{8} \qquad n = 2 \sim 8 \tag{3.6}$$

The match-line power of CAML II can be written as follows:

$$\begin{aligned} P_{CAMLI} &= \frac{2\times 8}{128}P_{ML} + \frac{2\times 8}{128}P_{ML}Pb_{S1} + \frac{2\times 8}{128}P_{ML}Pb_{S2} + \ldots + \frac{2\times 8}{128}P_{ML}Pb_{S7} \\ &= \frac{2\times 8}{128}P_{ML} + \frac{2\times 8}{128}P_{ML}p^{8} + \frac{2\times 8}{128}P_{ML}p^{16} + \ldots + \frac{2\times 8}{128}P_{ML}p^{56} \end{aligned} \tag{3.7}$$

3.5.4.2 Tree-style AND-type Match-line

For further speed-up of the PF-CDPD AND scheme, three versions of tree-style match lines are investigated in [90]. Fig. 3.22 shows a two-level tree-style match-line is proposed to improve the speed. The speed of critical path contains 6 PF-CDPD-AND gates plus one 4-input static AND-gate is higher than CAML I due to the large speed difference between the 6-input and 4-input static AND gates and reduced interconnected RC delay.

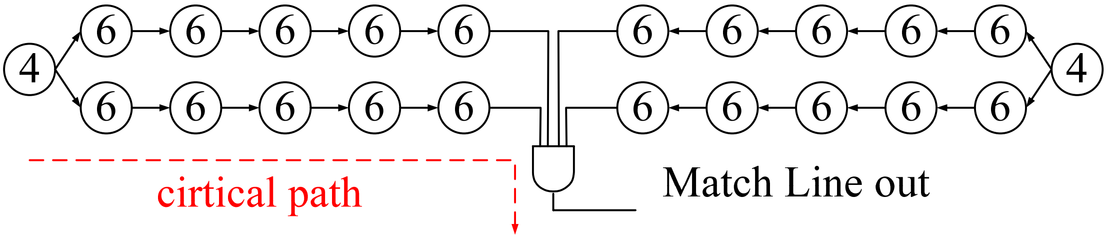


Fig. 3.22 2-level tree-style match-line scheme

The first stage consists of 4 CAM cells should allow only:

$$Pb_{S1} = p^4 \tag{3.8}$$

CAM segments to survive to the second stage which is same as CAML I. We can get Pb$_{sn}$ as follows:

$$Pb_{Sn} = Pb_{Sn-1} \times p^6 \qquad n = 2 \sim 6 \tag{3.9}$$

The match-line power of 2-level tree-style match-line scheme can be written as follows:

$$\begin{aligned} P_{tree} &= \frac{2\times4}{128}P_{ML} + \frac{4\times6}{128}P_{ML}Pb_{S1} + \frac{4\times6}{128}P_{ML}Pb_{S2} + \ldots + \frac{4\times6}{128}P_{ML}Pb_{S5} \\ &= \frac{2\times4}{128}P_{ML} + \frac{4\times6}{128}P_{ML}p^4 + \frac{4\times6}{128}P_{ML}p^{10} + \ldots + \frac{4\times6}{128}P_{ML}p^{28} \end{aligned} \tag{3.10}$$

3.5.4.3 Butterfly AND-type Match-line

Butterfly match-line scheme connects each CAM segment in interleaving style. Fig. 3.23 shows an example of mismatching Type I butterfly match-line scheme. The second stage on the left side has only one matching CAM segment. Therefore, the third stage on the left side is disabled. The first stage has the same Pb_{s1} compares to CAML I. However, only when both CAM segments are all exactly matched, the following stage will be enabled.

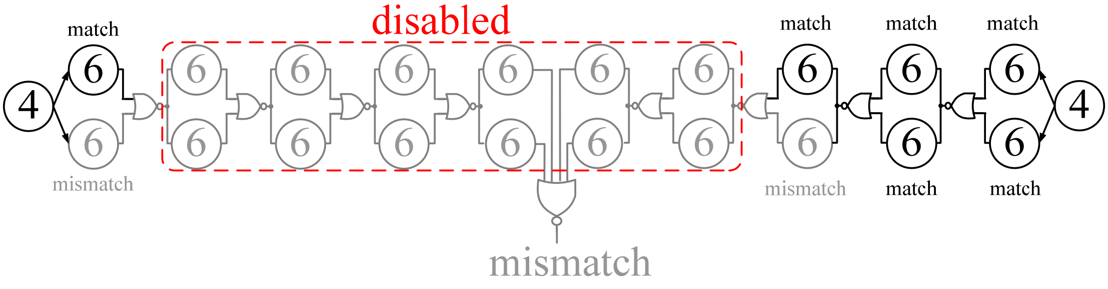


Fig. 3.23 A mismatch example of Type I butterfly match-line scheme

We can demonstrate that Pb_{sn} is as follows:

$$Pb_{Sn} = Pb_{Sn-1} \times p^{12} \qquad n = 2 \sim 6 \tag{3.11}$$

The match-line power of Type I butterfly match-line scheme can be written as follows:

$$\begin{aligned} P_{TypeI} &= \frac{2\times4}{128}P_{ML} + \frac{4\times6}{128}P_{ML}Pb_{S1} + \frac{4\times6}{128}P_{ML}Pb_{S2} + \ldots + \frac{4\times6}{128}P_{ML}Pb_{S5} \\ &= \frac{2\times4}{128}P_{ML} + \frac{4\times6}{128}P_{ML}p^{4} + \frac{4\times6}{128}P_{ML}p^{16} + \ldots + \frac{4\times6}{128}P_{ML}p^{52} \end{aligned} \tag{3.12}$$

Compare to tree-style match-line scheme, each stage only has half probability to be enabled after the second stage.

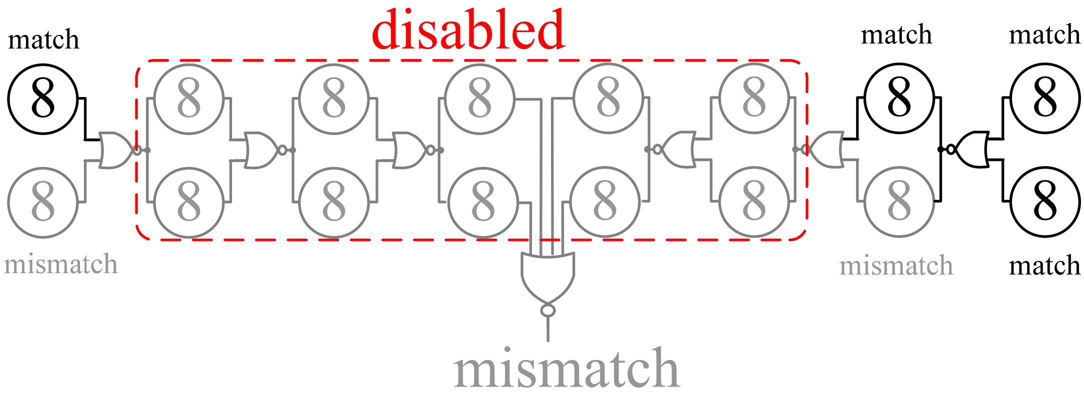


Fig. 3.24 A mismatch example of Type II butterfly match-line scheme

Fig. 3.24 shows an example of mismatching Type II butterfly match-line scheme. Both CAM segments of the first stage on the right side are matched; hence its following stage is enabled. Consider of the left side, one CAM segment of the first stage is mismatch. Therefore, the second stage on the left side is disabled.

We can demonstrate that $\mathrm{Pb_n}$ is as follows:

$$Pb_{S_n} = Pb_{S_{n-1}} \times p^{16} \qquad n = 1 \sim 4 \tag{3.13}$$

The match-line power of Type II butterfly match-line scheme can be written as follows:

$$\begin{aligned} P_{TypeII} &= \frac{4\times 8}{128} P_{ML} + \frac{4\times 8}{128} P_{ML} Pb_{S1} + \ldots + \frac{4\times 8}{128} P_{ML} Pb_{S3} \\ &= \frac{4\times 8}{128} P_{ML} + \frac{4\times 8}{128} P_{ML} p^{16} + \ldots + \frac{4\times 8}{128} P_{ML} p^{64} \end{aligned} \tag{3.14}$$

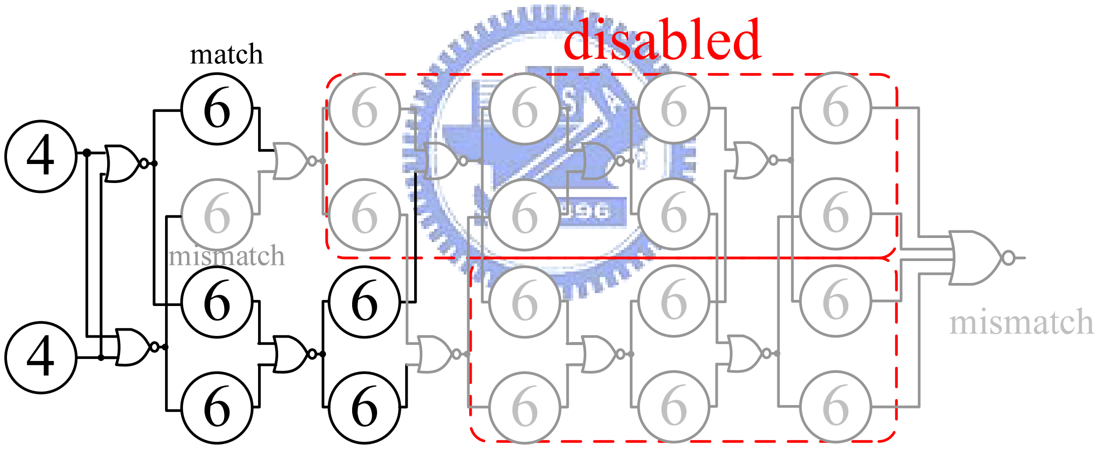


Fig. 3.25 A mismatch example of Type III butterfly match-line scheme

Fig. 3.25 shows an example of mismatching Type III butterfly match-line scheme. If any CAM segment in the same stage is mismatch, the stage after its following stage will be certainly disabled. This is the key feature of Type III butterfly match-line scheme. As shown in Fig. 3.25, one of the CAM segment in the second stage is mismatch. The fourth stage is disabled. The analysis of power consumption is presented as follows:

$$Pb_{S1} = p^{8} \tag{3.15}$$

$$Pb_{S_2} = Pb_{S1} \times p^{6} \times p^{6} = p^{20} \tag{3.16}$$

$$Pb_{S_3} = \left(Pb_{S2} \times p^{6}\right) \times \left(Pb_{S2} \times p^{6}\right) = p^{52} \tag{3.17}$$

$$Pb_{S4} = \left(Pb_{S3} \times p^{6} \times p^{6}\right) \times p^{6} \times p^{6} = p^{68} \tag{3.18}$$

$$Pb_{S5} = \left(Pb_{S4} \times p^{6} \times p^{6}\right) \times p^{6} \times p^{6} = p^{92} \tag{3.19}$$

The match-line power of Type III butterfly match-line scheme can be written as follows:

$$\begin{aligned} P_{TypeIII} &= \frac{2\times4}{128}P_{ML} + \frac{4\times6}{128}P_{ML}Pb_{S1} + \frac{4\times6}{128}P_{ML}Pb_{S2} + \ldots + \frac{4\times6}{128}P_{ML}Pb_{S5} \\ &= \frac{2\times4}{128}P_{ML} + \frac{4\times6}{128}P_{ML}p^{8} + \frac{4\times6}{128}P_{ML}p^{20} + \frac{4\times6}{128}P_{ML}p^{56} + \frac{4\times6}{128}P_{ML}p^{68} + \frac{4\times6}{128}P_{ML}p^{92} \end{aligned} \tag{3.20}$$

3.5.5 Performance Comparisons

Regarding performance comparisons, three 256 words x 128 bits content-addressable memories are designed. These three memories are connected with Type I, Type II, and Type III butterfly match-line scheme respectively. The circuit implementation is presented in section 3.5.3.2. We choose ternary-CAM as the memory cell, however, don't-care state will not be considered. Besides, I/O circuits and memory cells are similar to these three memories. Hence, we focus on search operation only. Two factors, search time and power consumption, will be discussed in this section. All simulation results are based on TSMC 130nm CMOS technology at 1.2V supply voltage.

3.5.5.1 Search Time

Search speed is one of the most important parts for performance comparison of CAM. The search time is defined as the period from clock becoming HIGH to match-line output which is matched with search data becoming HIGH, as shown in Fig. 3.26.

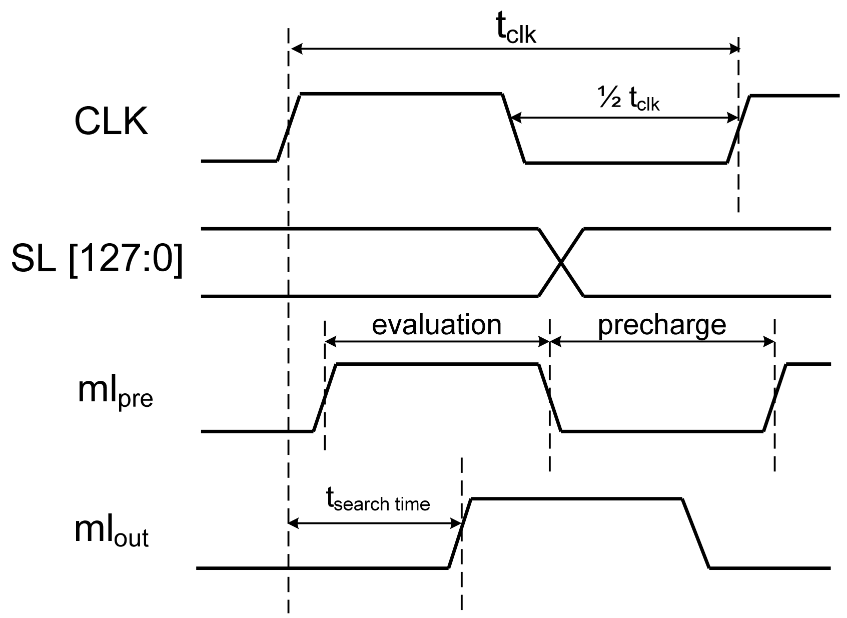


Fig. 3.26 Timing diagram of CAM during the search operation

The simulation result is shown in Fig. 3.27. Type II butterfly match-line scheme has the shortest search time 0.52ns. The maximum operation frequency for Type II is 625 Mhz. As mentioned in 3.5.3.1, Type II butterfly match-line scheme has only 4 stages on the critical path. Hence, Type II has fastest search speed. The search time for Type I and Type III are both 0.71ns. The main reason is that both Type I and Type III has 11 stages which is more than Type II. The maximum operation frequency for Type I and Type III is 500 Mhz.

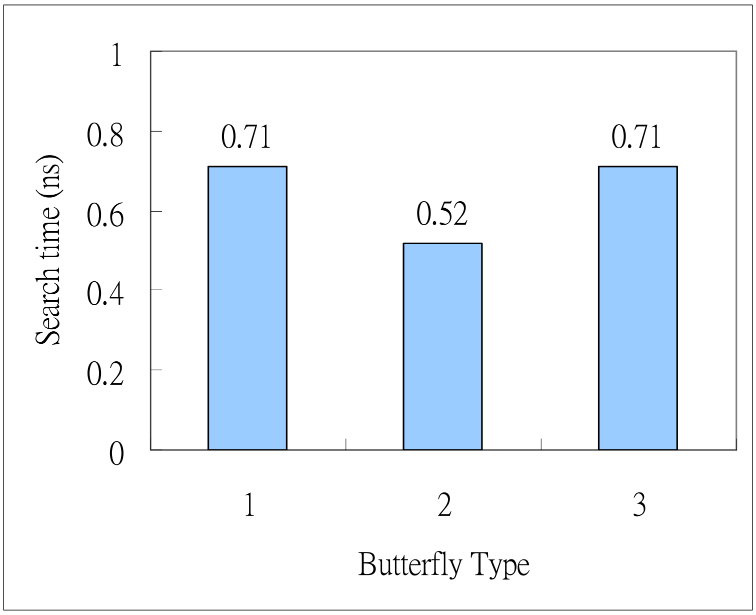


Fig. 3.27 Search time comparisons for three types butterfly match-line scheme

3.5.5.2 Power Consumption

Power consumption is another performance factor for CAM. Fig. 3.28 shows the simulation result. Type I and Type III butterfly match-line schemes operate at 500 Mhz. Type II operates at 625 Mhz. The normalized power consumption is shown in Fig. 3.29.

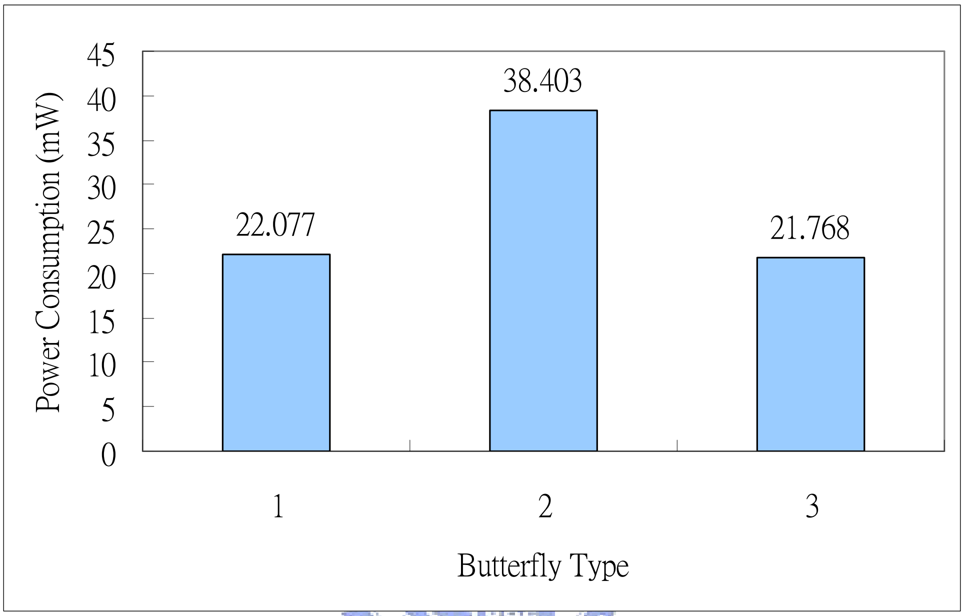


Fig. 3.28 Power consumptions for three types butterfly match-line scheme

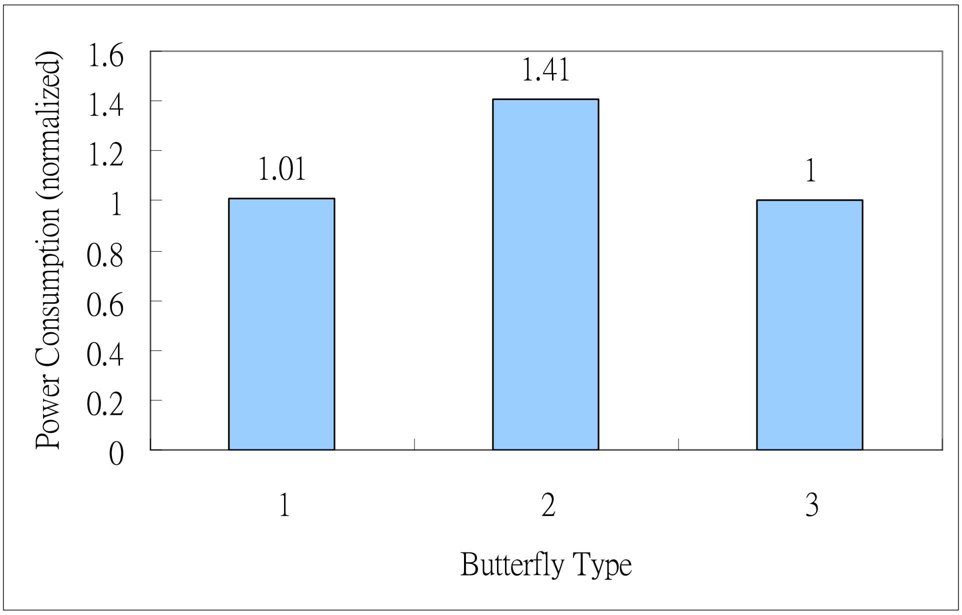


Fig. 3.29 Normalized power consumptions for three types butterfly match-line scheme

It is obvious that Type I and Type III butterfly match-line scheme consume less power than Type II. To explain the reason, recall from equation 3.12, equation 3.14 and equation 3.20. The first entry $\frac{4\times 8}{128}P_{ML}$ of equation 3.14 is much bigger than $\frac{2\times 4}{128}P_{ML}$ of equation 3.12 and equation 3.20. This is why Type II has larger power consumption. Compared with Type I and Type III butterfly match-line scheme, their power consumption are almost the same because the data stored in CAM is random generated. In this situation, the probability p which CAM cell would match the search data approaches to 0.5. Hence, the advantage of Type III butterfly match-line scheme could not be emerged.

