

# 國立交通大學

電子工程學系 電子研究所

博士論文

應用於多核心系統晶片之節能晶內資料傳輸



研究 生：黃 柏 蒼

指 導 教 授：黃 威 教 授

中 華 民 國 九 十 九 年 十 一 月

應用於多核心系統晶片之節能晶內資料傳輸  
— 以記憶儲存為重心

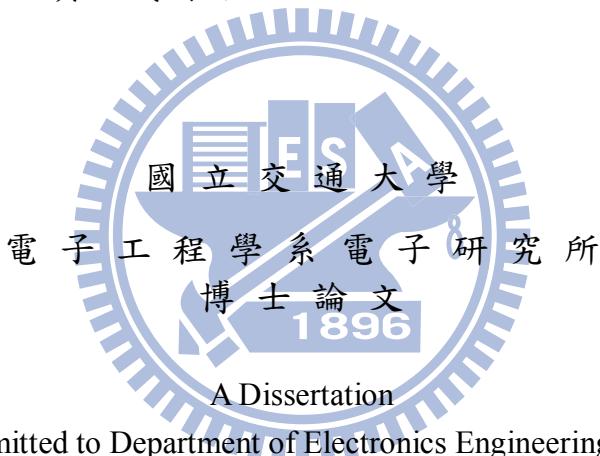
Energy-Efficient Memory-Centric On-Chip Data  
Communication for Multi-Core SoCs

研究 生：黃 柏 蒼

Student : Po-Tsang Huang

指 導 教 授：黃 威 教 授

Advisor : Prof. Wei Hwang



Submitted to Department of Electronics Engineering and  
Institute of Electronics  
College of Electrical and Computer Engineering  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Doctor of Philosophy  
in  
Electronics Engineering

November 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年十一月

# 應用於多核心系統晶片之節能晶內資料傳輸 —以記憶儲存為重心

學生：黃 柏 蒼

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

隨著人們對於無所不在的無線高速資料傳輸多媒體影音需求逐年增加，多核心系統晶片要能有效地提供大量的資料運算、資料傳輸以及資料儲存才有辦法達到未來的系統需求。在本論文中，我們提出了適合於異質多核心系統晶片之晶內資料傳輸平台—以記憶儲存為重心之節能晶內資料傳輸平台，此平台由兩部分構成，分別為以記憶體為重心之晶內連線網路(memory-centric on-chip interconnection network)以及隨記憶體子系統(on-demand memory sub-system)。此平台提供足夠的資料傳輸頻寬、記憶體存取頻寬以及記憶儲容量，並將其應用在無線影像娛樂系統上。

以記憶體為重心之晶內連線網路提供微架構及構成要素給晶內資料傳輸平台，構成要素包括了路由器(router)、連接導線(link wires)及網路介面(network interface)。在連接導線部分，我們提出了具能源效應及可靠度頻道設計，利用自我修正節能編碼及自我補償電壓調整技術，能有效地減少導線間的耦合效應，並且提供頻道錯誤更正及電壓調整機制。在路由器方面，我們設計了兩階層先進先出資料暫存器，藉由提升集中式資料暫存器的使用率及降低線頭阻塞來提升晶內網路效能。為了更進一步的提升網路效能，我們也提出了具堵塞感知之適應性路由演算法，藉由偵測路由器附近的資料傳輸情況，可以避開壅塞的路線。此外，

我們提出了具能源效益的路由表設計給晶內連線網路及IPv6使用，此路由表是由高速且低功率三元定址記憶體(TCAM)陣列構成，其中包含了互斥或邏輯閘的條件式維持器、蝴蝶式比較線連接、階層式搜尋線及電源截段技術。

隨選記憶體子系統負責有效地管理異質多核心系統晶片中多執行程序的記憶體存取，讓系統達到最佳的記憶體使用率。在所提出的隨選記憶體子系統中，主要包含了私有式記憶體管理器以及集中式記憶體管理器。在私有式記憶體管理器中，主要是負責控制第一層快取(L1 cache)存取；此外，我們也提出了一個借取機制，此機制可以動態地分配快取中的記憶體資源給晶內網路封包的暫存使用，以減少處理單元暫停的情況。而在集中式記憶體管理器中，主要是負責管理第二層快取(L2 cache)及外部記憶體之資料存取，利用所提出的適應性快取控制機制，我們可以根據不同處理單元的記憶體存取需求，動態地分配所需記憶體資源。此外，在集中式記憶體管理器中也建構了一個外部記憶體存取介面來有效地存取晶片外的動態記憶體(DRAM)。另外，針對應用於無線影像娛樂系統上的可階式視訊編碼(scalable video coding)，我們提出了跨層間預取資料 (inter-layer pre-fetch) 機制和有效率的動態記憶體位址轉換器來減少快取記憶體的失誤率以及記憶體的能源消耗。

# **Energy-Efficient Memory-Centric On-Chip Data Communication for Multi-Core SoCs**

Student : Po-Tsang Huang      Advisors : Prof. Wei Hwang

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

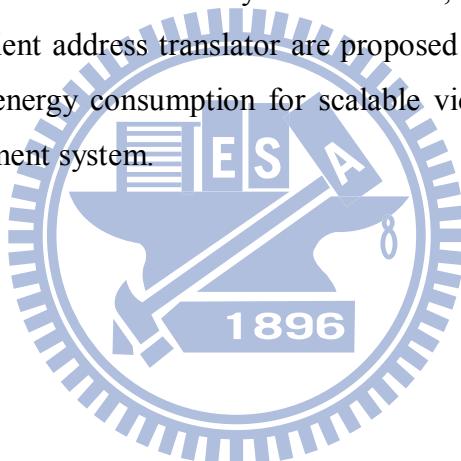
## **ABSTRACT**

With increasing demands on ubiquitous wireless high-data-rate multimedia services, it is critical to have efficient capabilities of the data processing, data communication and data storage to sustain the growth in multi-core system-on-chips (SoCs). In this dissertation, an energy-efficient memory-centric on-chip data communication platform, consisting of a memory-centric on-chip interconnection network (OCIN) and an on-demand memory sub-system, is proposed to provide enough data communication bandwidth, memory bandwidth and memory capacity for heterogeneous multi-core SoCs in wireless video entertainment systems.

The memory-centric OCIN provides the micro-architecture and building blocks, including routers, link wires and network interfaces (NIs), for the on-chip data communication platform. Therefore, an energy-efficient and reliable channel design is presented via a self-corrected green coding scheme and a self-calibrated voltage scaling technique to reduce the coupling effects of link wires and provide the error correction and voltage scaling mechanisms. Consequently, a two-level FIFO buffer router is proposed to enhance the on-chip network performance by increasing the utility of the centralized buffer and reducing the head-of-line blocking problems. Accordingly, an adaptive congestion-aware routing algorithm is also proposed to further increase the performance of mesh networks by detecting the traffic around a routing node. Moreover, energy-efficient routing tables are presented for OCINs and IPv6 applications via the high-performance and low-power ternary content addressable memory (TCAM), which is designed by noise-tolerant XOR-based

conditional keepers, butterfly match-lines, don't-care-based hierarchical search-lines and don't-care-based power gating.