3.6 Summary

In this chapter, three types of butterfly match-line scheme are proposed. Type I and Type III butterfly match-line schemes consume less power but longer search time compare to Type II. Compare to conventional AND-type match-line scheme [89], Type I, Type II and Type III reduce about 68.8%, 77.1%, 68.8% search time respectively. Not only search time, butterfly match-line scheme reduce power consumption also. 錯誤! 書籤的自我參照不正確。 shows the comparison for butterfly match-line scheme and conventional AND-type match-line scheme [89].

Table 3.2 Performance comparison for butterfly match-line scheme and conventional AND-type match-line scheme

	Butterfly match-line scheme			Conventional AND-type match-line scheme
	Type I	Type II	Type III	
Technology (μm)	0.13			0.1
Search time (ns)	0.71	0.52	0.71	1.75
Energy metric (fj/bit/search)	0.478	0.609	0.471	0.57

Chapter 4
Don’t-care Based Low Power Technique

CAMs are popular in network routers for packet forwarding and packet classification in the recent years. Network routers forward data packets from an incoming port to an outgoing port, using an address-lookup function. Ternary cells, in addition, store an “X” value. The “X” value is a don’t-care, that represents both “0” and “1”, allowing a wildcard operation. Wildcard operation means that an “X” value stored in a cell causes a match regardless of the input bit. This is a feature used in packet forwarding in Internet routers. In this chapter, we proposed two don’t-care based low power techniques to reduce match-line power and search-line power respectively.

4.1 IPv6 Addressing Architecture

IPv6 addresses are 128-bit identifiers for interfaces and sets of interfaces. There are three types of addresses:

Unicast: An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.

Anycast: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance).

Multicast: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.

The type of an IPv6 address is identified by the high-order bits of the address, as shown in Table 4.1:

Table 4.1 IPv6 address type identification

Address type	Binary prefix	IPv6 notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Site-local unicast	1111111011	FEC0::/10
Global unicast	(everything else)	

IPv6 unicast addresses are aggregable with prefixes of arbitrary bit-length similar to IPv4 addresses under Classless Interdomain Routing. There are several types of unicast addresses in IPv6, in particular global unicast, site-local unicast, and link-local unicast. There are also some special-purpose subtypes of global unicast, such as IPv6 addresses with embedded IPv4 addresses or encoded NSAP addresses. Additional address types or subtypes can be defined in the future.

4.1.1 Unspecified Unicast Addresses

The address 0:0:0:0:0:0:0:0 is called the unspecified address. It must never be assigned to any node. It indicates the absence of an address. One example of its use is in the Source Address field of any IPv6 packets sent by an initializing host before it has learned its own address.The unspecified address must not be used as the destination address of IPv6 packets or in IPv6 Routing Headers. An IPv6 packet with a source address of unspecified must never be forwarded by an IPv6 router.

4.1.2 Loopback Unicast Addresses

The address 0:0:0:0:0:0:0:0 is called the unspecified address. It must never be assigned to any node. It indicates the absence of an address. One example of its use is in the Source Address field of any IPv6 packets sent by an initializing host before it has learned its own address.The unspecified address must not be used as the destination address of IPv6 packets or in IPv6 Routing Headers. An IPv6 packet with a source address of unspecified must never be forwarded by an IPv6 router.

4.1.3 Global Unicast Addresses

The general format for IPv6 global unicast addresses is shown in Fig. 4.1, where the global routing prefix is a (typically hierarchically-structured) value assigned to a site (a cluster of subnets/links), the subnet ID is an identifier of a link within the site, and the interface ID is used to identify interfaces on a link. All global unicast addresses other than those that start with binary 000 have a 64-bit interface ID field (i.e., n + m = 64). Global unicast addresses that start with binary 000 have no such constraint on the size or structure of the interface ID field. Examples of global unicast addresses that start with binary 000 are the IPv6 address with embedded IPv4 addresses and the IPv6 address containing encoded NSAP addresses specified in [NSAP]. An example of global addresses starting with a binary value other than 000 (and therefore having a 64-bit interface ID field) can be found in [AGGR].

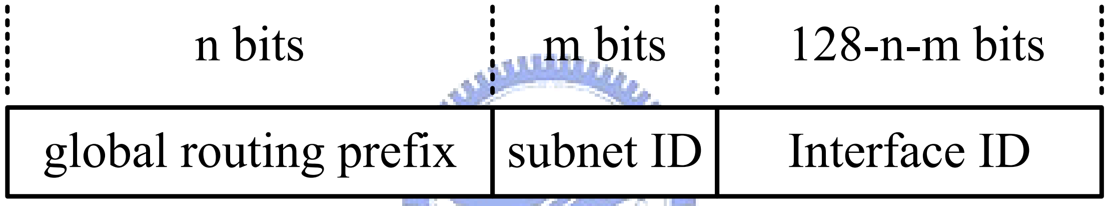


Fig. 4.1 The general format for IPv6 global unicast addresses

4.1.4 Local-Use IPv6 Unicast Addresses

There are two types of local-use unicast addresses defined. These are Link-Local and Site-Local. The Link-Local is for use on a single link and the Site-Local is for use in a single site. Fig. 4.2 (a) shows the format of Link-Local addresses. Link-Local addresses are designed to be used for addressing on a single link for purposes such as automatic address configuration, neighbor discovery, or when no routers are present. Routers must not forward any packets with link-local source or destination addresses to other links. Fig. 4.2 (b) shows the format of Site-Local addresses.

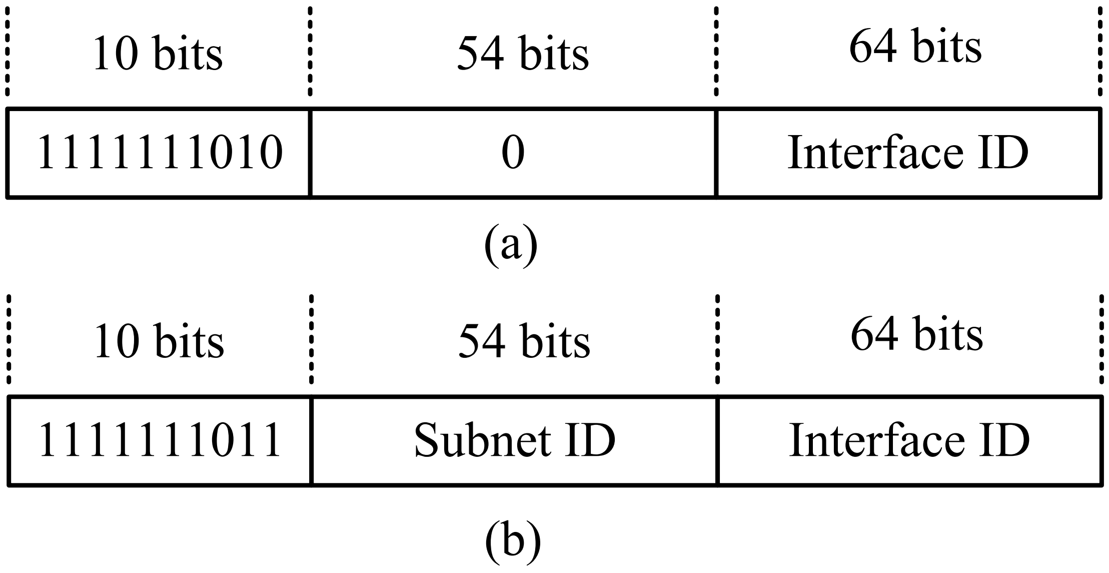


Fig. 4.2 The format of local-use IPv6 unicast addresses

4.2 IP Address Lookup

The primary role of routers is to forward packets toward their final destination. To this purpose, a router must decide for each incoming packet where to send it next. More exactly, the forwarding decision consists of finding the address of the next-hop router as well as the egress port through which the packet should be sent. This forwarding information is stored in a forwarding table computed by the router based on the information gathered by routing protocols. To consult the forwarding table, the router uses the packet’s destination address as a key; this operation is called address lookup. Once the forwarding information is retrieved, the router can transfer the packet from the incoming link to the appropriate outgoing link, in a process called switching.

One of the fundamental objectives of the Internet Protocol is to interconnect networks, so routing on a network basis was a natural choice (rather than routing on a host basis). Thus, the IP address scheme initially used a simple two-level hierarchy. This hierarchy is reflected in the fact that an IP address consists of two parts, a network part and a host part. The network part identifies the network to which a host is attached, and thus all hosts attached to the same network will be identical in the network part of their IP addresses. Since the network part corresponds to the first bit of the IP address, it is called the address prefix. With a two-level hierarchy, IP routers forwarded packets based only on the network part, until packets reached the destination network. As a result, a forwarding table only needed to store a single entry to forward packets to all the hosts attached to the same network. This technique is called address aggregation and allows prefixes used to represent a group of addresses.

To allow the use of the IP address space more efficiently and to slow down the growth of the backbone forwarding tables, a new scheme called classless inter-domain routing (CIDR) was introduced. CIDR allows finer granularity in the prefix lengths to achieve such advantage. That is why prefixes can be of arbitrary length. To address the problem of forwarding table explosion, CIDR allows address aggregation at several levels. The idea is that the allocation of addresses has a topological significance. Then, we can recursively aggregate addresses at various points within the hierarchy of the Internet's topology. As a result, backbone routers do not maintain forwarding information at the network level, but at the level of arbitrary aggregates of networks. Thus, recursive address aggregation reduces the number of entries in the forwarding table of backbone routers. A router must find the most specific match, which is the longest matching prefix. In summary, the address lookup problem in routers requires searching the forwarding table for the longest prefix that matches the destination address of a packet.

While CIDR allows the size of the forwarding tables to be reduced, the address lookup problem now becomes more complex. With CIDR, the destination prefixes in the forwarding tables have arbitrary lengths and no longer correspond to the network part since they are the result of an arbitrary number of network aggregations. Therefore, when using CIDR, the search in a forwarding table can no longer be performed by exact matching because the length of the prefix cannot be derived from the address itself. As a result, determining the longest matching prefix involves not only comparing the bit pattern itself, but also finding the appropriate length.

When the IPv6 routing protocol is introduced where the address length is 128 bits, the problem of routing millions of communication packets every second becomes a labyrinth. Especially, the most difficult point is based on longest prefix matching. In these circumstances, where IP routing tables are expanding in both the dimensions, i.e., address length and number of prefixes, the routing mechanisms developed should be capable of providing the throughput demand from high-speed transmission links.

Ternary content-addressable memory (TCAM) is a popular hardware device that can be used to perform fast lookups. TCAM adopts parallel technique and can reach O(1) lookup complexity. Compare with other lookup technologies, the advantages of TCAM are high lookup speed and simple operation. However, TCAM has three explicit disadvantages. First, with the same storage capacity, TCAM chips are much more expensive than SDRAM or DRAM chips. Second, TCAM chips perform key word comparisons parallel and power consumption is high. Third, to perform longest prefix matching lookups, prefixes must be arranged in order inside TCAM chips. Incremental prefix updates are rather complex and difficult. A packet routing based on longest prefix matching mechanism is shown in Fig. 4.3.

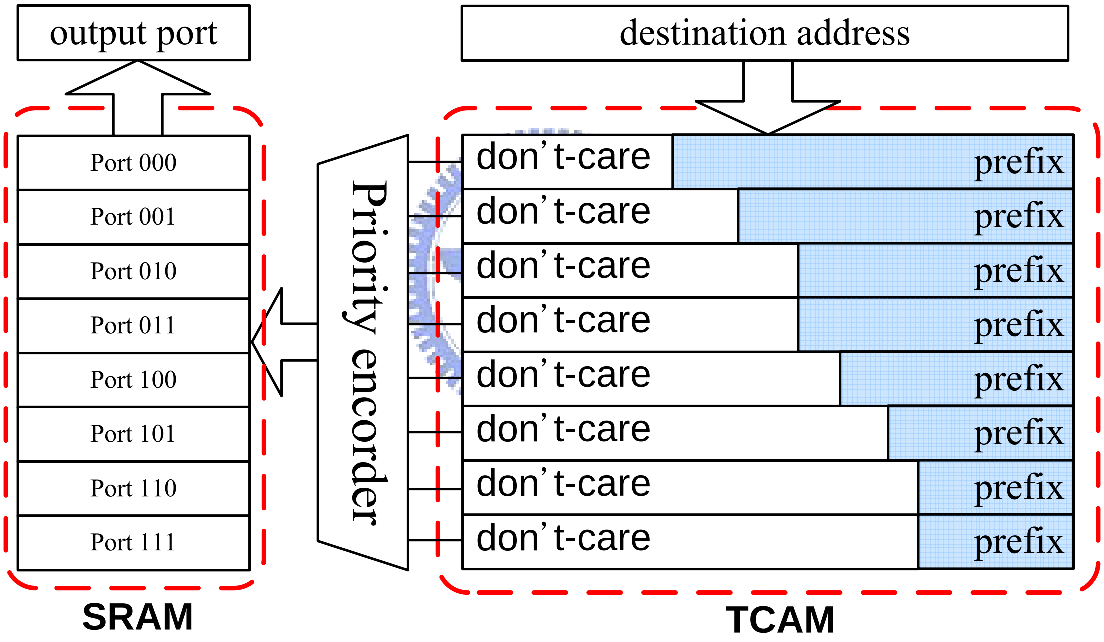


Fig. 4.3 Packet routing based on longest prefix matching mechanism

4.3 Don't-care Based Power-gating Technique

In general, the dynamic CMOS circuit operations are separated into two phases: 1) pre-charge phase and 2) evaluation phase. Whether NOR-type CAM word scheme or AND-type CAM word scheme at the pre-charge phase, the floating node would be charged to high by the PMOS, M_{pre}. Take AND-type match-line scheme for example shown in Fig. 4.4. If the all bits of search data match with stored data, the floating node would be discharged to ground. In other words, if any bit of search is not equal to stored data, the floating node would keep floating. Note that the match-line output stage of AND-type CAM is implemented by an inverter to ensure that the logic of match-line output is correct.

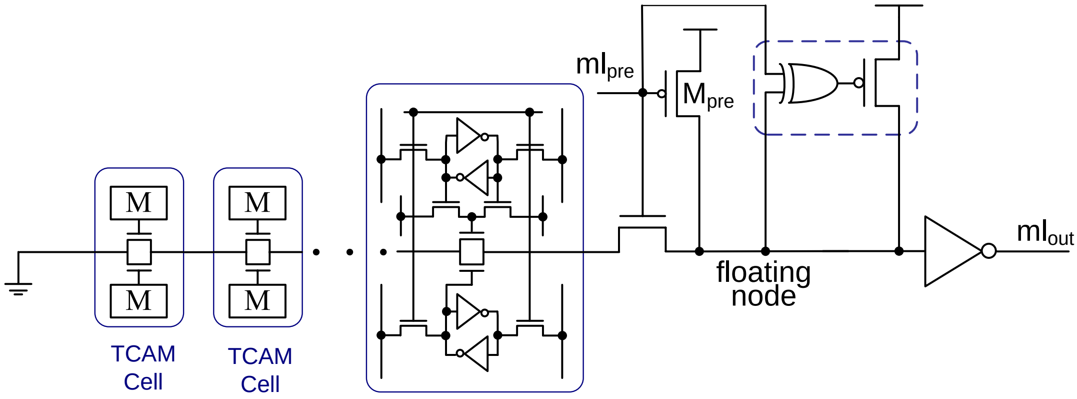


Fig. 4.4 AND-type TCAM match-line scheme with XOR-based conditional keeper

In IP addressing lookup application, TCAM has don’t-care state will always generate a matching signal. Consider of a single CAM segment, if all TCAM cells are don’t-care. The floating node must be discharged at evaluation phase in every search cycle. Fig. 4.5 shows the timing diagram. Since the match result is independent of input search data, precharge become pointless. If power-gating technique could be applied to disconnect voltage source at this moment, the dynamic power consumption of floating node can be avoided.

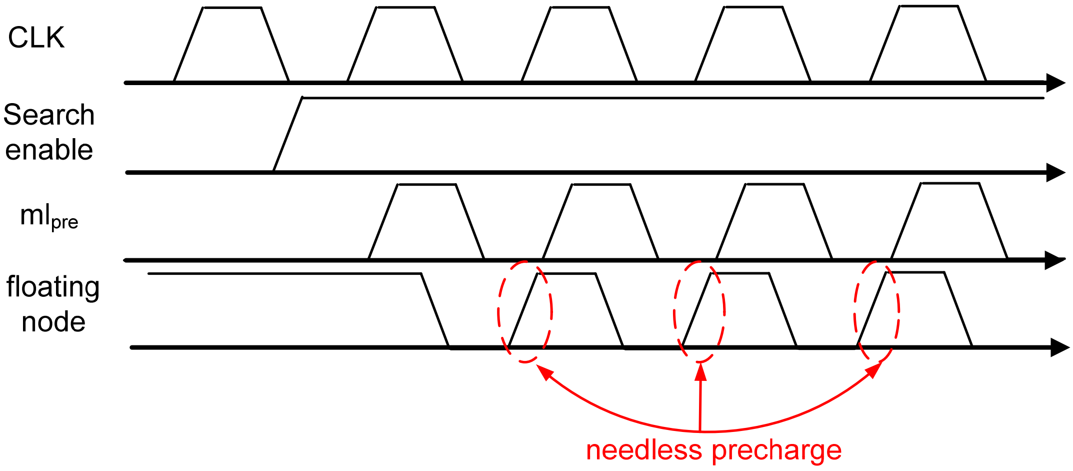


Fig. 4.5 Timing diagram of CAM segment when all TCAM cells are don’t-care.

As shown in Fig. 4.3, don’t-care bits are located at the left side of the primitive data element. Thus, if the MSB is don’t-care, all TCAM cells in this CAM segment are don’t-care. A novel don’t-care based power-gating match-line scheme is shown in Fig. 4.6. The corresponding timing diagram is shown in Fig. 4.7.

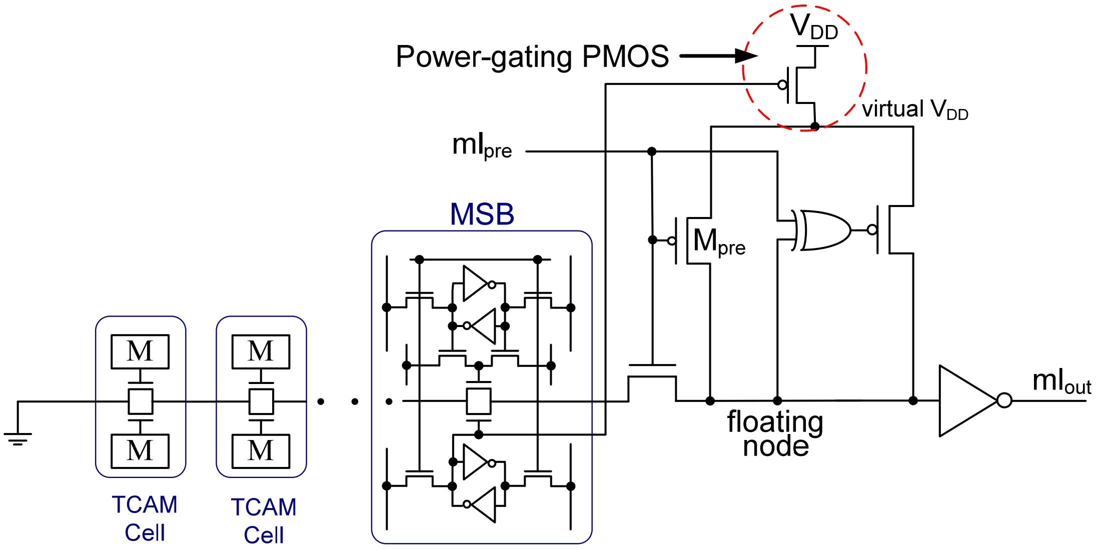


Fig. 4.6 Don’t-care based power-gating match-line scheme.

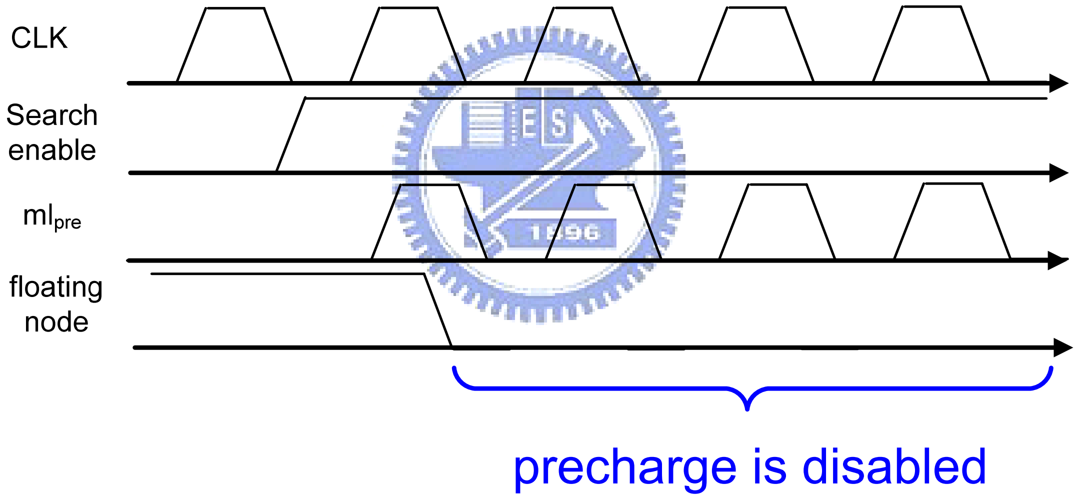


Fig. 4.7 Timing diagram of CAM segment with don’t-care based power-gating technique

When the MSB is don’t-care, the power-gating PMOS will be turn off. Charge path from V_{DD} to virtual V_{DD} is disconnected. Hence, the floating node will not be charged again. If the MSB is not don’t-care, the power-gating PMOS will be turn on. At this moment, floating node will be charged at precharge phase.

4.4 Don’t-care based Hierarchical Search-line scheme

Hierarchical search-line (HSL) scheme divides the search-lines into a two-level hierarchy, and they are global search-lines (GSLs) and local search-lines (LSLs). The GSLs are active every cycle, but the LSLs are active only when necessary. Conventional hierarchical search-line scheme was presented in section 2.5.2.1. The advantage is that an LSL will have no active match-lines in a given cycle in many cases. Hence, there is no need to activate the LSL. That is why the scheme can save power. However, the area and power overhead caused by the pipeline flip-flops and the clock driver diminishes the usefulness of the approach. The search speed is also been restricted by HSL scheme.

By taking the specific data pattern of IPv6 application, a novel don’t-care based hierarchical search-line scheme is proposed. Such scheme archives the same function as convention HSL scheme. However, it does not increase any search delay.

4.4.1 Architecture

If the TCAM cell is don’t-care, the matching signal is independent of input search data. In this situation, search-line can be disabled to save power. To archive this goal, search-line is divided into global search-line (GSL) and local search-line (LSL) respectively. A simplified architecture of proposed don’t-care based hierarchical search-line scheme is shown in Fig. 4.8. The entire TCAM is divided into four blocks. Each block consists of several math-lines and one global search-line to local search-line buffer (GSL-to-LSL buffer). For IPv6 addressing lookup application using TCAM, prefix is sorted in order of prefix length. Take Fig. 4.8 for example, the longest prefix is located at the bottom of the CAM. On the contrary, the shortest prefix is located at the top of the CAM. Hence, the upper CAM cell is certainly don’t-care. Hence, GSL-to-LSL buffer can be controlled by don’t-care state stored in the bottom CAM word in each block. Fig. 4.9 shows the circuit implementation of don’t-care based hierarchical search-line scheme. The GSL-to-LSL buffer is used to decide whether search data on GSL will broadcast to LSL or not.

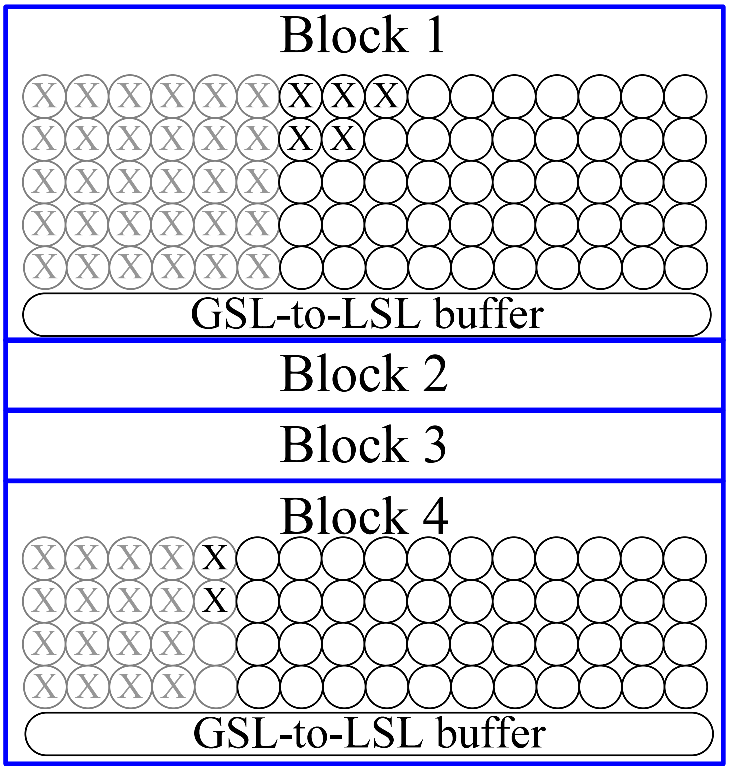


Fig. 4.8 A simplified architecture of don’t-care based hierarchical search-line scheme

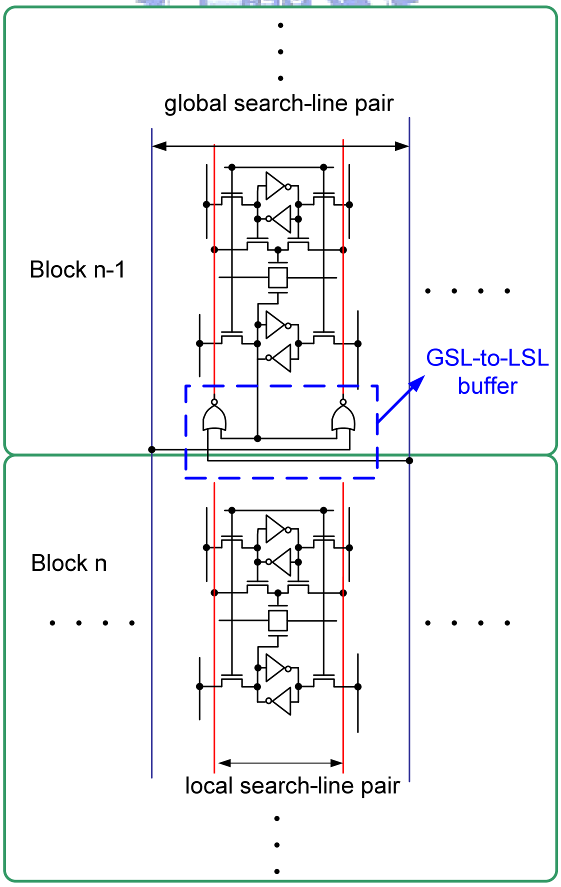


Fig. 4.9 Circuit implementation of don’t-care based hierarchical search-line scheme

4.4.2 Timing Analysis

The advantage comparing to conventional hierarchical search-line is that don't-care based hierarchical search-line do not increase search delay. Fig. 4.10 shows the timing diagram of don't-care based hierarchical search-line. Since control signal was stored into DFF in write operation, there is no timing overhead of control signal to control the GSL-to-LSL buffer. It is obviously that GSL-to-LSL buffer delay is shorter than half clock cycle. When clock is low, search data is already on the LSL. Hence, there is no speed overhead.

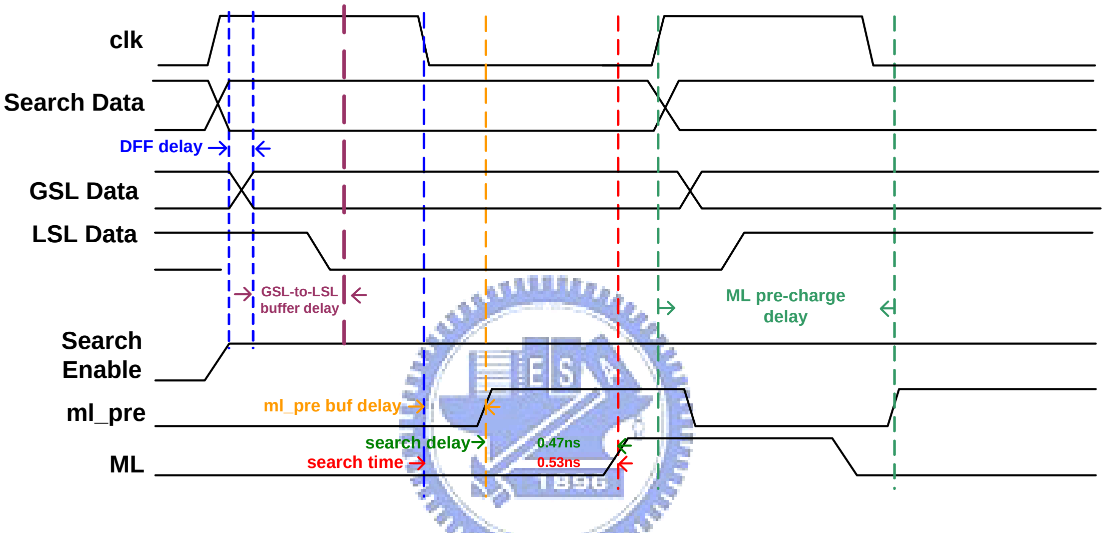


Fig. 4.10 Timing analysis of don't-care based hierarchical search-line scheme

4.5 Performance Comparisons

4.5.1 IPv6 prefix length distribution in the router

Don't-care based low power technique is based on properties of IPv6 addressing lookup table. In order to get more accuracy simulation result, we need to know the properties of routing table in current internet. Table 4.2 shows the prefix length distribution in the router of 6Bone. Obviously, we can observe two points from Table 4.2. First, the distributions of the prefix length equal to 32, 48, 35, 24, and 28 are approximately 86.21%, 4.76%, 3.78%, 1.31% and 1.31%. Second, there is no prefix length longer than 64 that is interface ID. The network address prefix distribution of IPv6 is shown in Fig. 4.11. According to this statistics, most TCAM cells in the routing table are don’t-care. Both power-gating and hierarchical search-line based on don’t-care are efficiently if there are more don’t-care TCAM cells.

Table 4.2 Prefix length distribution in the router of 6Bone

Prefix length	Route Entries	%
16	1	0.16
19	1	0.16
20	3	0.49
21	2	0.33
24	8	1.31
27	1	0.16
28	8	1.31
29	1	0.16
30	2	0.33
32	525	86.21
33	3	0.49
34	1	0.16
35	23	3.78
36	1	0.16
48	29	4.76

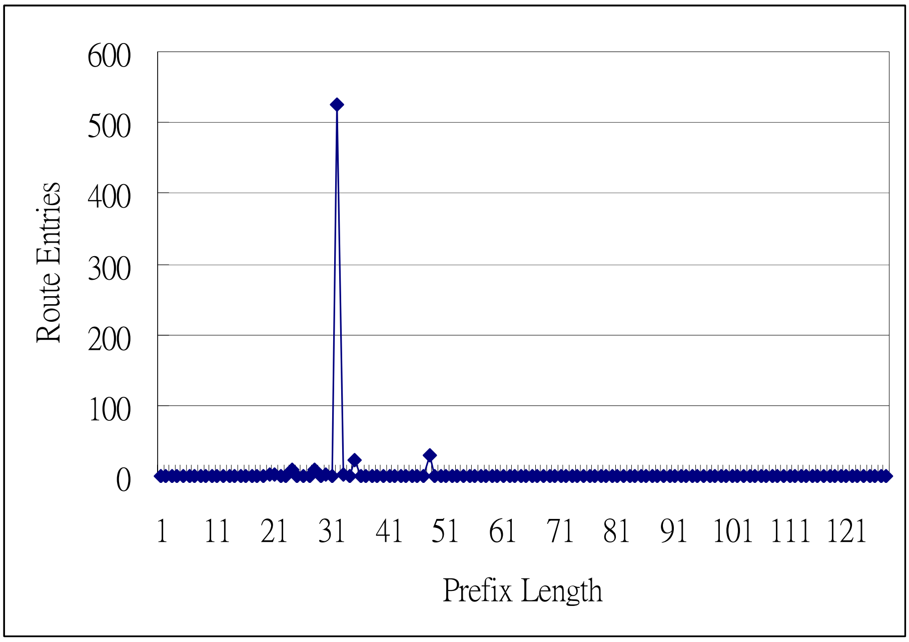


Fig. 4.11 Network address prefix distribution of IPv6

4.5.2 Simulation Result

We apply don't-care based power-gating technique to three types of butterfly match-line scheme to analyze the performance. The CAM size is 256 words x 128 bits for each type. The performance of don't-care based low power technique is related to the percentage of don't-care cells in entire CAM. Hence, we analysis 4 different density of don't-care cells to analysis this effect. All simulation results are based on TSMC 130nm CMOS technology at 1.2V supply voltage.

4.5.2.1 Don't-care Based Power-gating Technique

Fig. 4.12 shows the simulation results of Type I butterfly match-line scheme. The power consumption increases 2% due to additional power-gating circuit. When the percentage of don't-care cells increase, the power consumption increases also. Without don't-care based power-gating technique, 160% power is increased when there are more than 50% don't-care cells. If don't-care based power-gating technique could be applied, the power consumption are the same. Applying don't-care based power-gating technique to Type I butterfly match-line scheme saves about 153% power consumption when the percentage of don't-care cells increase.

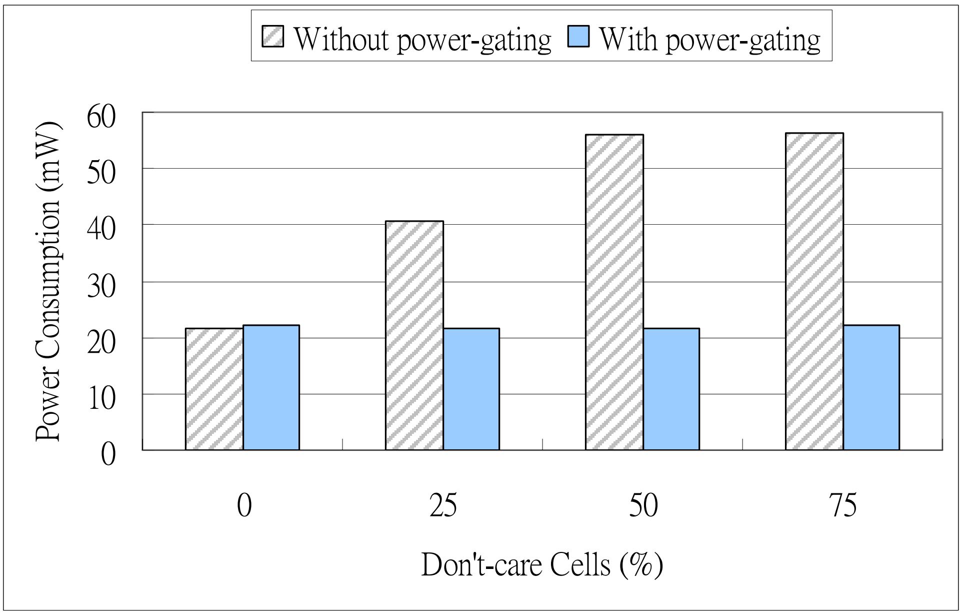


Fig. 4.12 Performance of Type I butterfly match-line scheme with don't-care based power-gating technique

For Type II butterfly match-line scheme, the simulation result is similar to Type I shown in Fig. 4.13. There is about 62% power consumption saved by applying don’t-care based power-gating technique (D-PG) to Type II butterfly match-line scheme. Compare to Type I with D-PG, the performance is limited by the first entry $\frac{4\times 8}{128}P_{ML}$ of equation 3.14. Type II butterfly match-line scheme consumes more power even if there is no don’t-care cells.

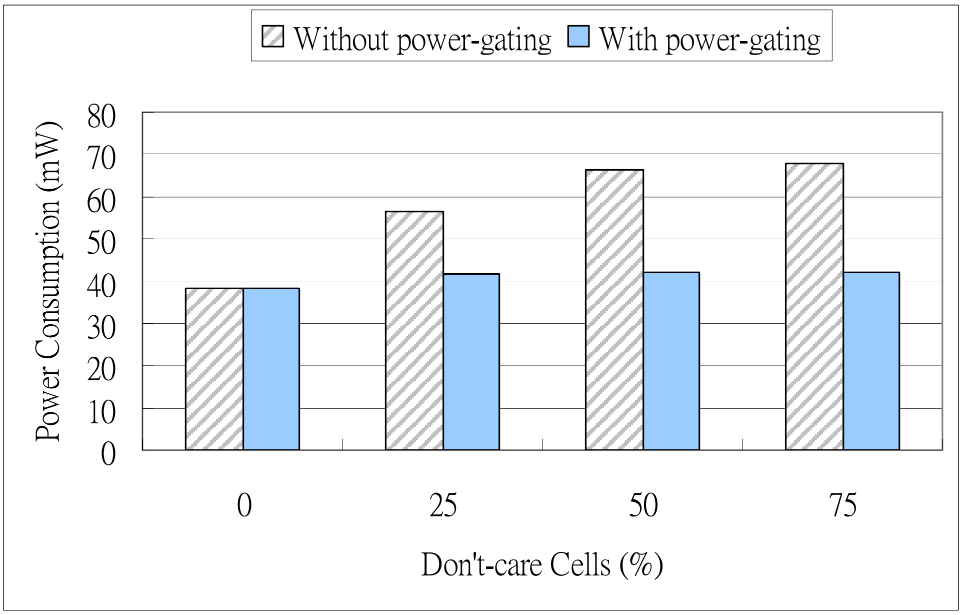


Fig. 4.13 Performance of Type II butterfly match-line scheme with don’t-care based power-gating technique

Similar to Type I with D-PG and Type II with D-PG, power consumption efficiently decreased by applying don’t-care based power-gating technique. However, only 59% of power consumption be saved. The real reason is the special connection of Type III butterfly match-line scheme make Type II immunity against of increasing don’t-care cells. Before applying don’t-care based power-gating technique, Type III consumes power about 35mW independent of percentage of don’t-care cells (except 0% don’t-care cells). After applying don’t-care based power-gating technique, the power consumption is kept at about 22mW.

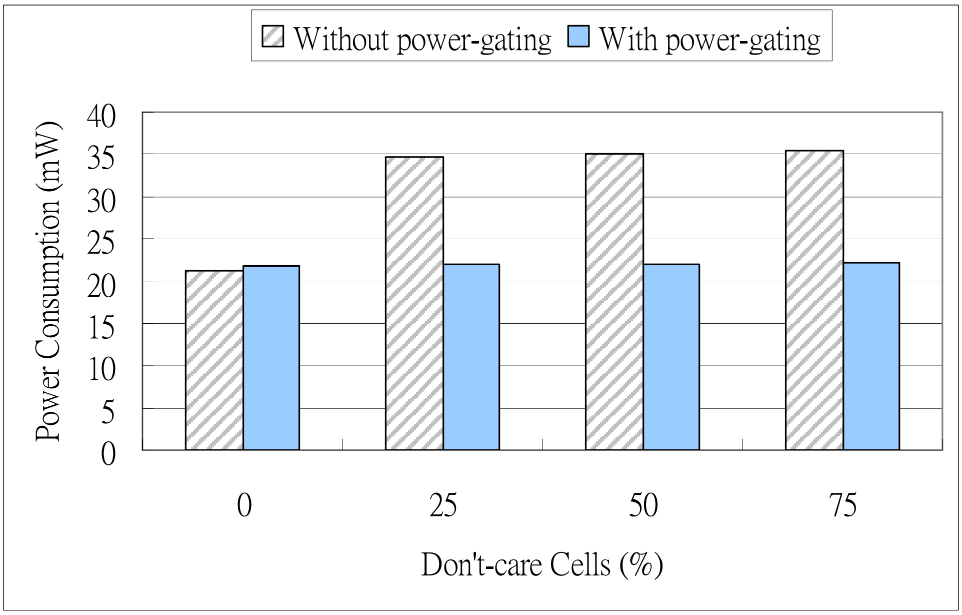


Fig. 4.14 Performance of Type III butterfly match-line scheme with don’t-care based power-gating technique

Fig. 4.15 shows performance analysis of butterfly match-line scheme without don’t-care based power-gating technique. In section 3.5.4, the equations show that Type III butterfly match-line scheme should consume least power consumption. However, the simulation results shown in Fig. 3.29 exhibit that same power consumption of Type I and Type II. When the percentage of don’t-care cells increase from 25% to 75%, power consumption of Type I and Type II butterfly match-line scheme are both increase. However, Type III consumes almost the same power even without don’t-care based power-gating technique. This can be explained as follows: For IP addressing lookup application, prefix is located at one side. Take Fig. 4.3 for example, prefixes are located at right side, don’t-care cells are located at left side. Since don’t-care cells results matching signal in every search operation, the floating node will be discharge in every cycles. However, the left side and right side are not independent in Type III butterfly match-line scheme. When the prefix at the right side results a mismatching signal, the search operation on the left side will be disabled. This is the most advantage of Type III butterfly match-line scheme. Since in IP addressing lookup application, don’t-care cells are certainly exist. Type III butterfly match-line scheme will be suit for this application. Fig. 4.16 shows the performance analysis of butterfly match-line scheme with don’t-care based power-gating technique. It’s obviously that after applying don’t-care based power-gating technique, power consumption on three types of butterfly match-line scheme are independent of don’t-care cells.