The on-demand memory sub-system is presented to efficiently manage memory accesses of multi tasks in heterogeneous multi-core SoCs via private memory management units (p-MMUs) and a centralized MMU (c-MMU). The p-MMUs control data accesses of L1 caches and dynamically allocate memory resources for network data buffering to reduce the stall of processor elements (PEs) based on the proposed borrowing mechanism. Furthermore, the c-MMU manages centralized on-chip memories (L2 cache) and off-chip memories. For different memory requirements of the PEs in multi tasks, adaptive memory resource allocation is realized to increasing the ovaerall performance using the proposed adaptive cache control. Additionally, an external memory interface (EMI) is develpoed in the c-MMU to access the off-chip DRAM efficiently. Furthermore, an inter-layer pre-fetch mechanism and an efficient address translator are proposed to reduce both the cache miss rate and memory energy consumption for scalable video coding (SVC) in the wireless video entertainment system.



## Acknowledgements

我要感謝我的指導教授黃威教授這幾年對我的指導和鼓勵，在研究過程中提供了很多方向和指引，才讓我的研究可以順利完成。特別感謝老師能讓我同時學習到記憶體設計、記憶體系統、晶內資料傳輸及多媒體與系統整合的領域，使我的研究雖然辛苦但是充滿了挑戰及樂趣。另外要感謝就是跟我同一個團隊的博士班同學和學弟們，使我在研究的過程中激盪出許多想法，也在我研究所的生活中，增添了許多的樂趣。在我的研究過程幫助了我很多也教導了我很多，從他們身上得到很多寶貴的建議。特別感謝 eHome 計畫團隊的黃威教授、黃經堯教授、闕河鳴教授、張錫嘉教授、張添烜教授、許騰尹教授、劉志尉教授、桑梓賢教授、陳宏明教授及黃俊達教授的指導，使我有系統整合的機會與經驗。在團隊工作期間，與各個子計畫的同學也互有往來，感謝各位同學的配合與指教，更提供了很多不同的方向的建議。

最後要感謝我的家人、幼幼社多年來的夥伴及電子系的家族學弟妹們，在研究過程給我的打氣與鼓勵以及關心，並陪伴我渡過許多的挫折及難關，讓我的研究過程能順利完成。

# Contents

## *Chapter 1: Introduction..... 1*

1.1	Motivation.....	2
1.2	Contributions of This Dissertation.....	7
1.2.1	Link Wires.....	8
1.2.2	Routers.....	8
1.2.3	Network Interfaces .....	10
1.2.4	On-Demand Memory Sub-system.....	10
1.3	Organization of This Dissertation .....	11

## *Chapter 2: Survey of On-Chip Data Communication..... 14*

2.1	Why NoC and OCIN? .....	15
2.2	Design Abstraction Levels of NoC .....	18
2.3	Network Topologies of OCINs .....	22
2.4	Flow Control and Switching Technique for OCINs.....	27
2.4.1	Packet-Buffer Flow Control.....	29
2.4.2	Flit-Buffer Flow Control.....	30
2.4.3	Buffer Management and Backpressure.....	32
2.5	Link Wires for OCINs .....	34
2.6	Routers for OCINs .....	39
2.6.1	Routing Algorithm for Link Control .....	40
2.6.2	Switching Matrix (Crossbar) in Routers.....	45
2.6.3	Arbitration Unit in Routers .....	46
2.6.4	Queuing Buffer in Routers.....	46
2.7	Network Interfaces (NIs) for OCINs.....	47
2.8	Power Analysis for NoCs .....	49
2.9	Power Management for NoCs.....	51
2.10	Memory is Network!!!.....	57
2.10.1	Cache Partition Methods.....	59
2.10.2	Data Consistency in Reconfigurable Cache.....	62
2.10.3	Reconfiguration Policy and Detection.....	63
2.11	Summary.....	64

## *Chapter 3: Energy-Efficient and Reliable Channels for OCINs*

.....	66
-------	----

3.1	Background .....	66
3.2	Self-Calibrated Low Power and Energy-Efficient Channel Design ...	70
3.3	Self-Corrected Green (SGC) Coding Scheme .....	72
3.3.1	TriPLICATION Error Correction Stage .....	72
3.3.2	Joint TriPLICATION Bus Power Model.....	76
3.3.3	Green Bus Coding Stage for Crosstalk Avoidance.....	80
3.4	Self-Calibrated Voltage Scaling Technique .....	83
3.4.1.	Crosstalk-Aware Test Error Detection Stage.....	86
3.4.2.	Run-Time Error Detection Stage.....	88
3.5	Simulation Results .....	92
3.6	Summary.....	103

## ***Chapter 4: Two-Level FIFO Buffer Design for Routers..... 105***

4.1	Background.....	105
4.2	Buffer Implementations and Architectures.....	108
4.3	Concept of Two-Level FIFO Buffer Scheme.....	113
4.4	Synchronous Two-Level FIFO Buffer Architecture.....	117
4.4.1	Header Decoder and Routing .....	118
4.4.2	Data-Link Scheduler and Centralized Level-2 FIFO .....	118
4.4.3	Distributed level-1 FIFO .....	121
4.4.4	Arbiter.....	121
4.5	Asynchronous Two-Level FIFO Buffer Architecture.....	122
4.6	Associated Two-Level FIFO Buffer Architecture .....	127
4.7	Simulation Results .....	128
4.7.1	Synchronous Two-Level FIFO Buffer .....	128
4.7.2	Asynchronous Two-Level FIFO Buffer.....	137
4.8	Summary.....	139

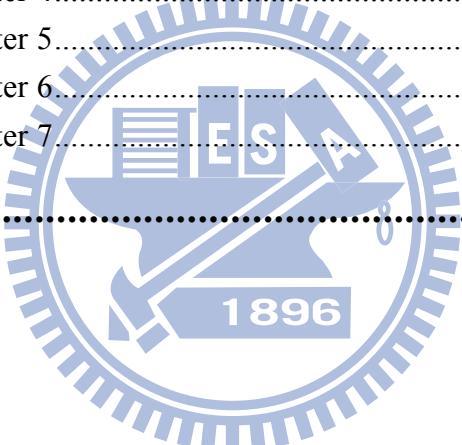
## ***Chapter 5: Adaptive Congestion-Aware Routing Algorithm for***

## ***Mesh On-Chip Interconnection Networks .....* 141**

5.1	Background.....	141
5.2	Congestion-Aware Routing Concept.....	144
5.3	Congestion-Aware Routing Algorithm.....	147
5.3.1	Deadlock Avoidance by the Odd-Even Turn Model.....	147
5.3.2	Score Calculator .....	148
5.3.3	Adaptive Decision Unit .....	150
5.3.4	Buffer Information Collector .....	151