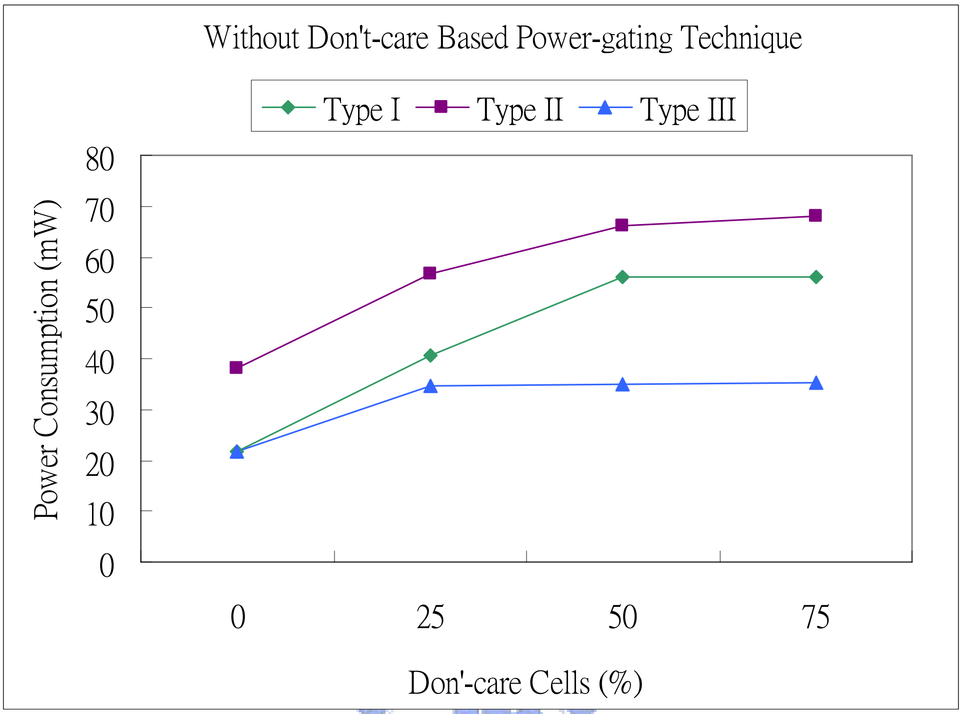


Fig. 4.15 Performance analysis of butterfly match-line scheme without don’t-care based power-gating technique

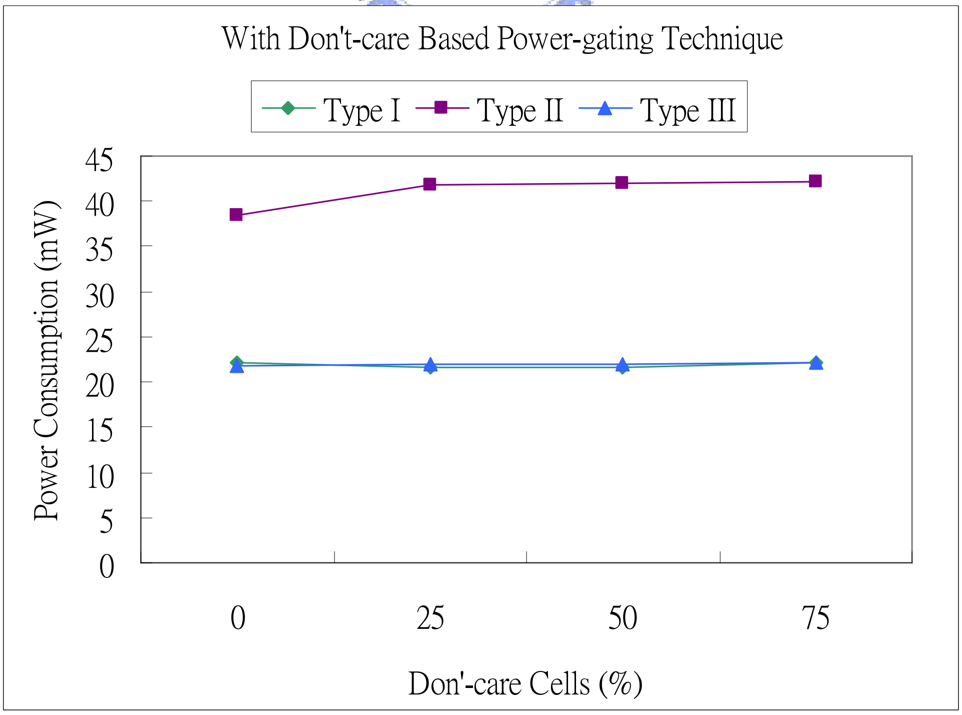


Fig. 4.16 Performance analysis of butterfly match-line scheme with don’t-care based power-gating technique

4.5.2.2 Don’t-care Based Hierarchical Search-line

Don’t-care based hierarchical search-line saves power consumption by disable local search-line. Fig. 4.17 shows the simulation result of Type I butterfly match-line scheme with don’t-care based hierarchical. We can note that power consumption is reduced even there is no don’t-care cells. This can be explained as follows: Search-line consumes large power due to large capacitances. In order to drive large capacitances, stronger search-line buffer should be used. Divide search-line into global search-line and local search-line can reduce capacitances. Unlike non-hierarchical search-line, there is less capacitance on local search-line. Stronger search-line buffer is unnecessary. Hence, size of search-line buffer can be reduced to save power consumption on search-line.

When the don’t-care cells increasing, the power consumption increasing also. This is because large numbers of floating node are discharge in search operation. However, the more don’t-care cells, the more local search-line disabled. This can be obvious from the percentage of don’t-care cells changes from 50% to 75%. Fig. 4.18 shows the simulation result of Type II butterfly match-line scheme with don’t-care based hierarchical search-line. Applying don’t-care based hierarchical search-line to Type I and Type II butterfly match-line scheme saves about 30% power consumption.

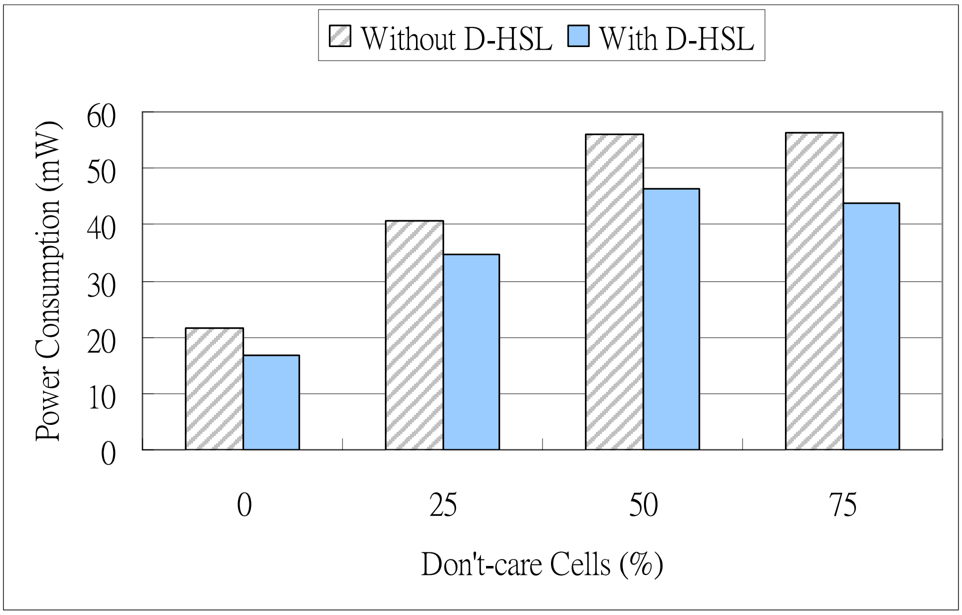


Fig. 4.17 Performance of Type I butterfly match-line scheme with don’t-care based hierarchical search-line

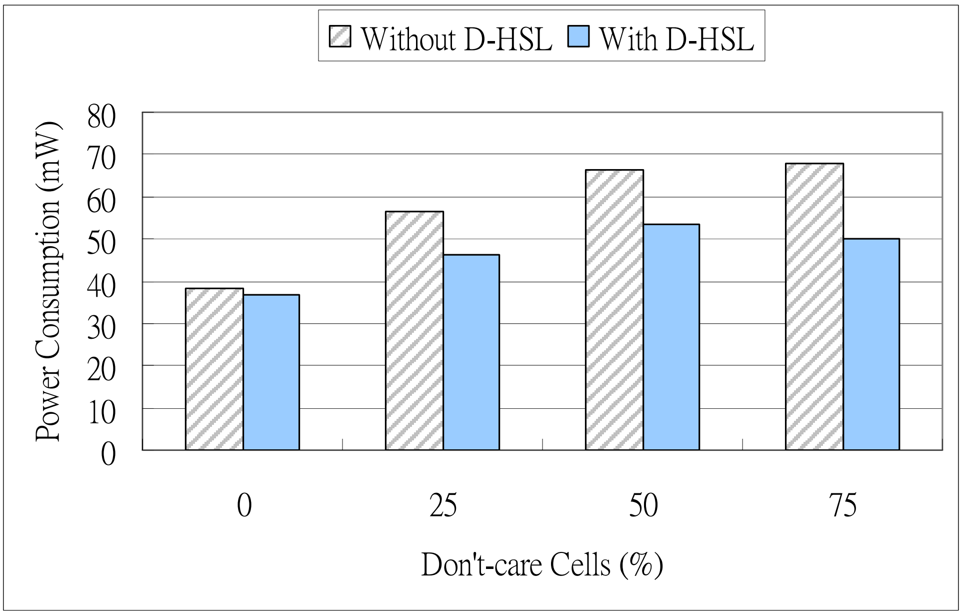


Fig. 4.18 Performance of Type II butterfly match-line scheme with don’t-care based hierarchical search-line

Unlike Type I and Type II butterfly match-line scheme, power consumption of Type III is insensitive to don’t-care cells. Hence, power consumption is reduced as don’t-care cells increasing obviously. Applying don’t-care based hierarchical search-line to Type III saves about 50% power consumption.

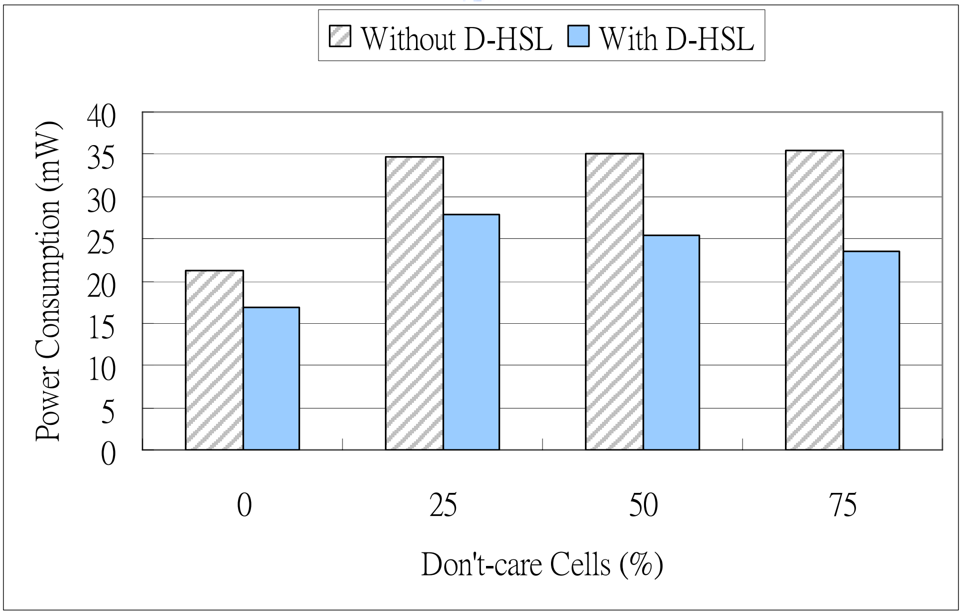


Fig. 4.19 Performance of Type III butterfly match-line scheme with don’t-care based hierarchical search-line

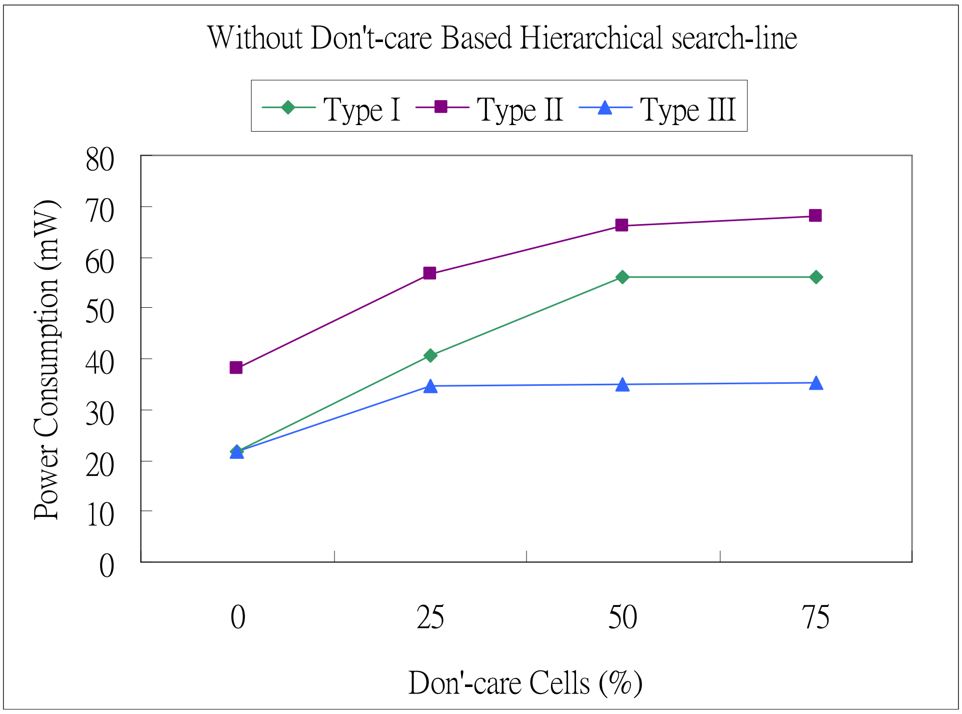


Fig. 4.20 Performance analysis of butterfly match-line scheme without don’t-care based hierarchical search-line

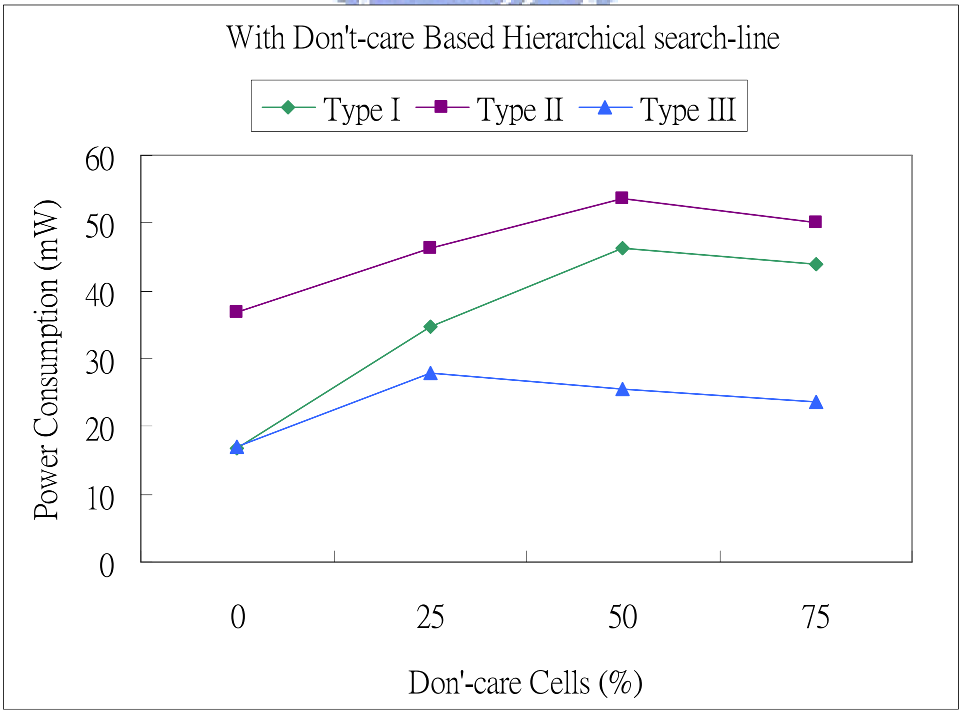


Fig. 4.21 Performance analysis of butterfly match-line scheme with don’t-care based hierarchical search-line

Fig. 4.20 shows the performance analysis of butterfly match-line scheme without don’t-care based hierarchical search-line. It’s obvious that Type III butterfly match-line scheme has the best performance. Applying don’t-care based hierarchical search line reduces power consumption efficiently as shown in Fig. 4.21. The more don’t-care cells, the more search-line disabled. For Type I, Type II and Type III, don’t-care based hierarchical search saves about 30%, 30%, 50.5% power consumption respectively.

4.6 Summary

In this chapter, don’t-care based low power technique is proposed to reduce power consumption on TCAM. Don’t-care based power-gating technique eliminates unnecessary precharge saves power consumption on match-line. For Type I butterfly match-line scheme about 160% power consumptions can be saved. Apply to Type II, 62% power consumptions can be saved. Type I and Type II butterfly match-line scheme, power consumption of Type III is insensitive to don’t-care cells. Without don’t-care based power-gating technique, Type III saves 60% power consumption compare to Type I. When the don’t-care based power-gating technique is applied to Type III, 59% power consumption can be saved.

Don’t-care based hierarchical search-line reduces the necessary driven strength of search-line buffer. The search-line power is reduced even there is no don’t-care cells inside CAM. By applying don’t-care based hierarchical search-line, about 30% power consumption saved for both Type I and Type II butterfly match-line scheme. For Type III, more than 50% power consumption can be saved. However, increase don’t-care cells also increase power consumption. Hence we make conclusion as follows: As the percentage of don’t-care cells increase, match-line power increase quickly. Although more don’t-care cells makes more local search-line disabled, power consumption saved by don’t-care based hierarchical is less than increased power consumption on match-line. The simplest way to solve this problem is apply don’t-care based power-gating technique and don’t-care based hierarchical search-line scheme at the same time. The analysis will present in chapter 5.

Chapter 5
Energy-Efficient Ternary Content-Addressable Memory for IPv6 Addressing Lookup Application

With the evolving nature of the Internet, the current Internet Protocol version 4 (IPv4) will be rapidly falling into disuse because of its limited address space, lack of needed functionality, and inadequate security features. With the manifest of the shortcomings of the existing IP, a new protocol, known as IPv6 (IP version 6), has been defined to ultimately replace IPv4 [91]. The addresses in the new Internet protocol are 128 bit long, whereas they are 32 bit long in the current IPv4-based Internet. Given the exponential growth of the Internet, and the advent of terabit per second transmission media, Internet routers are expected to make routing decisions within nanoseconds; the Internet will require high-speed forwarding engines. An important factor of these forwarding engines in an IPv6-based Internet is the speed and cost of doing routing table lookup on hierarchical 128-bit IPv6 addresses.

Ternary CAM ibe right back

ws extensively used in network applications. The main advantage of CAM is that the search time is bounded by one single memory access, thus it can guarantee a high lookup throughput. However, CAM consumes high power consumption due to fully parallel search operation and the switching of highly capacitive match-lines and search-lines.

In this chapter, we start from power sources in digital CMOS circuits. The MOSFET structure capacitances are presented in Section 5.2, and a proposed ternary CAM (TCAM) for IPv6 addressing lookup is introduced in Section 5.3. In order to design a low-power and high-speed TCAM, we modify the architecture, match-line schemes, search-line schemes of TCAM, and TCAM cells. Section 5.4 shows the simulation results. Furthermore, the layout is implemented in 0.13um CMOS technology and post-simulation is done in the Section 5.5. Finally, some conclusions and discussions are addressed in Section 5.6.

5.1 Power Sources in Digital CMOS Circuits

To design an energy-efficient ternary CAM (TCAM), low-power and high-speed design techniques are two definitely important issues. Here we aim at power consumption and introduce the three major components of digital CMOS integrated circuits: dynamic power, short-circuit power, and leakage power consumption, as follows.

5.1.1 Dynamic Power

Dynamic power due to charging and discharging capacitances is the dominant component among the three components of power sources, which is expressed as

$$P_{dynamic} = \alpha \cdot C_{switched} \cdot V_{DD}{}^{2} \cdot f_{clk} \tag{5.1}$$

where α stands for the switching activity, $C_{switching}$ represents the total effective switching capacitance, V_{DD} represents the power supply voltage, and f_{clk} is the switching frequency.

5.1.2 Short-Circuit Power

The second component of power consumption is the short-circuit power resulting from non-zero rise and fall times of the input waveforms. The non-zero input rise and fall times induce a direct path between V_{DD} and ground for a short time period during switching. Short-circuit power can be given by

$$P_{short-circuit} = t_{sc} \cdot V_{DD} \cdot I_{peak} \, f_{clk} \tag{5.2}$$

where t_{sc} represents the time that direct path is conducting.

5.1.3 Leakage Power

The third component is leakage power. The leakage current consists of various components, such as sub-threshold, band-to-band tunneling, gate tunneling, pn-junction reverse bias, DIBL, GIDL, and punch through leakage. However, sub-threshold leakage is the dominant one that given as

$$I_{leakage} = I_0 \exp\left(\frac{V_G - V_S - V_{T0} - \gamma V_S + \eta V_{DS}}{nV_{thermal}} \right) \cdot \left(1 - \exp\left(\frac{-V_{DS}}{V_{thermal}} \right) \right) \tag{5.3}$$

where $V_{thermal}$ is the thermal voltage, n is the sub-threshold swing coefficient constant, γ is the linearized body effect coefficient, and η is the DIBL coefficient.

5.1.4 Low-Power CAM Design

Consider that the dynamic power is the dominant component of power consumption, thus it is obvious that scaling down V_{DD} is the most efficient way to reduce dynamic power by eq. (5.1). However, reducing the power supply leads to the major influence of CAM, the stability/static noise margin (SNM) reduction. In chapter 3, butterfly match-line scheme not only increases search speed but also noise margin. For IPv6 addressing lookup application, search speed is also a key point. Hence, we wouldn't make any effort to reduce the frequency factor, f_{clk}, of eq. (5.1). Because of above two reasons, V_{DD} and frequency limitation, we just reduce the others factor of eq. (5.1), switching activity and switching capacitances, to decrease the dynamic power. Finally, in terms of short-circuit power, if reducing the time of direct path conducting, the short-circuit power will be reduced effectively.

5.2 MOSFET Structure Capacitances

From the foregoing description in eq. (5.1), it has been seen that higher switching capacitances cause higher power consumptions. Therefore, understanding more about the sources of switching capacitance could help us to reduce switching capacitances. In this section, we will discuss the MOS structure capacitances which comprise gate capacitances and junction capacitances.

5.2.1 Gate Capacitances

The gate capacitances, Cg, are decomposed into two elements, overlap capacitances and channel capacitances. Before starting to introduce these two types of capacitances, we have to know that the gate of the MOS transistor is isolated from the conducting channel by the gate oxide, which has a capacitance per unit area given by:

$$C_{ox} = \varepsilon_{ox} / t_{ox} \tag{5.4}$$

where ε_{ox} stands for the electrical permittivity of oxide and the t_{ox} is the oxide thickness.

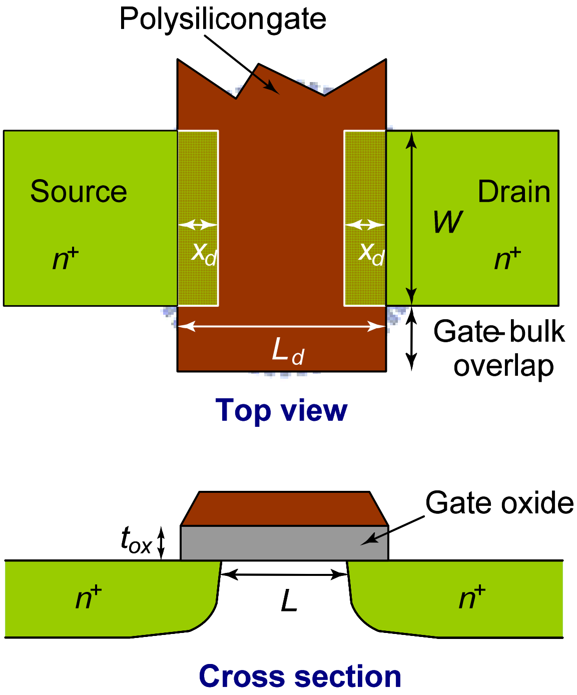


Fig. 5.1 MOSFET overlap capacitances.

5.2.1.1 Overlap Capacitances

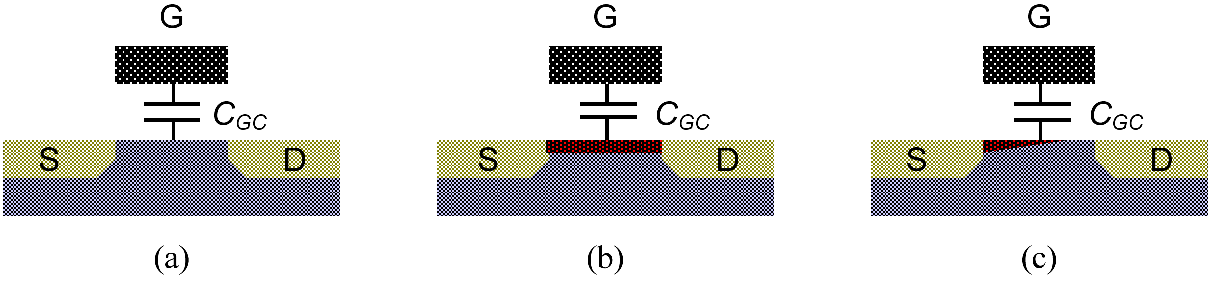
The MOSFET structure is depicted in Fig. 5.1. In real manufacture, both source and drain tend to extend somewhat below the oxide by an amount x_d, called the lateral diffusion. This phenomenon causes the parasitic capacitance between gate and source or between gate and drain that is called overlap capacitance. The value of overlap capacitance is derived as follows:

$$C_{GSO} = C_{GDO} = C_{ox} x_d W = C_o W \tag{5.5}$$

Since the x_d is determined by technology, the remain factor which can influence the value of overlap capacitance is the width of channel.

5.2.1.2 Gate-to-Channel Capacitances

The most significant MOS parasitic circuit element is the gate-to-channel capacitance C_{GC}. The gate-to-channel capacitance varies in both magnitude and in its division into three components C_{GCS}, C_{GCD}, and C_{GCB}, (C_{GCS} stands for gate-to-source, C_{GCD} means gate-to-drain, and C_{GCB} represents gate-to-body capacitances) depending on the operation regions and terminal voltages. This varying distribution is explained with the simple diagrams of Fig. 5.2. While the transistor works in the cut-off region (no channel exists) as depicted in Fig. 5.2 (a), and the total capacitance C_{GC} occurs between gate and body. In the resistive region (an inversion layer is formed) as illustrated in Fig. 5.2 (b), which acts as a conductor between source and drain. Consequently, $C_{GCB} = 0$ as the body electrode is shielded from the gate by the channel. Symmetry dictates that the capacitance distributes evenly between source and drain. Finally, in the saturation mode (the channel is pinched off) as illustrated in Fig. 5.2 (c). The capacitance between gate and drain approaches zero, and so is the gate-body capacitance. All the capacitances are therefore between gate and source. In order to make a simple analysis possible, a simplified piecewise-linear model with a constant capacitance value is adopted in each region of operation. The assumed values are summarized in Table 5.1.



(a) (b) (c)

Fig. 5.2 The gate-to-channel capacitance and how the operation region influences its

distribution over the three other terminals. (a) Cut-off. (b) Resistive. (c) Saturation.

Table 5.1 Average distribution of channel capacitance of MOS transistor for different operation regions.

Operation Region	C_{GCB}	C_{GCS}	C_{GCD}	C_{GC}	C_G
Cutoff	$C_{ox}WL$	0	0	0	$C_{ox}WL+2C_oW$
Resistive	0	(1/2) $C_{ox}WL$	(1/2) $C_{ox}WL$	$C_{ox}WL$	$C_{ox}WL+2C_oW$
Saturation	0	(2/3) $C_{ox}WL$	0	(2/3) $C_{ox}WL$	$(2/3)C_{ox}WL+2C_oW$

5.2.2 Junction Capacitances

The junction capacitances (also called the diffusion capacitances) are distributed by the reverse-biased source-body and drain-body pn-junctions. The structure of source (drain) region is shown in Fig. 5.3. The junction capacitances are formed by bottom-plate junction and side-wall junction. The detail value would be introduced in following:

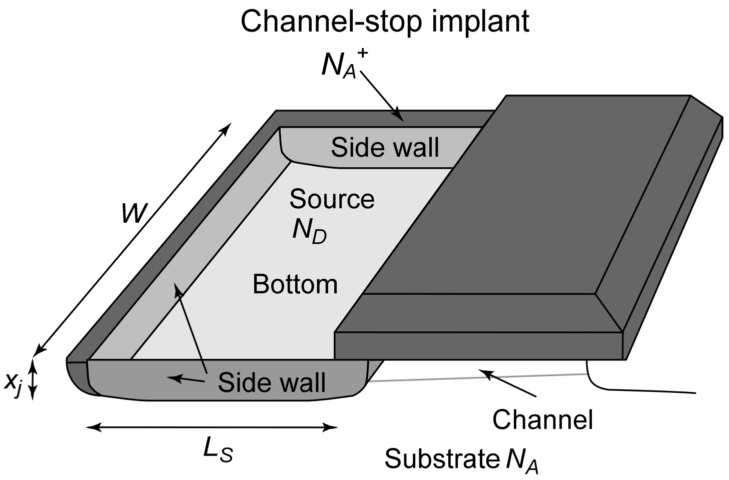


Fig. 5.3 Structure of source junction.

5.2.2.1 Bottom-Plate Junction Capacitances

The bottom-plate junction capacitance consists of the source region with doping N_D and the substrate with doping N_A. The total depletion region capacitance for this capacitance is as following:

$$C_{bottom} = C_j W L_S \tag{5.6}$$

Where C_j is the junction capacitance per unit area given by:

$$C_j = \frac{C_{j0}}{\left(1 - V_D/\phi_0\right)^m} \tag{5.7}$$

Where the C_{j0} is the capacitance under zero-bias conditions and ϕ_0 is the built-in potential in the junction under zero-bias. Finally, the m means the grading coefficient. As the bottom-plate junction typically is of the abrupt type, the grading coefficient m is approximatively equal to 0.5.

5.2.2.2 Side-Wall Junction Capacitances

The side-wall junction capacitance is formed by the source region with doping N_D and the p^+ channel-stop implant with doping level N_A^+. The capacitance value is given by:

$$C_{sw} = C'_{sw} x_j (W + 2 \times L_s) = C_{jsw}(2L_s + W) \tag{5.8}$$

Combine the eq. (5.6) and eq. (5.8), the junction capacitance value is expressed as:

$$C_{junction} = C_{bottom} + C_{sw} = C_j L_S W + C_{jsw}(2L_S + W) \tag{5.9}$$

From above discussions, it has been seen that overlap capacitance, gate-to-channel capacitance, and junction capacitance are all in proportion to width of channel. Therefore, if we want to reduce the switching capacitances, the simplest way is to reduce the size of loading transistors.

5.3 Proposed Ternary Content-Addressable Memory

A proposed ternary content-addressable memory (TCAM) will be introduced in this section. The low-power and high-speed TCAM is realized by applying butterfly match-line scheme and don’t-care based low power technique. We start from the architecture issues, and next one is match-line scheme design. Finally, we care about the TCAM cell implementation. The size of proposed TCAM array is a 256-word x 128-bit ternary CAM which is divided into two sub-arrays, as illustrated in Fig. 5.4.

5.3.1 Architecture

IPv6 addresses are 128-bit identifiers for interfaces and sets of interfaces. No mater NOR-type or AND-type match-line scheme, it takes long time to finish the search operation. In order to solve this problem, pipelined match-line scheme is strongly necessary. The other main point is that TCAM consume high power in search operation. Therefore, our goal is to reduce power consumption associated with the large amount of parallel active circuitry without sacrificing speed or memory density. NOR-type has high search speed property. When any bit is mismatch in NOR-type (implying that this word is mismatch), the match-line will be discharged. In IP addressing lookup application, destination address matches with more than one word in lookup table. Compare to entire IP addressing lookup table, the percentage of matching word is still few. Hence, NOR-type match-line scheme consumes a lot of power in search operation. Unlike NOR-type scheme, AND-type match-line scheme discharged only when all bits are matched (implying that this word is matched). It is obviously that AND-type match-line scheme consumes less power than NOR-type match-line scheme. However, two drawbacks of the AND match-line are a quadratic delay dependence on the number of cells, and a low noise margin.

In chapter 3, a novel butterfly match-line scheme is proposed to reduce the search delay. Furthermore, noise immunity is also improved by applying XOR-based conditional keeper. Thus, the disadvantage of AND-type match-line scheme could be solved. From eq. (5.1), decreasing switch activity is useful to reduce dynamic power consumption. Therefore, we choose the AND-type CAM architecture to implement the TCAM due to the switching activity issue. IPv6 addresses consists of 128-bit, this is unfavorable to search speed. In order to reduce the effect of this problem, we suggest the CAM array might be divided into two sub-arrays, as illustrated in Fig. 5.4. If the two sub-arrays architecture is adopted, being two smaller groups, the search time can be decreased effectively because the compared bits number of match-line is divided by two.

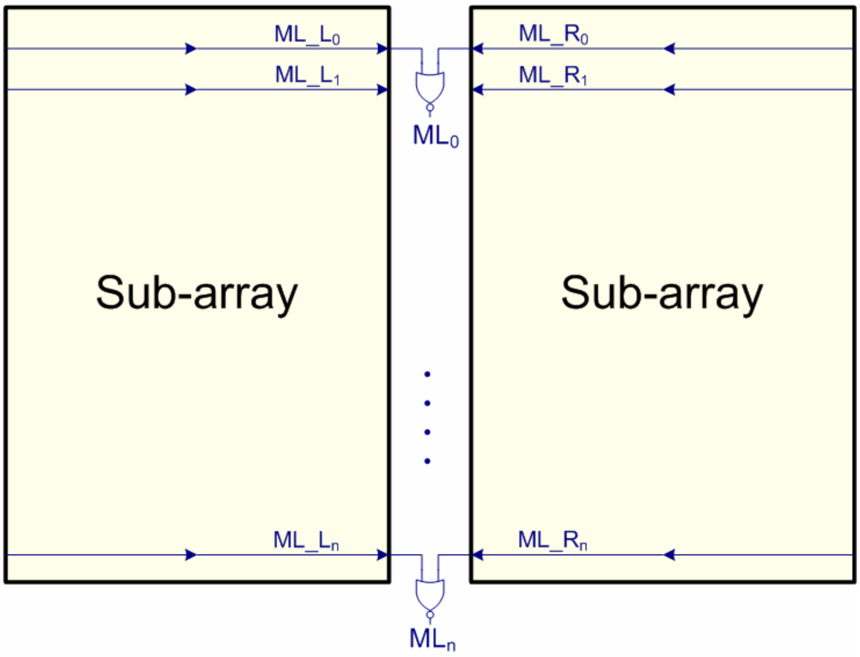


Fig. 5.4 CAM array divided into two sub-arrays.

From Table 4.2, most prefix length in IP addressing lookup table is short. If the CAM segment is don’t-care, precharge of match-line is unnecessary. Furthermore, don’t-care CAM segment results a matching signal independent of destination address. For further power reduction, don’t-care based low power technique presented in chapter 4 will be applied to our design. Don’t-care based power-gating technique reduces switching activity on match-line. Don’t-care based hierarchical search-line scheme reduce switching capacitance on GSL and switching activity on LSL.

5.3.2 Match-line Scheme

5.3.2.1 Butterfly AND-type match-line scheme for IPv6

There are three types of butterfly match-line schemes. The properties of IPv6 addressing architecture should be considered to get the best performance. In section 4.5.6, we know that most prefix length is shorter than 64-bit. Fig. 5.5 shows the IP addressing lookup table based on CAM. TCAM cell has don’t-care state will result a matching signal independent of destination address. Those don’t-care bits are located at the left sub-array. Hence, there are more matching CAM segments in left sub-array. If mismatch in right sub-array can also disable left sub-array, search power can be further reduced. As mentioned in section 3.5.4.3, Type III butterfly match-line has the property. If any CAM segment in the same stage is mismatch, the stage after its following stage will be certainly disabled. Therefore, Type III butterfly match-line scheme will be applied to our design. Fig. 5.6 shows the architecture applied to our design. The entire word is divided into two parts. If there is any one mismatching CAM segment both left and right sub-array can be stopped.

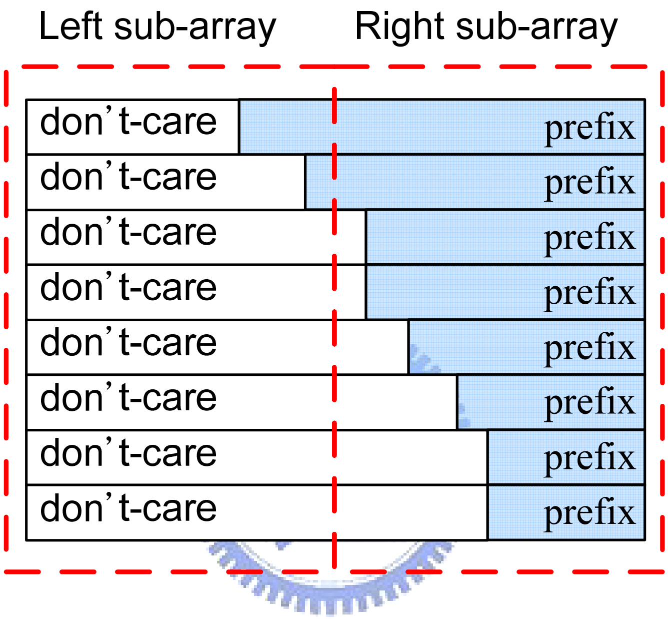


Fig. 5.5 IP addressing lookup table based on CAM

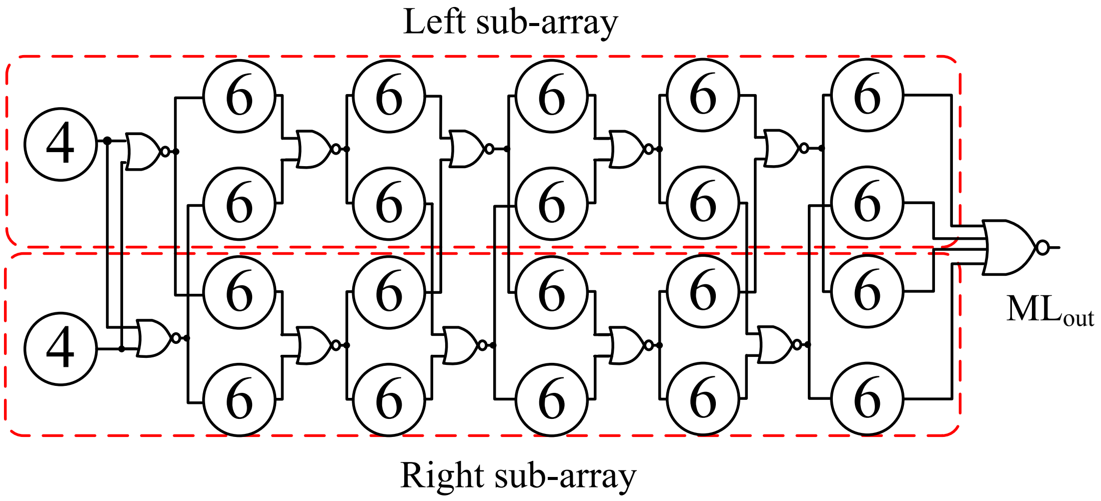


Fig. 5.6 Butterfly match-line scheme for IPv6 addressing lookup application

5.3.2.2 Applying Don't-care based Power-Gating Technique

Don't-care based power-gating technique prevents unnecessary precharge in search operation. Fig. 5.7 shows the circuit implementation of proposed match-line scheme. The power-gating PMOS is added between precharge circuit and power source. It's obviously, applying don't-care based power-gating technique do not increase search delay.