5.4	QoS Guarantee Arbitration Mechanism .....	151
5.5	QoS Guarantee Arbitration Mechanism .....	153
5.6	Summary.....	158
<b>Chapter 6: Energy-efficient Routing Tables for OCINs and IPv6 Applications.....</b>	<b>160</b>	
6.1	Background.....	160
6.2	Routing Tables in OCINs.....	162
6.3	Architecture of TCAM Macro in Network Routers.....	164
6.4	Energy-Efficient Match-Line.....	167
6.4.1	Butterfly Match-line Scheme .....	168
6.4.2	XOR-based Conditional Keeper for Match-Line .....	171
6.5	Don't-Care Based Hierarchy Search-Line Scheme.....	174
6.6	Don't-Care Based Hierarchy Power Gating Techniques .....	177
6.6.1	Multi-Mode Data-Retention Power Gating .....	178
6.6.2	Super Cut-Off Power Gating.....	181
6.7	TCAM Macro Implementation and Measurements .....	186
6.8	Summary.....	195
<b>Chapter 7: On-Demand Memory Sub-system for Multi-Task Wireless Video Entertainment Systems.....</b>	<b>197</b>	
7.1	Background.....	198
7.2	Wireless Video Entertainment System .....	202
7.2.1	Wireless Processing Unit (WPU) .....	205
7.2.2	Medium Access Control (MAC) .....	206
7.2.3	Luby-Transform (LT) Coding .....	208
7.2.4	Scalable Video Coding (SVC).....	210
7.3	Architecture of On-Demand Memory Sub-System.....	212
7.4	Private Memory Management Unit (p-MMU) .....	216
7.4.1	Buffer Borrowing Mechanism .....	217
7.4.2	Borrowing Address Generator.....	218
7.4.3	Buffering Control .....	220
7.4.4	Simulation Results of Buffer Borrowing Mechanism .....	222
7.5	Centralized Memory Management Unit (c-MMU).....	222
7.5.1	Adaptive Cache Controller .....	224
7.5.2	External Memory Interface in the DRAM Controller .....	229

7.6	Pre-fetch Mechanism and Address Translator for SVC .....	235
7.6.1	Inter-Layer Pre-Fetch Scheme .....	235
7.6.2	Address Translator for SVC.....	241
7.7	Analysis of On-Demand Memory Sub-System .....	245
7.8	Summary.....	256
<b><i>Chapter 8: Conclusions and Future Works.....</i></b>		<b>259</b>
8.1	Conclusions.....	259
8.2	Futures Works .....	261
<b><i>Bibliography .....</i></b>		<b>263</b>
	References of Chapter 1.....	263
	References of Chapter 2.....	264
	References of Chapter 3.....	280
	References of Chapter 4.....	283
	References of Chapter 5.....	287
	References of Chapter 6.....	288
	References of Chapter 7.....	292
<b><i>Vita .....</i></b>		<b>296</b>



# List of Figures

---

## *Chapter 1*

Fig. 1.1 A heterogeneous network environment for wireless video entertainment systems.....	1
Fig. 1.2 Vertical exploration of a multi-core system. [1.14].....	4
Fig. 1.3 Comparison between memory bandwidth, computation capability and communication efficiency in multi-core SoCs.....	5
Fig. 1.4 Relative complexity of a video system. [1.15].....	5
Fig. 1.5 Energy-efficient on-chip data communication platform with a memory-centric OCIN and an on-demand memory sub-system.....	6
Fig. 1.6 The contribution matrix of energy-efficient memory-centric on-chip data communication .....	7

## *Chapter 2*

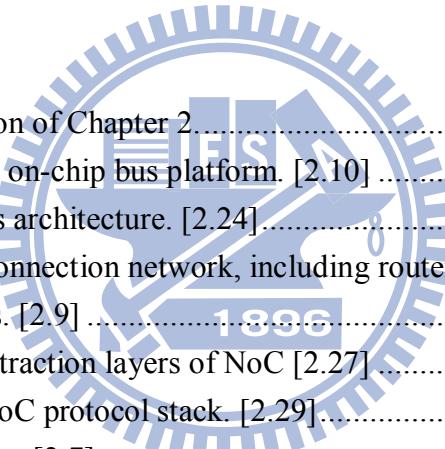


Fig. 2.1 The organization of Chapter 2.....	15
Fig. 2.2 A conventional on-chip bus platform. [2.10] .....	15
Fig. 2.3 Multi-layer bus architecture. [2.24] .....	17
Fig. 2.4 On-chip interconnection network, including routers, link wires and network interfaces. [2.9] .....	18
Fig. 2.5 The design abstraction layers of NoC [2.27] .....	19
Fig. 2.6 The reduced NoC protocol stack. [2.29] .....	19
Fig. 2.7 Data abstraction. [2.7].....	21
Fig. 2.8 NoC research areas versus OSI model based on the flow of data. [2.31]..	21
Fig. 2.9 NoC Research category based on design abstraction layers and flow of data abstraction. [2.31].....	22
Fig. 2.10 Conventional network topologies of OCIN (a) SPIN (b) Mesh (c) Torus (d) Folded tours (e) Octagon (f) Butterfly Fat tree. [2.35].....	23
Fig. 2.11 Xipies Architecture. [2.42] .....	24
Fig. 2.12 A Hierarchical OCIN architecture. [2.43] .....	25
Fig. 2.13 (a) Energy consumption (b) network area according to a number of PEs. [2.48].....	26
Fig. 2.14 Buffered flow control methods can be classified based on their granularity of channel bandwidth allocation and buffer allocation. [2.54]....	28
Fig. 2.15 Virtual channels. [2.8].....	31
Fig. 2.16 A robust self-calibrating transmission scheme for OCIN links. [2.62]....	34
Fig. 2.17 Duplicate-add-parity(DAP) coding (a)Encoder (b)Decoder. [2.71].....	36

Fig. 2.18 Implementation of boundary-shift codes (BSC). [2.72] .....	37
Fig. 2.19 A unified coding framework for link wires. [2.71].....	37
Fig. 2.20 A bi-direction channels to optimize the utilization of link wires. [2.74] .	38
Fig. 2.21 Lookahead-based adaptive voltage scheme. [2.77] .....	39
Fig. 2.22 Micro-architecture of a router for mesh-based OCINs. [2.9].....	40
Fig. 2.23 A category of Routing Algorithms Grouping. [2.91] .....	41
Fig. 2.24 An example of deadlock. [2.92] .....	42
Fig. 2.25 A dynamic routing algorithm for avoiding hot spots. [2.90].....	43
Fig. 2.26 Example for neighbors-on-path algorithm. [2.94].....	44
Fig. 2.27 Crossbar partial activation technique. [2.48] .....	46
Fig. 2.28 Network interfaces connecting cores to the NoC and possible message dependencies in (a) shared-memory and (b) message-passing communication paradigms. [2.121] .....	48
Fig. 2.29 Block diagram of network interface supporting the CTC protocol. [2.121] .....	48
Fig. 2.30 Four levels for power analysis of OCINs.....	50
Fig. 2.31 Power model database development flow in system. [2.131] .....	51
Fig. 2.32 Design Flow of NoC Synthesis. [2.134] .....	51
Fig. 2.33 A methodology for managing power consumption of NOCs via the estimator. [2.136] .....	52
Fig. 2.34 Point-to-point GALS architecture. [2.142] .....	53
Fig. 2.35 GALS systems based on plausible clocking. [2.143] .....	54
Fig. 2.36 GALS systems based on clock gating. [2.143] .....	54
Fig. 2.37 Locally delayed latching (LDL) synchronization. [2.145] .....	55
Fig. 2.38 Bypass architecture of the self-timed ring. [2.154] .....	56
Fig. 2.39 Architecture of GALS NoC unit in a voltage island. [2.157].....	57
Fig. 2.40 Memory is network. [2.161].....	58
Fig. 2.41 Associativity-based partitioning organization for reconfigurable caches. [2.165].....	59
Fig. 2.42 A selective-ways organization and a selective-sets organization. [2.169]60	60
Fig. 2.43 Cache access method in Molecules. [2.171] .....	61
Fig. 2.44 An example of typical CMP cache partitioning. [2.172] .....	62