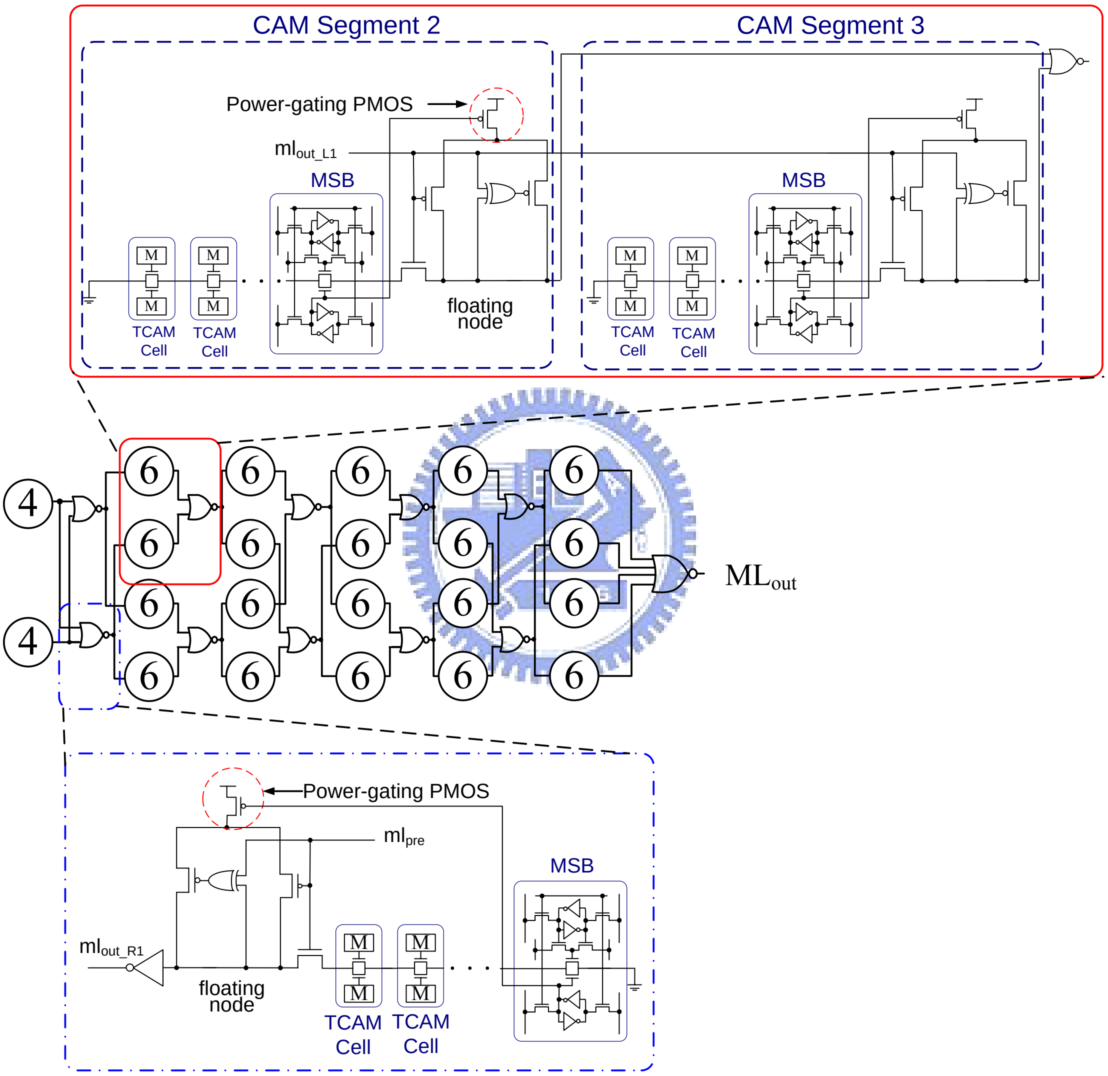


Fig. 5.7 Circuit implementation of proposed match-line scheme.

5.3.3 Ternary CAM Cell

In section 2.2.2.2, implementation of AND-type TCAM is clearly explained. However, the other important sources of power consumption are the high capacitive switching bit-lines and search-lines. Fig. 5.8 (a) shows the 16-transistor TCAM cell. In this TCAM cell, the bit-lines pair and search-lines pair are shared thus the switching capacitances (consists of junction capacitances) of these shared lines become extra large. Either the write operation or read operation, the bit-lines are switching frequently. Besides, during the search operation, the search-lines are also switching frequency. On the other words, these highly capacitive bit-lines/search-lines are always switching except TCAM working on standby mode; therefore, they cause large amount of power consumption. In order to avoid switching larger capacitances, we divide into two pairs of lines. One pair of lines are used to bit-lines and the other one are used to search-lines, as shown in Fig. 5.8 (b). We use an additional pair of lines, not only to reduce the power consumption during write, read, and search operations but also to simplify the peripheral circuits design. However, the penalty is that more lines complicate the layout router.

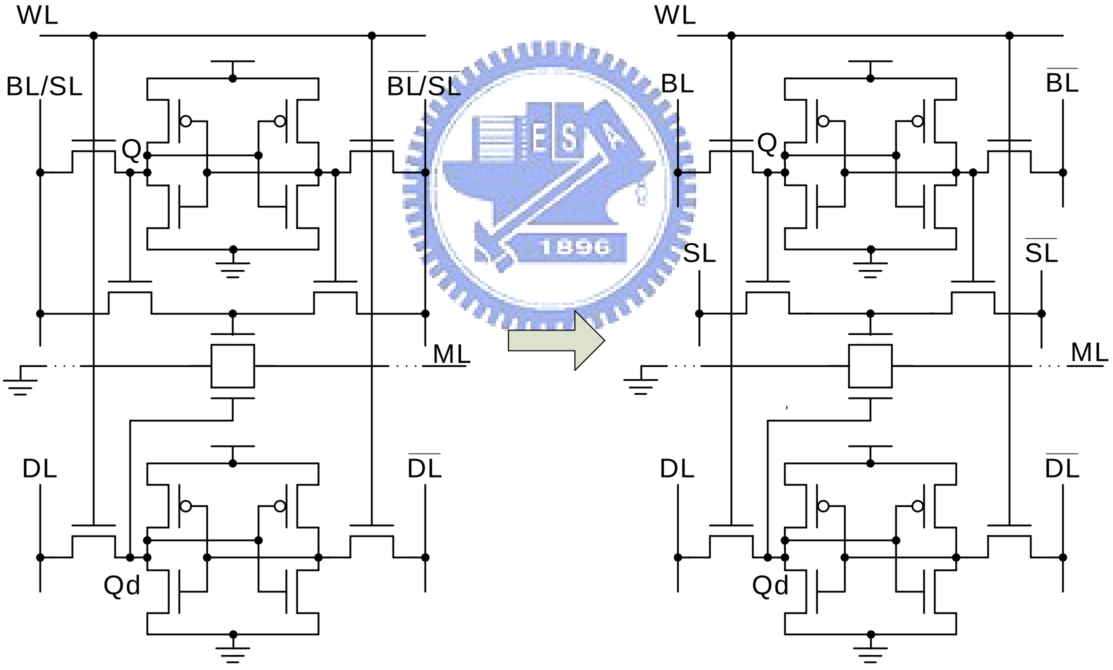


Fig. 5.8 16-transistor AND-type TCAM cell. (a) Bit-line and Search-line are shared. (b) Bit-line and Search-line are separated.

Table 5.2 State assignments for TCAM cell.

State	Q	Qd	SL	ML
Zero (0)	0	0	0	0
	0	0	1	floating

One (1)	1	0	0	floating
	1	0	1	0
Don’t Care (X)	0	1	0	0
	0	1	1	0
	1	1	0	0
	1	1	1	0

5.3.4 Don’t-care based hierarchical search-line scheme for IPv6

The proposed don’t-care based hierarchical search-line scheme is presented in chaper4. From Table 4.2, most prefix length is distributed from 16 to 48. Most prefix are shorter than 64. In our design, we divide 256 words into 5 blocks. Each block consists of 64 words for block1, block2 and block 3 respectively. For block 4 and block 5, each block consists of 32 words. There is one GSL-to-LSL buffer between each block. Fig. 5.9 shows the block diagram of proposed hierarchical search-line scheme.

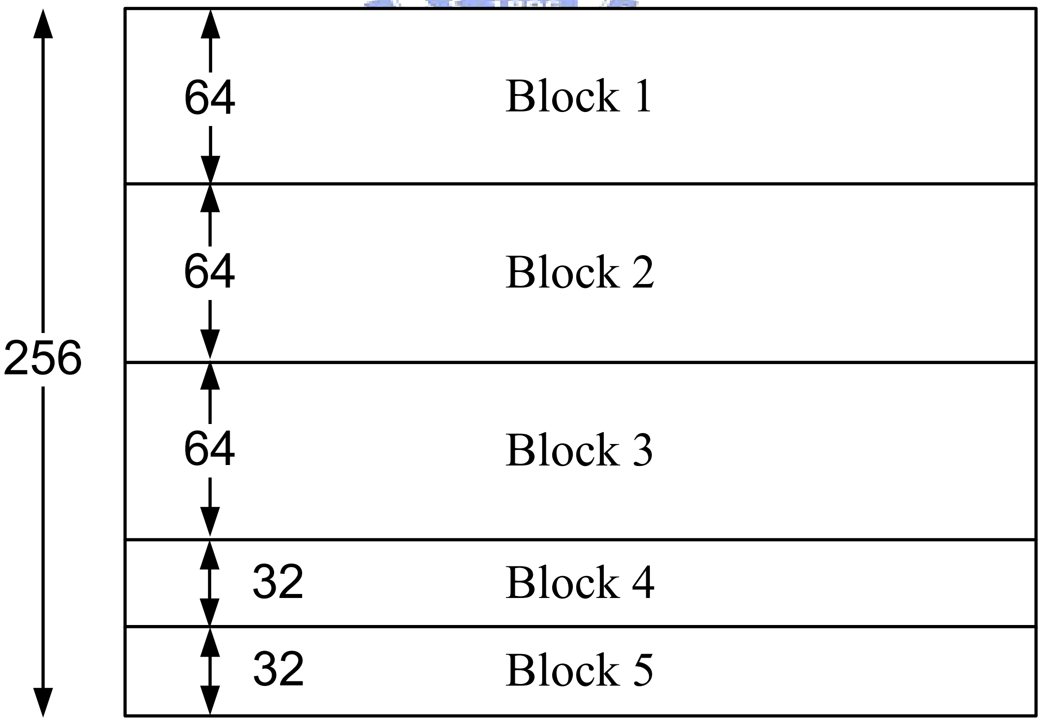


Fig. 5.9 Block diagram of proposed hierarchical search-line scheme

5.4 Simulation Results

In chapter 4, don’t-care based low power technique is proposed. In this section, we analysis the performance for butterfly match-line scheme combine with both don’t-care based power-gating technique and don’t-care based hierarchical search line. The CAM size is 256 words x 128 bits for each type. The performance of don’t-care based low power technique is related to the percentage of don’t-care cells in entire CAM. Hence, we analysis 4 different density of don’t-care cells to analysis this affection. All simulation results are based on TSMC 130nm CMOS technology at 1.2V supply voltage. The operation frequency for Type I and Type III butterfly match-line scheme is 500Mhz. Due to shortest search time, Type II butterfly match-line scheme operates at 625Mhz.

Fig. 5.10 show the performance compare for Type I butterfly match-line scheme. When there are no don’t-care cells, don’t-care based low power technique saves about 21% power consumption. While the don’t-care cells increase from 0% to 75%, the power consumption is decreasing from 16.98mW to 9.77mW. However, without don’t-care based low power technique power consumption is increasing from 21.615mW to 56.20mW. About 78% power consumption is reduced by applying don’t-care based low power technique. The energy metric for Type I with don’t-care based low power technique is 0.267fJ/bit/search.

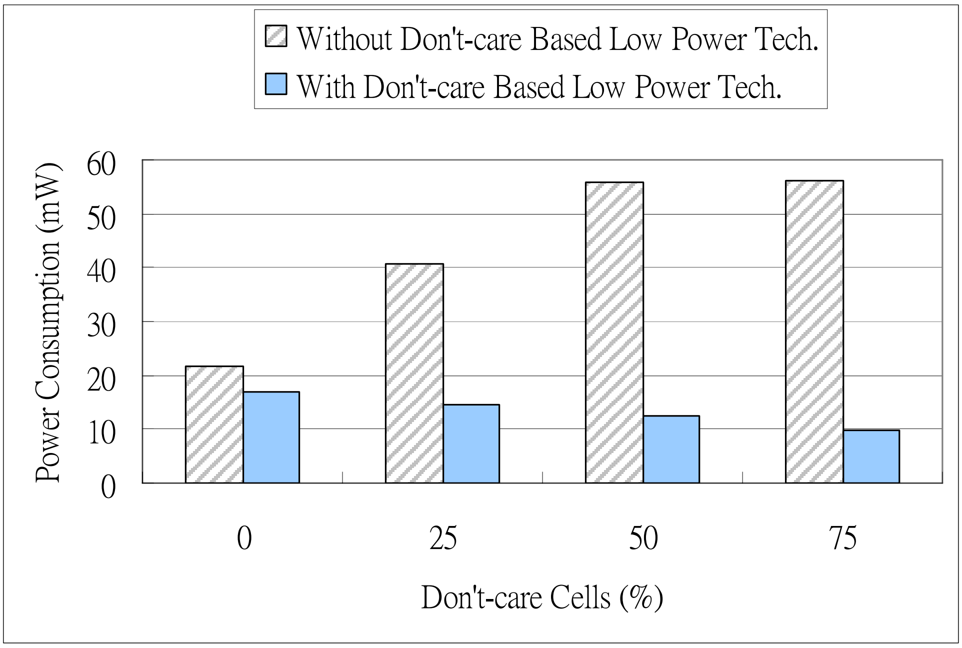


Fig. 5.10 Performance of Type I butterfly match-line scheme with don’t-care based

low power technique.

Type II butterfly match-line scheme has the fastest search speed compares to Type I and Type II. However, it consumes large power consumption. Fig. 5.11 shows the simulation result for Type II butterfly match-line scheme. Similar to Type I, while the don’t-care cells increase from 0% to 75%, the power consumption is decreasing from 36.588mW to 25.45mW. As mentioned at 3.5.5.2, the first entry $\frac{4\times 8}{128}P_{ML}$ of equation 3.14 is much bigger than $\frac{2\times 4}{128}P_{ML}$ of equation 3.12 and equation 3.20. This is why Type II has larger power consumption. The energy metric for Type II with don’t-care based low power technique is 0.396fJ/bit/search.

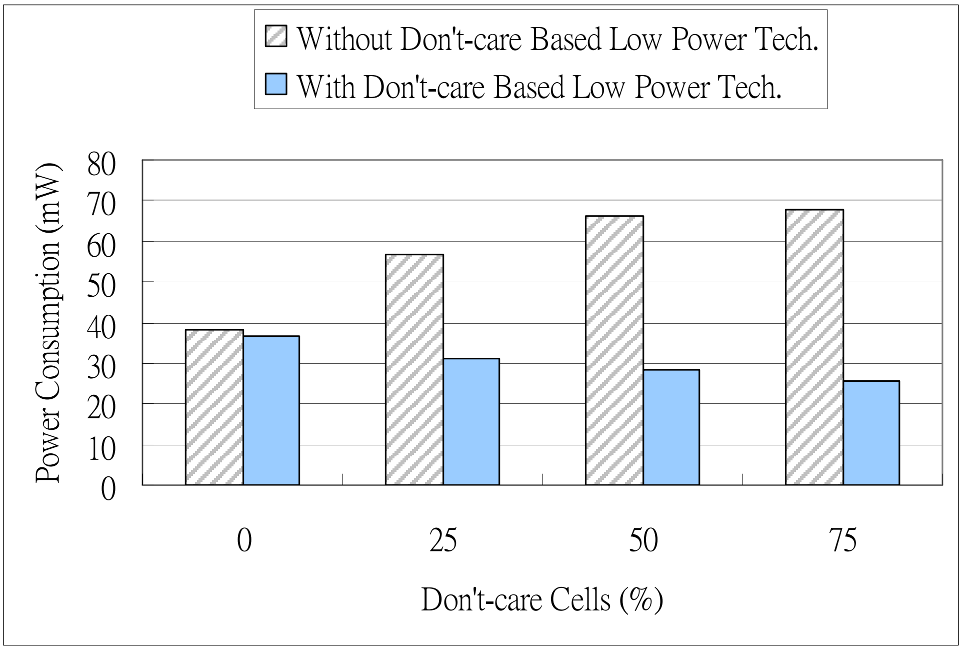


Fig. 5.11 Performance of Type II butterfly match-line scheme with don’t-care based low power technique.

In chapter 4, we know that Type III butterfly match-line scheme is insensitive to don’t-care cells. Without don’t-care based power-gating technique, Type III butterfly match-line scheme saves 60% power consumption compare to Type I. Now, both don’t-care based power-gating technique and don’t-care based hierarchical is applied, Fig. 5.12 shows the simulation result. While the don’t-care cells increase from 0% to 75%, the power consumption is decreasing from 16.937mW to 10.29mW. The simulation result is similar to Type I. This is because of don’t-care based power-gating technique is applied. Since unnecessary precharge has been disabled, all butterfly match-line schemes are insensitive to don't-care cells. Hence, Type I and Type II butterfly match-line scheme consumes almost the same power. The energy metric for Type II with don't-care based low power technique is 0.268fJ/bit/search.

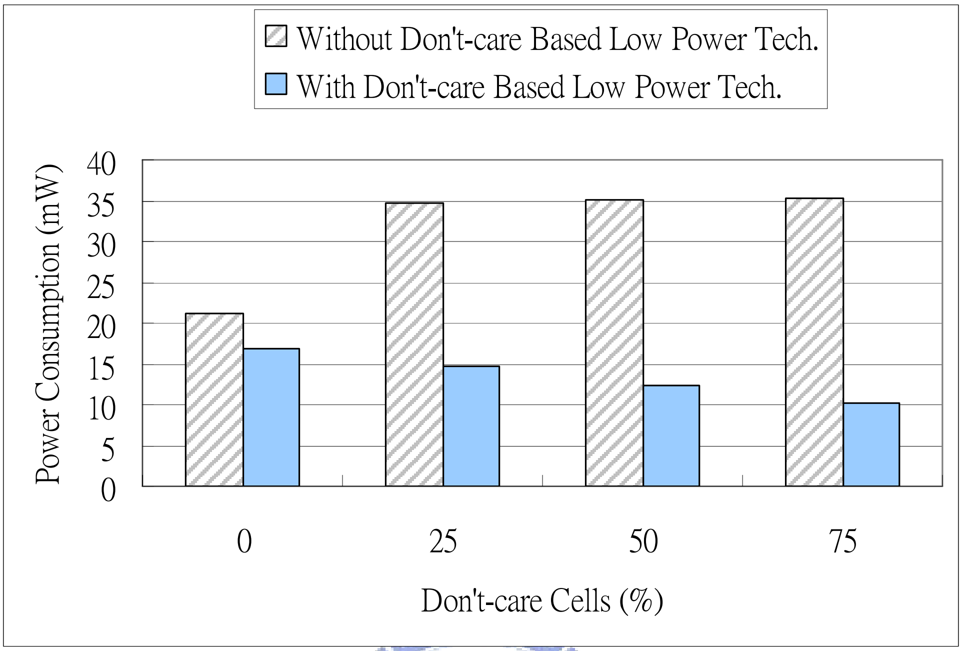


Fig. 5.12 Performance of Type III butterfly match-line scheme with don't-care based low power technique.