## ***Chapter 3***

Fig. 3.1 A unified framework for joint crosstalk avoidance code and error correction code. .....	68
Fig. 3.2 Self-calibrated energy-efficient and reliable channels for on-chip interconnection networks with self-corrected green (SCG) coding scheme	

and self-calibrated voltage scaling technique.....	70
Fig. 3.3 Triplication error correction stage of SCG coding scheme.....	72
Fig. 3.4 The corresponding voltages of specific error correction coding versus different un-coded word-error- rate with (a) $k = 8$ (b) $k = 32$ .....	76
Fig. 3.5 (a) Bus Model for 4 bits (b) The approximate bus power model. ....	77
Fig. 3.6 Five transition types for two adjacent wires. ....	78
Fig. 3.7 The design flow of the green bus coding stage. ....	80
Fig. 3.8 (a) The mapping table between 4-bit dataword and 5-bit codeword of the green bus coding stage (b) The two sets and Boolean expression of the green bus coding stage.....	81
Fig. 3.9 The encoder and decoder of green bus coding stage. ....	82
Fig. 3.10 The block diagrams of self-calibrated voltage scaling technique with crosstalk-aware test error detection stage and run-time error detection stage. ....	83
Fig. 3.11 (a) Low swing voltages (b) Low swing driver (c) Level converter. ....	85
Fig. 3.12 The control policy of self-calibrated voltage scaling technique.....	85
Fig. 3.13 MAF based test pattern generator (a) 8 states complete 6 faults test of MAF model (b) Hardware implementation. ....	87
Fig.3.14 Modified double sampling data checking circuit and Wwaveforms (a) error-free (b) delay error (c) glitch error. ....	90
Fig. 3.15 The energy-delay product (EDP) reduction to un-coded code under different values of $\lambda$ with (a) full swing signal (b) the lowest swing signal. ..	95
Fig. 3.16 Simulation environment setup with different number of routers (N) and different lengths (M) of link wires.....	96
Fig. 3.17 Energy reduction under different lengths of link wires and different number of routers.....	97
Fig. 3.18 Energy dissipation of an 8x8 mesh-NoC with different joint CAC and ECC coding schemes. ....	97
Fig. 3.19 The data path delay( $t_d$ ) distributions of rising speed-up, falling speed-up, rising delay, falling delay, normal rising and normal falling cases under (a) high voltage (1.0v) (b) medium voltage (0.85v) (c) low voltage (0.7v).....	99
Fig. 3.20 Voltage levels of the self-calibrated voltage scaling technique under six phases with different noise distributions and timing variations. ....	100
Fig. 3.21 Energy analysis of the self-calibrated energy-efficient and reliable interconnection architecture. ....	101

## ***Chapter 4***

Fig. 4.1 A generic router architecture. ....	106
Fig. 4.2 (a) A router with the centralized buffer (b) A generic router.....	107

Fig. 4.3 Head-of-line blocking problem induced by insufficient buffer.....	108
Fig. 4.4 Different buffer implementation (a) Shift Register (b) Bus-In Shift-Out Register (c) Bus-In Bus-Out Register (d) Bus-In MUX-Out Register. ....	109
Fig. 4.5 Diagram of input buffer, middle buffer and output buffer. ....	111
Fig. 4.6 Concepts of (a) dynamic virtual channel allocation (b) centralized shared buffer .....	112
Fig. 4.7 Data flow of two-level FIFO buffer scheme. ....	114
Fig. 4.8 Concept of the data-link-based FIFO. ....	115
Fig. 4.9 An example of two-level FIFO buffer scheme. ....	116
Fig. 4.10 Two-level FIFO buffer architecture in routers. ....	117
Fig. 4.11 Data-linked based centralized level-2 FIFO and data-link scheduler. ....	119
Fig. 4.12 Implementation of the centralized level-2 FIFO. ....	120
Fig. 4.13 Example of the arbitration policy in deterministic routing algorithms...122	122
Fig. 4.14 Asynchronous two-level FIFO buffer architecture. ....	123
Fig. 4.15 Data-link scheduler and centralized level-2 FIFO for asynchronous two-level FIFO buffer. ....	125
Fig. 4.16 STG specification for the read operation and write operation of the centralized level-2 FIFO. ....	126
Fig. 4.17 Different associations between the distributed level-1 FIFO and the centralized level-2 FIFO for 8 output channels. ....	127
Fig. 4.18 Pipeline stages of the generic router, DSB router and two-level FIFO buffer router. ....	129
Fig. 4.19 Normalized performance versus FIFO sizes with different buffer organizations in (a) low injection load (b) medium injection load (c) high injection load. ....	130
Fig. 4.20 Average latencies of different buffer architectures in (a) uniform patterns (b) hotspot patterns. ....	132
Fig. 4.21 Average latencies of XY, DyXY and adaptive routing algorithms in (a) uniform patterns (b) hotspot patterns. ....	133
Fig. 4.22 Power and area analysis of the different associated two-level FIFO buffers in an 8input/8output router. ....	136
Fig. 4.23 The demonstrated 8x8 DCT system for an asynchronous router.....	137
Fig. 4.24 Latencies, area and energy dissipations for an 8x8 DCT with different buffers. ....	138

## Chapter 5

Fig. 5.1 The architecture of the congestion-aware router. ....	144
Fig. 5.2 The flow diagram of the routing procedure. ....	146

Fig. 5.3 The architecture of adaptive congestion-aware routing algorithm .....	147
Fig. 5.4 An example of the score calculation.....	148
Fig. 5.5 An example of the adaptive decision. ....	150
Fig. 5.6 QoS guarantee arbitrator. ....	152
Fig. 5.7 The average latencies versus the specific switching values under the uniform patterns (a) without hotspots (b) with 6 hotspots. ....	153
Fig. 5.8 Hotspots setting for the 8x8 mesh network.....	154
Fig. 5.9 The average latencies versus the specific switching values under the transpose patterns (a) without hotspots (b) with 6 hotspots. ....	155
Fig. 5.10 The comparisons under the uniform patterns (a) without hotspots (b) with 6 hot spots.....	156
Fig. 5.11 The comparisons under the transpose patterns. ....	157