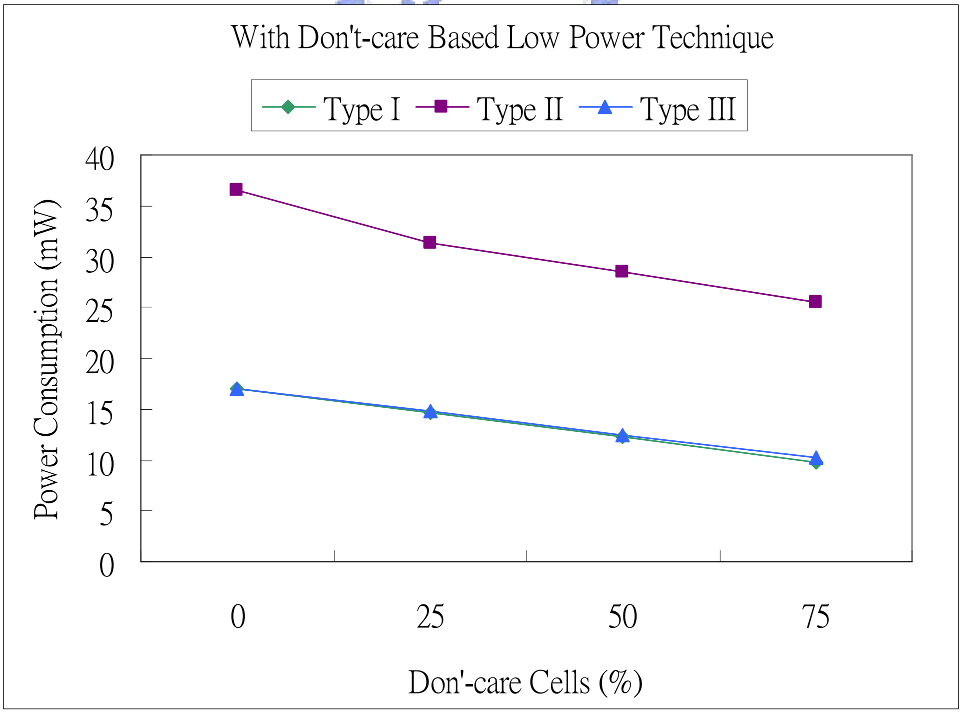


Fig. 5.13 Performance Analysis of proposed TCAM

Fig. 5.13 shows the performance analysis for all types of butterfly match-line scheme. No mater which type, when don’t-care cells increase, power consumption is decreased. Fig. 5.14 shows the search time comparisons. Although Type II butterfly match-line scheme has the shortest search time, however it consumes almost twice power consumption compare to Type I and Type III.

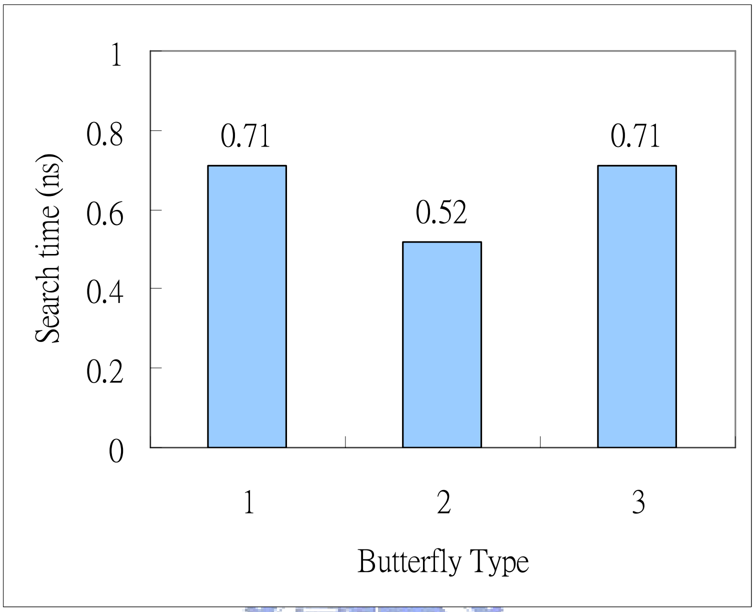


Fig. 5.14 Search time comparisons for three types butterfly match-line scheme

5.5 Layout and the Post-Simulations

A TCAM cell is decomposed into two SRAM cells and comparison circuits and the layout of 1-bit TCAM cell is illustrated in Fig. 5.15. The area of a TCAM cell is 5.73 μm × 4.5 μm = 25.785 μm^2. Besides, the 6 bits TCAM cell and match-line schemes is shown in Fig. 5.16. The size of 6-bit match-line scheme is 5.73 μm × 26.05 μm = 149.2665 μm^2. And the layout of 256x128b proposed TCAMZ array is done and is depicted in Fig. 5.17. Finally, the post-simulation of the proposed TCAM is summarized in Table 5.3.

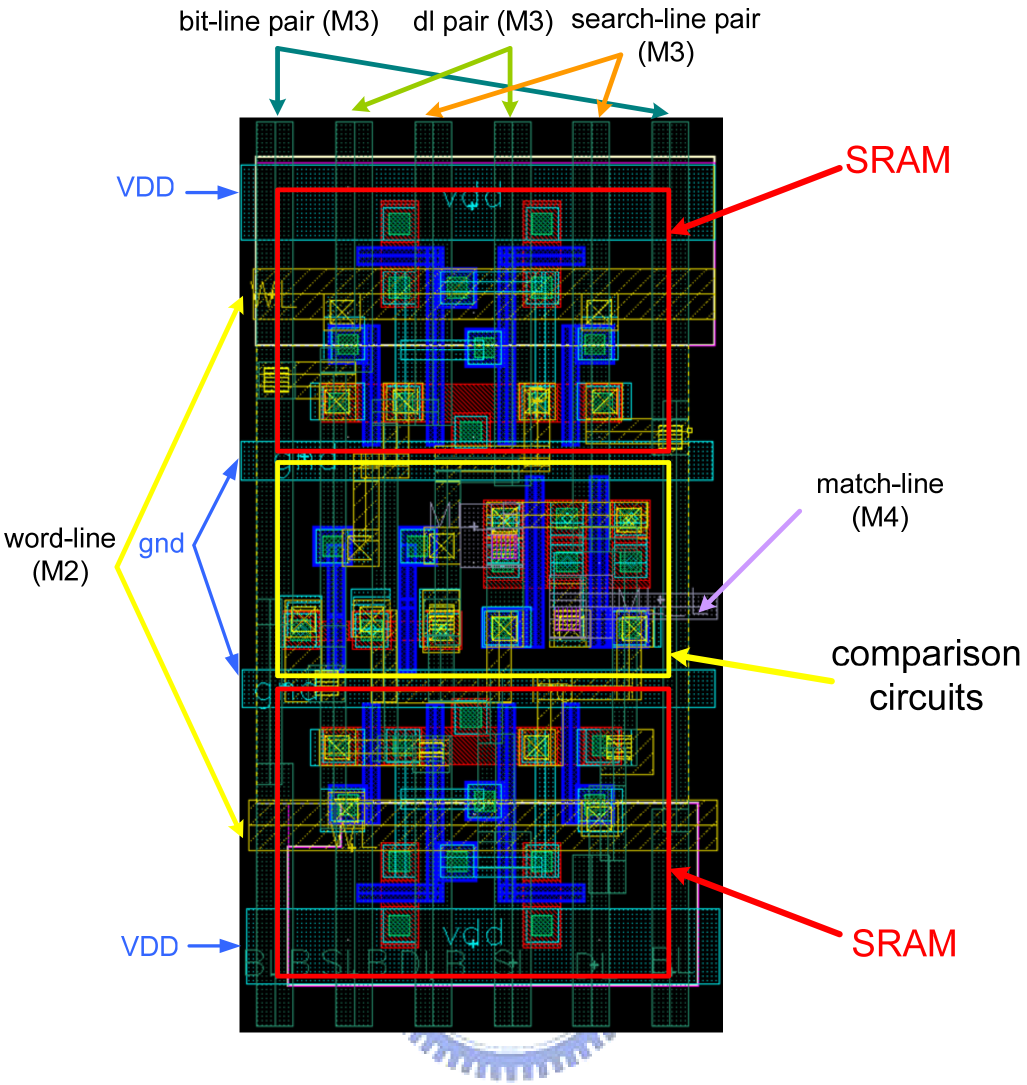


Fig. 5.15 Layout of 1-bit TCAM cell.

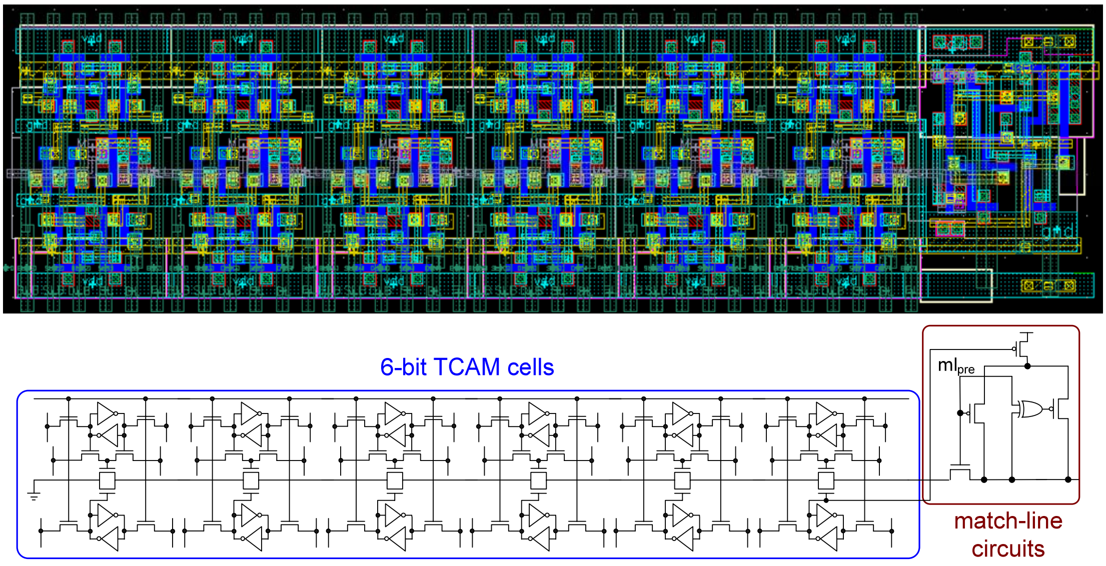


Fig. 5.16 Layout of 6-bit TCAM match-line scheme

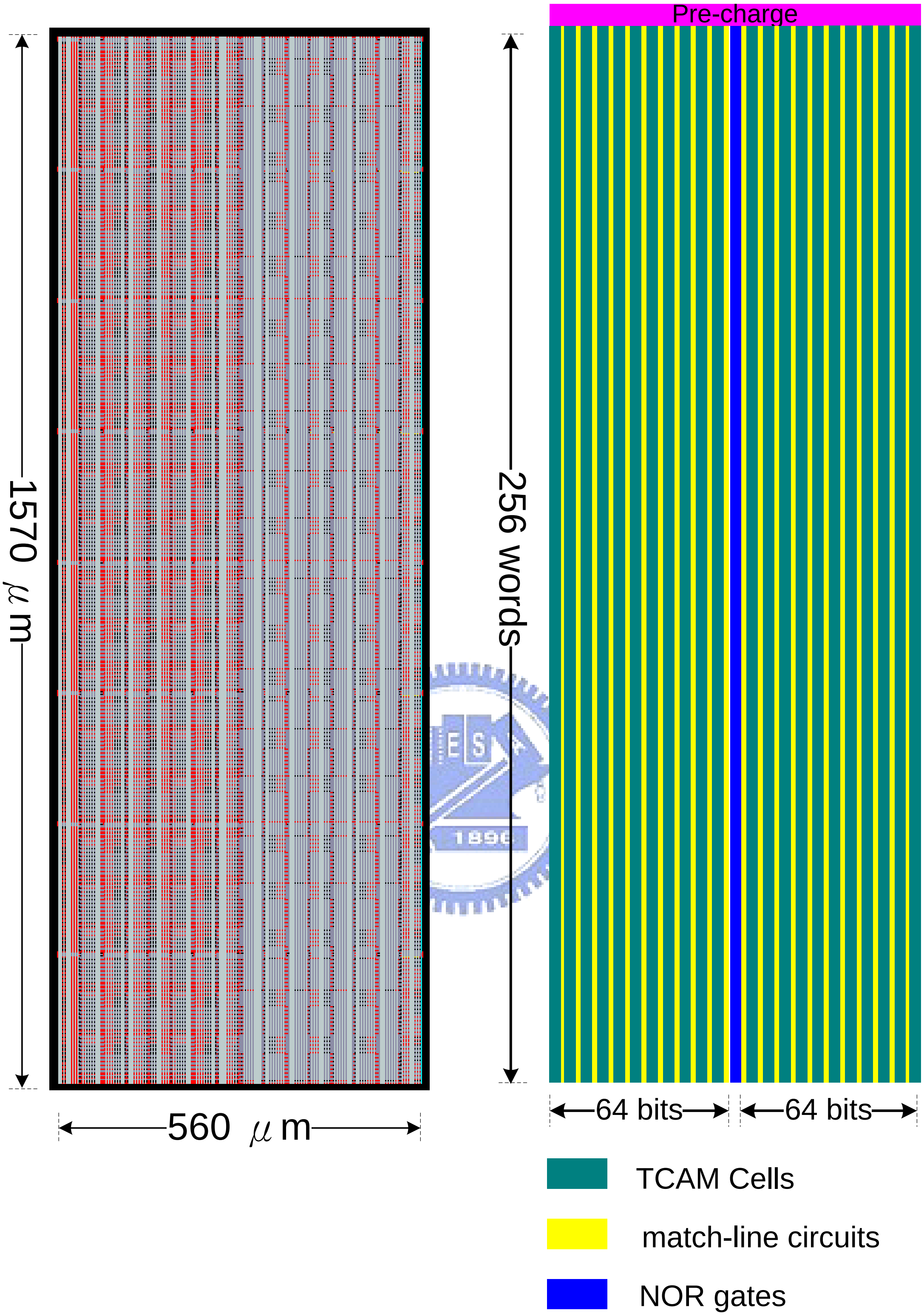


Fig. 5.17 Layout of 256-word x 128-bit proposed TCAM.

Table 5.3 Summary of the proposed TCAM scheme.

TCAM configuration	256 words x 128 bits
Technology	0.13μm CMOS
Supply voltage	1.2V
Clock frequency	500MHz
Search time	0.75ns
Power consumption	11mW
Energy metric	0.26fJ/bit/search

5.6 Summary

The design issues of low-power and high-speed CAM are discussed in this chapter. Moreover, an energy-efficient 256-word x128-bit ternary CAM scheme is realized.

The proposed TCAM scheme is composed of Type III butterfly match-line scheme and don’t-care based low power technique. Applying Type III butterfly match-line scheme reduce not only search time but also power consumption. For IP addressing lookup application, don’t-care based low power technique is proposed. Don’t-care based power-gating technique avoids unnecessary precharge on match-line. Hence, power consumption on match-line of proposed TCAM is insensitive to don’t-care cells. Besides, don’t-care based hierarchical divides search-line into global search-line and local search-line which reduces search-line capacitances. If the TCAM cell is don’t-care, the matching signal is independent of input search data. In this situation, search-line can be disabled to save power. The simulation results shows that increase don’t-care cells make more local search-line disabled. Thus is power consumption on search-line can be reduced.

The layout is implemented with 0.13 μm CMOS technology, and a TCAM cell occupies an area of about 5.73 μm × 4.5 μm. The simulation results show that the search time of proposed TCAM scheme is 0.71ns and power consumption of proposed TCAM scheme is 17mW to 11mW depend on the percentage of don’t-care cells. The TCAM is operates under 500Mhz at 1.2V supply voltage. In addition, the energy metric is 0.26fJ/search.

Chapter 6
Conclusions and Future Work

6.1 Conclusions

Content-addressable memory (CAM) has a single clock cycle throughput making them faster than other hardware- and software-based search systems. However, the speed of a CAM comes at the cost of increased silicon area and power consumption. As CAM applications grow, the larger CAM size demanding, the further exacerbated power problem. Reducing power consumption, without sacrificing speed or area, is the main thread of recent research topic in large-capacity CAMs. The conventional CAM architecture and applications of CAM were described in Chapter 2 in detail. Besides, several low-power and high-speed CAM design techniques were also introduced in this chapter.

IPv6 addresses are 128-bit identifiers for interfaces and sets of interfaces. Each address needs 128 TCAM cells to store data results a long search delay path in network routers for packet forwarding application. AND-type TCAM has more search power saving but longer search time is the price. Thus, reducing the search time by modifying TCAM architecture becomes the inevitable issue. A novel butterfly match-line scheme is proposed in chapter 3. There are 3 types of butterfly match-line scheme. Compare to conventional AND-type match-line scheme, more than half search time is reduced. Furthermore, power consumption on match-line of Type III butterfly match-line scheme is insensitive of don't-care cells. This is the most advantage of Type III in IP addressing lookup application. For 256 words x 128 bits TCAM array, the energy metric is 0.471 for Type III butterfly match-line scheme.

In Chapter 4, two don't-care based low power techniques are proposed. In IP addressing lookup application, TCAM has don't-care state will always generate a matching signal. Consider of a single CAM segment, if all TCAM cells are don't-care.

The floating node must be discharged at evaluation phase in every search cycle. Applying don't-care based power-gating technique eliminates unnecessary precharge saves power consumption on match-line. For Type I butterfly match-line scheme about 160% power consumptions can be saved. Apply to Type II, 62% power consumptions can be saved. When the don't-care based power-gating technique is applied to Type III, 59% power consumption can be saved. If the TCAM cell is don't-care, the matching signal is independent of input search data. In this situation, search-line can be disabled to save power. Don't-care based hierarchical search-line reduces the necessary driven strength of search-line buffer. By applying don't-care based hierarchical search-line, about 30% power consumption saved for both Type I and Type II butterfly match-line scheme. For Type III, more than 50% power consumption can be saved.

For IP addressing lookup application, a 256-word x 128-bit energy-efficient ternary CAM array was implemented in Chapter 5. The proposed TCAM scheme is composed of Type III butterfly match-line scheme and don't-care based low power technique. The simulation results show that the search time of proposed TCAM scheme is 0.71ns and power consumption of proposed TCAM scheme is 17mW to 11mW depend on the percentage of don't-care cells. The TCAM is operates under 500Mhz at 1.2V supply voltage. In addition, the energy metric is 0.26fJ/search.

6.2 Future Work

For the future integrated-circuit (IC) and System-on-Chip (SoC) designs, energy-efficient designs are essentially required. As the technology scale down to deep-submicron and nano-scale eras, both supply voltage (V_{DD}) and threshold voltage (V_t) are reduced for low-power and high-performance designs.

The standby leakage is becoming more serious in deep-submicron and nano-scale technologies. In deep-submicron technologies, sub-threshold leakage is the critical component among all the leakage currents. Sub-threshold leakage increases due to the reduction of threshold voltage with the scaling of technology. In order to compensate the performance degradation of descending supply voltage, threshold voltage is scaled down to satisfy speed requirement. Therefore, the influence of leakage power is becoming significant if threshold voltage keeps on scaling down.

One more leakage source called gate-tunneling leakage (or gate leakage) is becoming important in nano-scale technologies. The increase of gate-tunneling leakage mainly results from the scaling of thickness of gate oxide. Many predictions show that gate leakage has the potential to exceed sub-threshold leakage and dominate the standby leakage current in the future. In recent years, the power-gating techniques are widely used to reduce leakage current of CMOS circuits in standby mode.

Fig. 6.1 (a) depicts a SRAM cell with a gating device between virtual GND and actual GND. As the gating device is turned off, the virtual GND vss0 is floating and charged by leakage current. Thus, the voltage increasing of virtual GND vss0 is obvious. By contrast, Fig. 6.1 (b) shows another modified SRAM cell with gating device and a diode-connected NMOS (data retention device) virtual GND vss1 and actual GND. Through adding NMOS, the voltage of vss1 will be limited. Moreover, the voltage of vss1 is determined by the size of diode-connected NMOS.