## ***Chapter 6***

Fig. 6.1 Binary CAM (BCAM) cell and ternary CAM (TCAM) cell. ....	161
Fig. 6.2 A routing table organized with source routes. ....	163
Fig. 6.3 A simplified block diagram of a TCAM macro and packet forwarding by an address-lookup table in network routers. ....	164
Fig. 6.4 PF-CDPD And-type match-line scheme.....	166
Fig. 6.5 Transfer dynamic logic into clock-and-data pre-charge dynamic (CDPD) circuits. ....	166
Fig. 6.6 Butterfly match-line scheme.....	168
Fig. 6.7 AND-type match-line with XOR-based conditional keeper. ....	171
Fig. 6.8 (a) Search time (b) Power consumption versus UNG margin for different keepers. ....	172
Fig. 6.9 Butterfly connection style with XOR-based conditional keeper and don't-care based power gating scheme. ....	174
Fig. 6.10 Packet routing based on longest prefix matching mechanism. ....	175
Fig. 6.11 (a) A simplified architecture (b) Circuit implementation of don't-care based hierarchical search-line scheme. ....	175
Fig. 6.12 Timing analysis of don't-care based hierarchical search-line scheme....	177
Fig. 6.13 The architecture of super cut-off power gating and multi-mode data-retention power gating techniques. ....	178
Fig. 6.14 Multi-mode data-retention power gating technique. ....	179
Fig. 6.15 Relation between noise margin, leakage saving and scale factor. ....	180
Fig. 6.16 Super cut-off power gating technique.....	181
Fig. 6.17 Control circuits for (a) PMOS (b) NMOS cut-off switch. ....	183
Fig. 6.18 (a) VBB generator (b) Voltage doubler for super cut-off power gating..	184

Fig. 6.19 Analysis of voltage generators for different number of match-lines .....	185
Fig. 6.20 Analysis of the search time under different energy-efficient schemes ..	186
Fig. 6.21 Analysis of the energy consumption under different energy-efficient schemes .....	187
Fig. 6.22 Standby power analysis under different power gating techniques .....	188
Fig. 6.23 Search time of one stage under $3\sigma$ process variations. ....	188
Fig. 6.24 Layout view of 1-bit TCAM cell and a TCAM segment with 6-bit TCAM cells .....	189
Fig. 6.25 Layout view of 256x144 TCAM array and test chip micrograph. ....	191
Fig. 6.26 Measurement setup. ....	191
Fig. 6.27 (a) Block diagram of the test chip (b) Test strategy .....	192
Fig. 6.28 Energy consumption under different don't care patterns. ....	193
Fig. 6.29 Network address prefix distribution of IPv6. ....	193
Fig. 6.30 Average standby power with different power gating modes. ....	195

## Chapter 7

Fig. 7.1 Memory hierarchy.....	198
Fig. 7.2 Simplified architecture of a DRAM. ....	200
Fig. 7.3 Configurations of different layers of the proposed memory controller ...	201
Fig. 7.4 Multi-Task wireless video entertainment system. ....	203
Fig. 7.5 Block diagram of the wireless video entertainment system.....	204
Fig. 7.6 Single-FFT Architecture for MIMO Modem. ....	205
Fig. 7.7 Architecture of FD boundary detector. ....	206
Fig. 7.8 MAC Layer Architecture. ....	207
Fig. 7.9 An example of decidable codewords which BP decoding fails to decode.	209
Fig. 7.10 Architecture of an SVC encoder. ....	211
Fig. 7.11 Memory hierarchy in on-demand memory sub-system. ....	213
Fig. 7.12 The data stream of wireless video entertainment systems. ....	214
Fig. 7.13 Architecture of on-demand memory sub-system in eH-II platform. ....	215
Fig. 7.14 Block diagram of a local node.....	216
Fig. 7.15 p-MMU and efficient network interface. ....	217
Fig. 7.16 Buffer borrowing interface between NI and p-MMU.....	218
Fig. 7.17 Borrowing mechanism in p-MMU. ....	218
Fig. 7.18 Architecture of the empty memory block searching. ....	219
Fig. 7.19 Searching flow chart of the borrowing mechanism in p-MMU. ....	220
Fig. 7.20 Block diagrams of borrowing mechanism in network interface. ....	220
Fig. 7.21 Borrowing control policy of the buffering control. ....	221
Fig. 7.22 (a) Execution time under various injection loads and queue sizes (b)	

Transferred packets under various injection loads and queue sizes. ....	222
Fig. 7.23 Block diagram of the c-MMU. ....	223
Fig. 7.24 oncept of the adaptive memory resource allocation. ....	224
Fig. 7.25 Illustration of the memory partition. ....	224
Fig. 7.26 Cache table checking by a bank assignment table. ....	225
Fig. 7.27 Illustration of checking multiple requests. ....	226
Fig. 7.28 Flow chart of adaptive cache control. ....	227
Fig. 7.29 The overall architecture of the c-MMU. ....	228
Fig. 7.30 Connection between the external memory interface (EMI) and DRAM. ....	229
Fig. 7.31 Architecture of the external memory interfaces. ....	231
Fig. 7.32 State diagram of EMI Finite State Machines. ....	231
Fig. 7.33 Bank-miss rescheduling. ....	233
Fig. 7.34 Read/Write rescheduling. ....	233
Fig. 7.35 Row-conflict rescheduling. ....	234
Fig. 7.36 inter-layer motion prediction [7.29]. ....	237
Fig. 7.37 Illustration of inter-layer residual prediction [7.29]. ....	237
Fig. 7.38 Illustration of inter-layer intra prediction [7.29]. ....	238
Fig. 7.39 Data relations of three spatial layers for inter-layer prediction. ....	238
Fig. 7.40 Inter-layer pre-fetch scheme (IPS). ....	239
Fig. 7.41 p-MMU architecture with the pre-fetch command generator. ....	240
Fig. 7.42 Conventional mapping scheme for DRAM. ....	242
Fig. 7.43 DRAM organization in the wireless video entertainment system. ....	242
Fig. 7.44 Video frame arrangement of a GOP. ....	243
Fig. 7.45 Memory mapping for a QCIF frame. ....	244
Fig. 7.46 Task-level parallel organization and flow of data stream. ....	245
Fig. 7.47 Total execution cycles and memory energy consumption. ....	246
Fig. 7.48 Video quality versus channel bit-rate [7.34] ....	247
Fig. 7.49 SVC memory requirements of different scalable layers for a GOP. ....	248
Fig. 7.50 Memory energy consumption for different SVC levels. ....	249
Fig. 7.51 Execution cycles for different SVC levels. ....	250
Fig. 7.52 Various bit-rates in the wireless channel and the corresponding SVC quality levels. ....	250
Fig. 7.53 Total execution cycles and memory energy consumption. ....	250
Fig. 7.54 Miss rate of the L1 cache versus L1 cache size with/without inter-layer pre-fetch scheme. ....	251
Fig. 7.55 Memory accesses of L2 cache with/without inter-layer pre-fetch scheme. ....	251
Fig. 7.56 Memory accesses of DRAM with/without inter-layer pre-fetch scheme. ....	252

Fig. 7.57 Energy measurement of L1 cache in p-MMU.....	252
Fig. 7.58 DRAM row-miss rate.....	253
Fig. 7.59 Number of DRAM row-conflict.....	253
Fig. 7.60 DRAM activate power. ....	254
Fig. 7.61 DRAM bandwidth utilization. ....	254
Fig. 7.62 DRAM energy consumption. ....	255
Fig. 7.63 On-chip cache energy consumption. ....	255
Fig. 7.64 Total memory energy consumption. ....	256

## *Chapter 8*

Fig. 8.1 A femtocell home multimedia center. ....	261
---	-----



# List of Tables

---

## *Chapter 2*

Table 2.1 Related work of reconfigurable caches .....	64
---	----

## *Chapter 3*

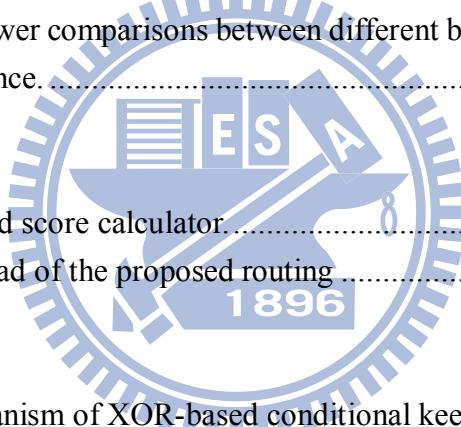
Table 3.1 Comparisons between green bus coding and increasing wire spacing....	82
Table 3.2 Different combinations of joint coding schemes .....	93
Table 3.3 Summaries of different joint coding schemes for 8-bit link wires.....	93
Table 3.4 Summaries of SCG coding and self-calibrated voltage scaling.....	103

## *Chapter 4*

Table 4.1 Area and power comparisons between different buffer architectures in the same buffer size. ....	134
Table 4.2 Area and power comparisons between different buffer architectures with similar performance. ....	137

## *Chapter 5*

Table 5.1 The modified score calculator.....	149
Table 5.2 Area overhead of the proposed routing.....	158



## *Chapter 6*

Table 6.1 Control organism of XOR-based conditional keeper. ....	171
Table 6.2 The corresponding control signals under different operations.....	182
Table 6.3 Features Summary and Comparisons. ....	194

## *Chapter 7*

Table 7.1 Cost-performance for various memory technologies.....	199
Table 7.2 System Specification (Receiver) .....	204
Table 7.3 Micron DDR3 configurations .....	230
Table 7.4 Simulation environment. ....	235
Table 7.5 Memory requirement assumption and corresponding bank assignment for c-MMU .....	246
Table 7.6 Summaries of SVC parameters.....	248
Table 7.7 c-MMU bank assignment for wireless video entertainment systems....	249

## *Chapter 8*

Table 8.1 The summary of this dissertation .....	260
--	-----



# ***Chapter 1: Introduction***

With the advancement of the wireless communication and multimedia techniques, great amount of digital electronic devices are developed in human life. These modern electronic products provide a convenient entertainment environment. Fig. 1.1 presents a heterogeneous network environment in our life that provides wireless video entertainment systems anytime and anywhere. In recent years, merging different networks, electronic appliances and media devices into a heterogeneous integrated platform becomes a trend that provides a friendly and energy-efficient digital environment for people enjoying their life [1.1]. Therefore, heterogeneous multi-core system-on-Chip (SoC) designs provide an integrated solution for merging processor elements (PEs) or intellectual properties (IPs) in communications, multimedia and consumer electronics. A successful SoC design depends on the availability of methodologies that allow designers to meet two major challenges—the miniaturization of interconnecting features, and the requirement of memory capacity/memory bandwidth.



Fig. 1.1 A heterogeneous network environment for wireless video entertainment systems.

## 1.1 Motivation

Modern multi-core SoC designs face a number of problems caused by the data communication among PEs and memory accesses. In addition to shrinking processing technologies, the ratio of interconnection delay to gate delay will increase in advanced technologies [1.2], indicating that on-chip interconnection architectures will dominate performance in future SoC designs. Furthermore, in current multi-core SoC designs, reducing power consumption is the primary challenge for advanced technologies. Thus, using an on-chip bus to create a platform is a solution for multi-core SoC designs. This on-chip bus platform provides interfaces between multiple processor elements and verification environments [1.3], [1.4]. However, the requirements for on-chip communication bandwidth and PEs are growing continually beyond that which can accommodate standard on-chip buses. Moreover, advanced SoC designs using nano-scale technologies face a number of challenges. First, the shared bus architecture will become a development-critical factor for integration with an increasing number of processor elements. Existing bus architectures and techniques are not scalable, and cannot meet the specific requirements associated with low power and high performance [1.5]. Second, the interconnect delay across the chip exceeds the average clock period of IP blocks. Thus, the ratio of global interconnect delay to average clock period will continue increasing according to the International Technology Roadmap for Semiconductors (ITRS) [1.2]. Third, advanced technologies increase the coupling effect for interconnects, such as capacitive and inductive crosstalk noise. The increasing coupling effect aggravates power-delay metrics and degrades signal integrity [1.6]. Fourth, system design and performance are limited by the complexity of the interconnection between the different modules and blocks with a single clock design [1.7].

As design complexity of multi-core SoC continues to increase, a global approach is needed to effectively transport and manage on-chip communication traffic, and to optimize wire efficiency. Therefore, process-independent network-on-chip (NoC) has been considered an effective solution for integrating a multi-core system. NoC was investigated for dealing with the challenges of on-chip data communication caused by the increasing scale of next generation SoC designs [1.8], [1.9]. The most important characteristics of NoC can be considered as a packet switched approach [1.10] and a flexible and user-defined topology [1.11]. Furthermore, on-chip interconnection networks (OCINs) provide the micro-architecture and the building blocks for NoCs, including network interfaces, routers and link wires [1.12], [1.13]. The generic OCIN is based on a scalable network, which considers all requirements associated with on-chip data communication and traffic. OCINs have a few beneficial characteristics, namely, low communication latency, low energy consumption constraints, and design-time specialization. The motivation in establishing OCINs is to achieve performance using a system communication perspective.