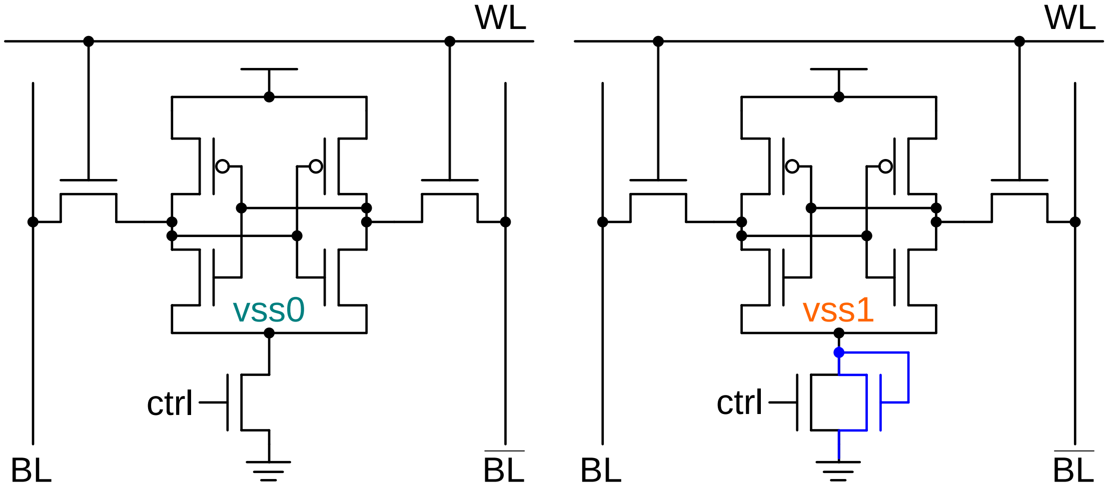


Fig. 6.1 Modified SRAM cell. (a) without diode and (b) with diode.

Bibliography

[1] M. Meribout, T. Ogura, and M. Nakanishi, “On using the CAM concept for parametric curve extraction,” *IEEE Trans Image Process.*, vol. 9, no.12, pp. 2126–2130, Dec. 2000.

[2] M. Nakanishi and T. Ogura, “Real-time CAM-based Hough transform and its performance evaluation,” *Machine Vision Appl.*, vol. 12, no. 2, pp. 59–68, Aug. 2000.

[3] E. Komoto, T. Homma, and T. Nakamura, “A high-speed and compactsize JPEG Huffman decoder using CAM,” in Symp. *VLSI Circuits Dig. Tech. Papers*, 1993, pp. 37–38.

[4] L.-Y. Liu, J.-F.Wang, R.-J.Wang, and J.-Y. Lee, “CAM-based VLSI architectures for dynamic Huffman coding,” *IEEE Trans. Consumer Electron.*, vol. 40, no. 3, pp. 282–289, Aug. 1994.

[5] B. W. Wei, R. Tarver, J.-S. Kim, and K. Ng, “A single chip Lempel-Ziv data compressor,” in Proc. *IEEE Int. Symp. Circuits Syst. (ISCAS),* vol. 3, 1993, pp. 1953–1955.

[6] R.-Y. Yang and C.-Y. Lee, “High-throughput data compressor designs using content addressable memory,” in Proc. *IEEE Int. Symp. Circuits Syst. (ISCAS),* vol. 4, 1994, pp. 147–150.

[7] C.-Y. Lee and R.-Y. Yang, “High-throughput data compressor designs using content addressable memory,” *IEE Proc.—Circuits, Devices and Syst.*, vol. 142, no. 1, pp. 69–73, Feb. 1995.

[8] D. J. Craft, “A fast hardware data compression algorithm and some algorithmic extansions,” *IBM J. Res. Devel.*, vol. 42, no. 6, pp. 733–745, Nov. 1998.

[9] S. Panchanathan and M. Goldberg, “A content-addressable memory architecture for image coding using vector quantization,” *IEEE Trans. Signal Process.*, vol. 39, no. 9, pp. 2066–2078, Sep. 1991.

[10] T. Kohonen, Content-Addressable Memories, 2nd ed. New York: Springer-Verlag, 1987.

[11] L. Chisvin and R. J. Duckworth, “Content-addressable and associative memory: alternatives to the ubiquitous RAM,” *IEEE Computer*, vol. 22, no. 7, pp. 51–64, Jul. 1989.

[12] K. E. Grosspietsch, “Associative processors and memories: a survey,” *IEEE Micro*, vol. 12, no. 3, pp. 12–19, Jun. 1992.

[13] I. N. Robinson, “Pattern-addressable memory,” *IEEE Micro*, vol. 12, no. 3, pp. 20–30, Jun. 1992.

[14] S. Stas, “Associative processing with CAMs,” *Northcon/93 Conf. Record*, 1993, pp. 161–167.

[15] S. Deering, and R. Hinden, “RFC: 1883 Internet Protocol Version 6 (IPv6) Specification,” *Internet Engineering Task Force. (IETF)*, December 1995.

[16] R. Hinden, and S. Deering, “RFC: 3513 Internet Protocol Version 6 (IPv6) Addressing Architecture,” *Internet Engineering Task Force. (IETF)*, April 2003.

[17] C.H. Hua, Wei Hwang, and C.K. Chen, “Noise-Tolerant XOR-Based Conditional Keeper for High Fan-in Dynamic Circuits,” *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, 2005.

[18] J.R. Yuan, C. Svensson, and P. Larsson, “New Domino Logic Precharged by Clock and Data,” *Electronics Letters*, Vol. 29, No. 25, pp. 2188-2189, Dec. 1993.

[19] F. Shafai, K.J. Schultz, G.F.R. Gibson, A.G. Bluschke, D.E. Somppi, “Fully Parallel 30-MHz, 2.5-Mb CAM,” *IEEE Journal of Solid-State Circuits*, Vol. 33, No. 11, pp. 1690-1696, Nov. 1998.

[20] P.F. Lin and J.B. Kuo, “1-V 128-kb Four-Way Set-Associative CMOS Cache Memory Using Wordline-Oriented Tag-Compare (WLOTC) Structure with the Content Addressable-Memory (CAM) 10-transistor Tag Cell,” *IEEE Journal of Solid-State Circuits*, Vol. 36, No. 4, pp.666-675, April 2001.

[21] H. Miyatake and 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, June 2001.

[22] I. Arsovski and A. Sheikholeslami, “A Mismatch-Dependent Power Allocation Technique for Match-Line Sensing in Content-Addressable Memories,” *IEEE Journal of Solid-State Circuits*, Vol. 38, No. 11, Nov. 2003.

[23] I. Arsovski, 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.

[24] A. Roth, D. Foss, R. McKenzie and D. Perry, **“**Advanced Ternary CAM Circuits on 0.13 μm Logic Process Technology**,”** *Proceedings of the IEEE 2004 Custom Integrated Circuits Conference*, pp. 465-468, Oct. 2004.

[25] 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**,”** *Proceedings of the IEEE 2003 Custom Integrated Circuits Conference*, pp. 387-390, Sept. 2003.

[26] H. Noda, K. Inoue, H.J. Mattausch, T. Koide, K. Arimoto, **“**A Cost-Efficient Dynamic Ternary CAM in 130 nm CMOS Technology with Planar Complementary Capacitors and TSR Architecture**,”** *Symposium on VLSI Circuits Digest of Technical Papers*. pp.83-84, June 2003.

[27] V. Lines, A. Ahmed, P. Ma, S. Ma, R. McKenzie, H.S. Kim and C. Mar, **“**66 MHz 2.3 M Ternary Dynamic Content Addressable Memory**,”** Records of the 2000 *IEEE International Workshop on Memory Technology Design and Testing*, pp.101-105, Aug. 2000.

[28] J.G. Delgado-Frias, A. Yu and J. Nyathi, **“**A Dynamic Content Addressable Memory Using a 4-transistor Cell**,”** *Third International Workshop on Design of Mixed-Mode Integrated Circuits and Applications*, pp.110-113, July 1999.

[29] 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, T. Yoshihara, **“**A Cost-Efficient

High-Performance Dynamic TCAM with Pipelined Hierarchical Searching and Shift Redundancy Architecture," *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 1, pp. 245-253, Jan. 2005.

[30] H. Miyatake and 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, June 2001.

[31] T.-B. Pei and C. Zukowski, "VLSI implementation of routing tables: tries and CAMs," *Proc. IEEE INFOCOM*, vol. 2, 1991, pp. 515–524.

[32] , "Putting routing tables in silicon," *IEEE Network Mag.*, vol. 6, no. 1, pp. 42–50, Jan. 1992.

[33] A. J. McAuley and P. Francis, "Fast routing table lookup using CAMs," *Proc. IEEE INFOCOM*, vol. 3, 1993, pp. 1282–1391.

[34] N.-F. Huang, W.-E. Chen, J.-Y. Luo, and J.-M. Chen, "Design of multi-field IPv6 packet classifiers using ternary CAMs," *Proc. IEEE GLOBECOM*, vol. 3, 2001, pp. 1877–1881.

[35] G. Qin, S. Ata, I. Oka, and C. Fujiwara, "Effective bit selection methods for improving performance of packet classifications on IP routers," *Proc. IEEE GLOBECOM*, vol. 2, 2002, pp. 2350–2354.

[36] H. J. Chao, "Next generation routers," *Proc. IEEE,* vol. 90, no. 9, pp. 1518–1558, Sep. 2002.

[37] D. A. Patterson and J. L. Hennessy, "Computer Organization and Design," *Morgan Kaufmann*, 2nd edition.

[38] J. L. Hennessy and D. A. Patterson, "Computer Architecture," *Morgan Kaufmann*, 3rd edition.

[39] R. Panigrahy and S. Sharma, "Sorting and searching using ternary CAMs," in *Proc. Symp. High Performance Interconnects*, 2002, pp. 101–106.

[40] , "Sorting and searching using ternary CAMs," *IEEE Micro*, vol. 23, no. 1, pp. 44–53, Jan.–Feb. 2003.

[41] Http://www.commsdesign.com/main/1999/11/9911feat3.htm:Using Content Addressable Memory for Networking Applications

[42] Http://www.eecg.toronto.edu/~pagiamt/: Content-Addressable Memory (CAM) Primer.

[43] N. Mohan and M. Sachdev, **“**Low Power Dual Matchline Ternary Content Addressable Memory**,”** *Proceedings of the 2004 International Symposium on Circuits and Systems (ISCAS '04),* Vol. 2, pp.633-636, May 2004.

[44] C.A. Zukowski and S.Y. Wang, **“**Use of Selective Precharge for Low-Power on the Match Lines of Content-Addressable Memories**,”** *Proceedings of 1997 International Workshop on Memory Technology, Design and Testing*, pp. 64-68, Aug. 1997.

[45] K. Pagiamtzis and A. Sheikholeslami, **“**Pipelined Match-lines and Hierarchical Search-Lines for Low-Power Content-Addressable Memories**,”** Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, pp. 383-386, Sept. 2003.

[46] N. Mohan and M. Sachdev, **“**A Static Power Reduction Technique for Ternary Content Addressable Memories**,”** *Canadian Conference on Electrical and Computer Engineering*, Vol.2, pp.711-714, May 2004.

[47] M. M. Khellah and M. Elmasry, “Use of charge sharing to reduce energy consumption in wide fan-in gates,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, 1998, pp. 9–12.

[48] I. Arsovski, T. Chandler, and A. Sheikholeslami, “A ternary contentaddressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme,” *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 155–158, Jan. 2003.

[49] C. A. Zukowski and S.-Y. Wang, “Use of selective precharge for lowpower content-addressable memories,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 3, 1997, pp. 1788–1791.

[50] I. Y.-L. Hsiao, D.-H. Wang, and C.-W. Jen, “Power modeling and low-power design of content addressable memories,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 4, 2001, pp. 926–929.

[51] A. Efthymiou and J. D. Garside, “An adaptive serial-parallel CAM architecture for low-power cache blocks,” in *Proc. IEEE Int. Symp. Low Power Electronics and Design (ISLPED)*, 2002, pp. 136–141.

[52] , “A CAM with mixed serial-parallel comparison for use in low energy caches,” *IEEE Trans. VLSI Syst.*, vol. 12, no. 3, pp. 325–329, Mar. 2004.

[53] N. Mohan and M. Sachdev, “Low power dual matchline content addressable memory,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, 2004, pp. 633–636.

[54] K.-H. Cheng, C.-H.Wei, and S.-Y. Jiang, “Static divided word matchline line for low-power content addressable memory design,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, 2004, pp. 629–632.

[55] K. Pagiamtzis and A. Sheikholeslami, “Pipelined match-lines and hierarchical search-lines for low-power content-addressable memories,” in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 2003, pp. 383–386.

[56] , “A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme,” *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1512–1519, Sep. 2004.

[57] J. M. Hyjazie and C. Wang, “An approach for improving the speed of content addressable memories,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 5, 2003, pp. 177–180.

[58] I. Arsovski and A. Sheikholeslami, “A current-saving match-line sensing scheme for content-addressable memories,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2003, pp. 304–305.

[59] , “A mismatch-dependent power allocation technique for matchline sensing in content-addressable memories,” *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1958–1966, Nov. 2003.

[60] H. Noda, K. Inoue, M.Kuroiwa, A. Amo, A. Hachisuka, H. J. Mattausch, T. Koide, S. Soeda, K. Dosaka, and K. Arimoto, “A 143 MHz 1.1W4.5 Mb dynamic TCAM with hierarchical searching and shift redundancy architecture,”

in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2004, pp. 208–209.

[61] B. Gamache, Z. Pfeffer, and S. P. Khatri, **“**A fast ternary CAM design for IP networking applications**,”** *Proceedings of The 12th International Conference on Computer Communications and Networks (ICCCN 2003)*, pp. 434-439, Oct. 2003.

[62] S. Choi, K. Sohn, and H.J. Yoo, **“**A 0.7-fJ/bit/search 2.2-ns Search Yime Hybrid-Type TCAM Architecture**,”** *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 1, pp.254-260, Jan. 2005.

[63] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, **“**Leakage Current Mechanism and Leakage Reduction Technique in Deep-Submicrometer CMOS Circuits**,”** *Proceedings of the IEEE*, vol. 91, No. 2, pp. 305-327, Feb. 2003.

[64] Li Ding and P. Mazumder, **“**On Circuit Techniques to Improve Noise Immunity of CMOS Dynamic Logic**,”** *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 9, pp. 910-925, Sept. 2004.

[65] Li Ding and P. Mazumder, **“**On Circuit Techniques to Improve Noise Immunity of CMOS Dynamic Logi**c,”** *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 9, pp. 910-925, Sept. 2004.

[66] Wei Hwang, R.V. Joshi, and W.H. Henkels, “A 500-MHz, 32-Word x 64-Bit, Eight-Port Self-Resetting CMOS Register File,” *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 1, pp. 56-67, Jan. 1999.

[67] Wei Hwang, G.D. Gristede, Pia Sanda, S.Y. Wang, and D.F. Heidel, “Implementation of a Self-Resetting CMOS 64-Bit Parallel Adder with Enhanced Testability,” *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 8, pp. 1108-1117, Aug. 1999.

[68] M.H. Anis, M.W. Allam, and M.I. Elmasry, “Energy-Efficient Noise-Tolerant Dynamic Styles for Scaled-Down CMOS and MTCMOS Technologies,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 10, No. 2, pp. 71-78, Apr. 2002.

[69] M.W. Allam, M.H. Anis, and M.I. Elmasry, “High-Speed Dynamic Logic

Styles for Scaled-Down CMOS and MTCMOS Technologies," *Proceedings of the 2000 International Symposium on Low Power Electronics and Design (ISLPED '00)*, pp. 155-160, 2000.

[70] S.O. Jung; S.M. Yoo; K.W. Kim, and S.M. Kang, "Skew-Tolerant High-Speed (STHS) Domino Logic," *The 2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001)*, Vol. 4, pp. 154-157, May 2001.

[71] A. Alvandpour, R. Krishnamurthy, K. Soumyanath, and S. Borkar, "A Conditional Keeper Technique for Sub-0.13um Wide Dynamic Gates," *2001 Symposium on VLSI Circuits Digest of Technical Papers*, pp. 29-30, 2001.

[72] A. Alvandpour, R.K. Krishnamurthy, K. Soumyanath, and S.Y. Borkar, "A Sub-130-nm Conditional Keeper Technique," *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 5, pp. 633-638, May 2002.

[73] A. Alvandpour, R. Krishnamurthy, K. Soumyanath, and S. Borkar, "A Low-Leakage Dynamic Multi-Ported Register File in 0.13μm CMOS," *2001 International Symposium on Low Power Electronics and Design*, pp. 68-71, Aug. 2001.

[74] C.M. Lee and E.W. Szeto, "Zipper CMOS," *IEEE Circuits Devices Magazine*, vol. 2, pp. 10-17, May 1986.

[75] J. Pretorius, A. Shubat, and C. Salama, "Charge Redistribution and Noise Margins in Domino CMOS Logic," *IEEE Transactions on Circuits and Systems*, Vol. 33, No. 8, pp. 786-793, Aug 1986.

[76] G.P. D'Souza, "Dynamic Logic Circuit with Reduced Charge Leakage," U.S. Patent 5 483 181, Jan. 1996.

[77] E.B. Schorn, "NMOS Charge-Sharing Prevention Device for Dynamic Logic Circuits," U.S. Patent 5 838 169, Nov. 1998.

[78] Lei Wang and N. R. Shanbhag, "Noise-Tolerant Dynamic Circuit Design," *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems (ISCAS '99)*, Vol. 1, pp. 549-552, May-June 1999.

[79] G. Balamurugan and N.R. Shanbhag, "Energy-Efficient Dynamic Circuit

Design in the Presence of Crosstalk Noise," *Proceedings of 1999 International Symposium on Low Power Electronics and Design*, pp. 24-29, 1999.

[80] G. Balamurugan and N. R. Shanbhag, "A Noise-Tolerant Dynamic Circuit Design Technique," *Proceedings of the IEEE 2000 Custom Integrated Circuits Conference (CICC)*, pp. 425-428, May 2000.

[81] F. Murabayashi, T. Yamauchi, H. Yamada, T. Nishiyama, K. Shimamura, S. Tanaka, T. Hotta, T. Shimizu, and H. Sawamoto, "2.5 V CMOS Circuit Techniques for a 200 MHz Superscalar RISC Processor," IEEE Journal of Solid-State Circuits, Vol. 31, No. 7, pp. 972-980, July 1996.

[82] J.J. Covino, "Dynamic CMOS Circuits with Noise Immunity," U.S. Patent 5 650 733, July 1997.

[83] D.A. Evans, "Noise-Tolerant Dynamic Circuits," U.S. Patent 5 793 228, Aug. 1998.

[84] S. Bobba and I.N. Hajj, "Design of Dynamic Circuits with Enhanced Noise Tolerance," *Proceedings of 20th Annual IEEE International ASIC/SOC Conference*, pp. 54-58, Sept. 1999.

[85] C.H. Hua, Wei Hwang, and C.K. Chen, "Noise-Tolerant XOR-Based Conditional Keeper for High Fan-in Dynamic Circuits," *presented at IEEE International Symposium on Circuits and Systems*, Kobe, Japan, 2005.

[86] H. Mahmoodi-Meimand and K. Roy, "A leakage-Tolerant High Fan-In Dynamic Circuit Design Style," Proceedings of the 2003 IEEE International SOC Conference, pp. 117-120, Sept. 2003.

[87] A. Solomatnikov, D. Somasekhar, K. Roy, and C. K. Koh, "Skewed CMOS: Noise-Immune High-Performance Low-Power Static Circuit Family," Proceedings of 2000 International Conference on Computer Design, pp. 241-246, Sept. 2000.

[88] H. Mahmoodi-Meimand and K. Roy, "Diode-Footed Domino: a Leakage-Tolerant High Fan-In Dynamic Circuit Design Style," IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 51, No. 3, pp.

495-503, March 2004.

[89] J.S. Wang, H.Yu Li, C.C. Chen, and C.W. Yeh, **“**An AND-type Match-line Scheme for Energy-Efficient Content Addressable Memories**,”** *presented at IEEE International Solid-State Circuits Conference*, San Francisco, CA, 2005.

[90] Jinn-Shyan Wang, Chao-Ching Wang, and Chingwei Yeh, “TCAM for IP-Address Lookup Using Tree-style AND-type Match Lines and Segmented Search Lines,” *IEEE International Solid-State Circuits Conference*, pp. 166-167, Feb. 2006

[91] Stallings, Williams. 1996. IPv6: The New Internet Protocol [Document on-line]. available from http://www.ieee.org/ comsoc/stallings.htm; Internet.

Vita

PERSONAL INFORMATION

Birth Date: March. 04, 1982

Birth Place: Taipei, Taiwan, R.O.C.

Address:
Department of Electronics Engineering
National Chiao Tung University
1001 Ta-Hsueh Road
Hsin-chu, Taiwan 30010, R.O.C.

E-Mail Address: ayuinside.ee93g@nctu.edu.tw

EDUCATION

B.S. [2004] Department of Electronical Engineering, Chang Gung University.

M.A. [2006] Institute of Electronics, National Chiao-Tung University.