Multi-core SoCs have become a major trend of architecture in modern data computing systems. The multiple PEs are integrated on a single chip or package to exploit the parallelism of applications and achieve superior performance as well as energy efficiency. Because these systems are highly integrated, their designs and trade-offs are tightly coupled; a single design decision can impose significant impact on multiple design layers. Thus, for optimal results, designers have to consider multiple design layers (vertical exploration) and multiple architecture options (horizontal exploration) when mapping an application to an underlying multi-core system as shown in Fig. 1.2 [1.14]. In multi-core SoC designs, the processes of data streaming can be divided into three parts, including data computation, data storage

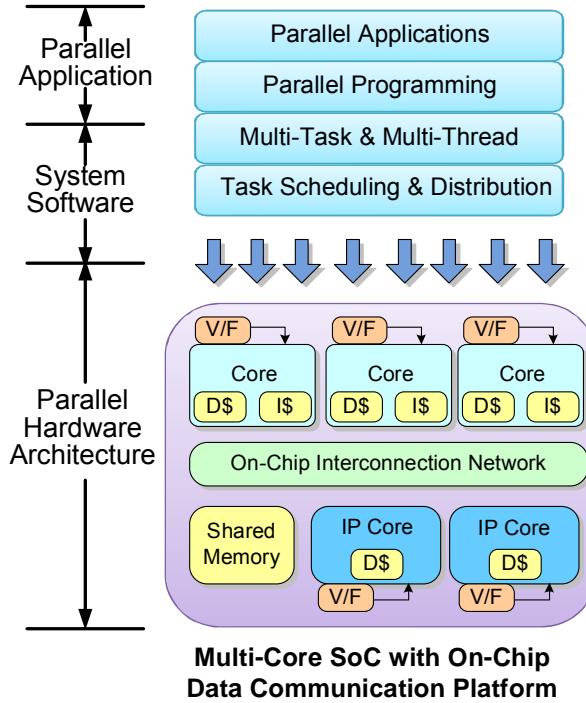


Fig. 1.2 Vertical exploration of a multi-core system. [1.14]

and data communication. With the increasing PEs in multi-core SoCs, the capability of data computation increases rapidly to satisfy the increasing demands of mobile multimedia services [1.5]. Additionally, multi-task processing is also provided via multi-core SoCs based on parallel programming and task scheduling as shown in Fig. 1.2. According to the task scheduling, the on-chip data communication platform builds the backbone of the parallel hardware architecture and provides data communication and data storage via the OCIN and memory sub-system, respectively. Furthermore, memory accesses and on-chip data communication dominate the overall performance of multi-core SoCs as shown in Fig. 1.3. Therefore, the development of memory sub-system in multi-core systems will affect the overall performance dramatically. Moreover, the relative complexity of a video system increases year by year as presented in Fig. 1.4 that indicates great amount of memory capacity and memory bandwidth are required for high quality or multiple scalable level video processing [1.15]. Therefore, the memory sub-system should provide large memory

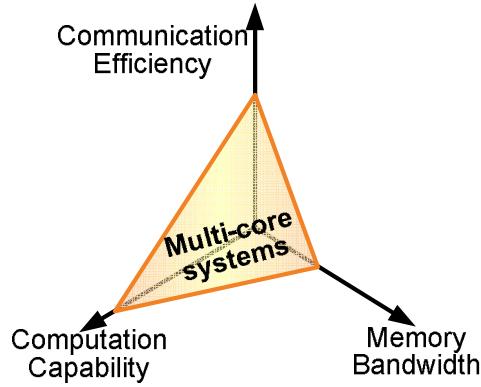


Fig. 1.3 Comparison between memory bandwidth, computation capability and communication efficiency in multi-core SoCs.

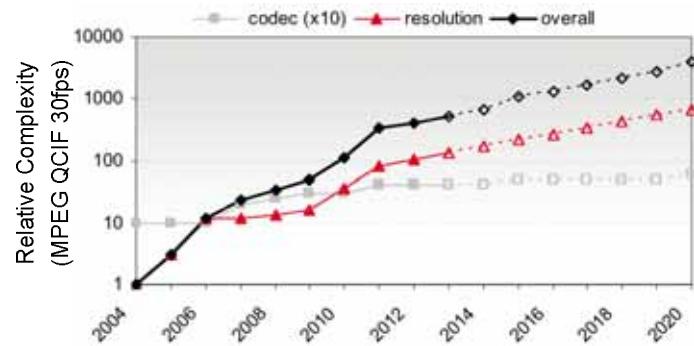


Fig. 1.4 Relative complexity of a video system. [1.15]

space and high memory-access bandwidth for satisfying the video real-time requirement. Accordingly, large amounts of high speed and low power memories are indispensable for multi-task and multi-system emerging. These memories should be able to support diverse memory requirement of different PEs in a wireless video entertainment system using a memory sub-system.

When process technologies shrink to nano-scale, the ever-increasing on-chip integrations in recent years have led to a dramatic increase in system performance and system scale. Unfortunately, as performance and area are improved, power dissipation and heat density are substantially increased [1.16]. Accordingly, power dissipation in multi-core SoC designs has become a critical design issue. In multi-core SoC implementations of mobile systems, especially for handheld audio and video applications, low power considerations dominate the overall performance since the

battery life and geometry of mobile systems are limited [1.17], [1.18]. The demand for reliability design will require designers to find new technologies and circuit to ensure high performance and long operating lifetimes, owing to the high cost of packaging and cooling in nano-scale CMOS technologies. Therefore, energy-efficient circuitry becomes one of the critical issues in multi-core SoC designs.

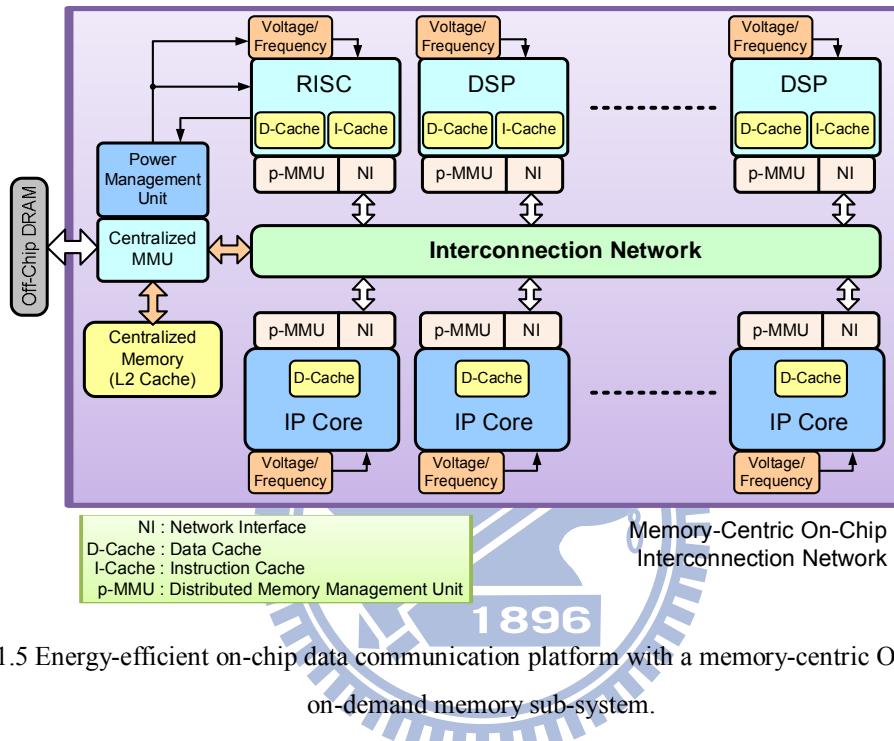


Fig. 1.5 Energy-efficient on-chip data communication platform with a memory-centric OCIN and an on-demand memory sub-system.

Based on the above crucial issues in multi-core SoCs, including the energy bound and the increasing requirements of data communication and data storage, an energy-efficient on-chip data communication platform is proposed in this dissertation as shown in Fig. 1.5. This on-chip data communication platform consists of a memory-centric OCIN and an on-demand memory sub-system. The memory-centric OCIN provides building blocks with on-demand memory sub-systems, including energy-efficient and reliable channels, congestion-aware routing algorithm, energy-efficient routing table, two-level FIFO buffer and buffer-efficient NIs. In addition, the on-demand memory sub-system provides high bandwidth and low power memory accesses for multi-core SoCs via a centralized memory management unit

(c-MMU) and private memory management units (p-MMUs). The on-demand memory sub-system can support variety memory resources for different PEs based on the memory behaviors. Moreover, when decoding the video frames, memory access characteristics of video decoders are generally regular and repeat. Therefore, the on-demand memory sub-system can improve the decoding performance via efficient memory management.

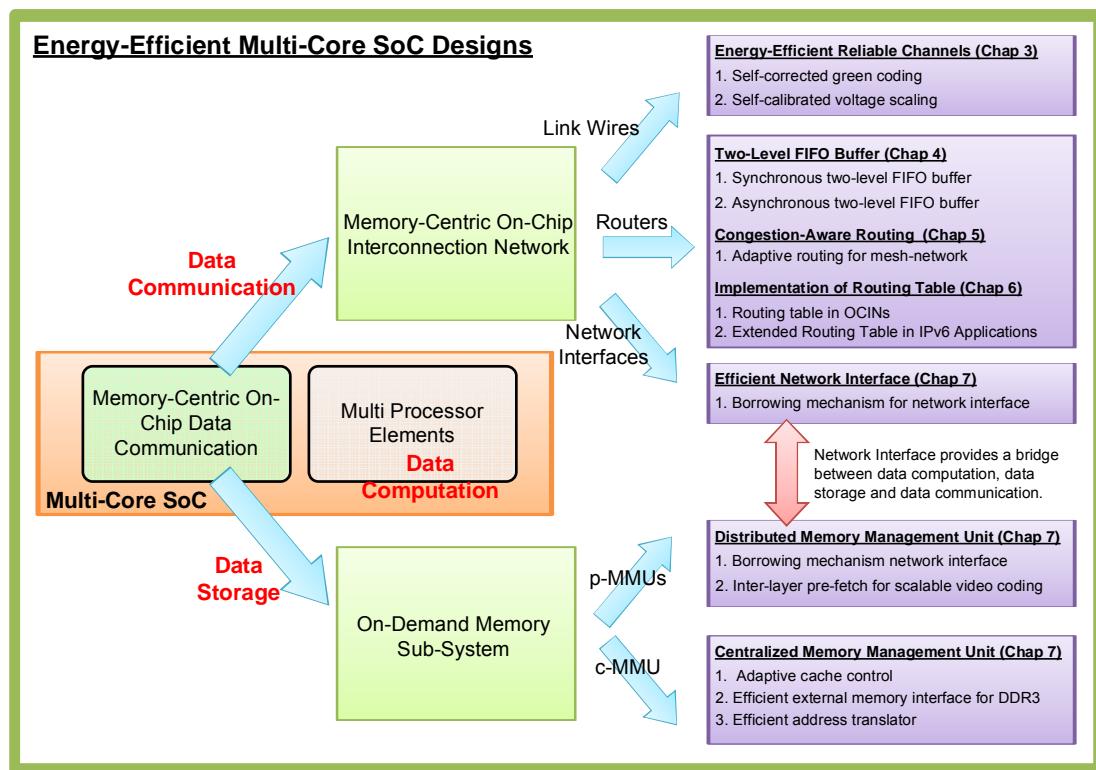


Fig. 1.6 The contribution matrix of energy-efficient memory-centric on-chip data communication

## 1.2 Contributions of This Dissertation

In this dissertation, an energy-efficient memory-centric on-chip data communication platform is proposed to deal with the increasing data communication and data storage for heterogeneous multi-core SoC designs. Fig. 1.6 presents the contribution matrix of energy-efficient memory-centric on-chip data communication which consists of a memory-centric OCIN and an on-demand memory sub-system. The memory-centric OCIN provides the micro-architecture for data communication

based on the building blocks, including link wires, routers and NIIs. In this dissertation, all building blocks are analyzed and developed to realize energy-efficient multi-core SoCs. Additionally, the on-demand memory sub-system enhances memory bandwidth and reduces the total execution time of the whole system via the centralized MMU and private MMUs. Moreover, the NI provides a bridge between the on-demand memory sub-system, memory-centric OCIN and heterogeneous PEs. The contributions of each block are described as follows.

### 1.2.1 Link Wires

For link wires, a novel self-calibrated energy-efficient and reliable channel design is proposed for OCINs. The proposed channels reduce the energy consumption while maintaining reliability. The channels are developed using the self-calibrated voltage scaling technique with the self-corrected green (SCG) coding scheme. The SCG coding is a joint bus and error correction coding scheme that provides a reliable mechanism for channels. In addition, it achieves a significant reduction in energy consumption via a joint triplication bus power model for crosstalk avoidance. Based on SCG coding scheme, the proposed self-calibrated voltage scaling technique adjusts voltage swing for energy reduction. Furthermore, this technique tolerates timing variations.

### 1.2.2 Routers

Routers are the essential components of OCINs. The router architecture depends on the topology and flow control of OCINs. A generic router architecture consists of a set of input buffers, an interconnect matrix, a set of output buffers and control circuitries, including a routing controller, an arbiter and an error detector. In this thesis, a data-link two-level FIFO (first-in first-out) buffer architecture with the centralized

shared buffer is proposed in this paper. The proposed two-level FIFO buffer architecture has a shared buffer mechanism allowing the output channels to share the centralized FIFO with sufficient buffer space. Additionally, the proposed architecture reduces the area and power consumption to achieve the same performance.

In addition to the proposed two-level FIFO buffer, an adaptive congestion-aware routing algorithm with a quality-of-service guarantee arbitration mechanism is proposed for mesh OCINs. Depending on the traffic around the routed node, the proposed routing algorithm provides not only minimum paths but also non-minimum paths for routing packets. Both minimum and non-minimum paths are based on the odd-even turn model to avoid deadlock and livelock problems. The decision of the minimum paths or non-minimum paths depends on the utilities of buffers in neighbor nodes and the specific switching value. In this adaptive algorithm, the congestion conditions and distributed hotspots will be avoided. It has the advantages of getting higher performance and also reducing the latency.

The implementation of routing tables is also proposed via content addressable memories (CAM). Moreover, the implementation of routing tables is extended for IPv6 network routers, which is the next generation of network routers, using ternary content addressable memories (TCAM). As routing tables become larger, energy consumption and leakage current become increasingly important issues in the design of TCAM in nano-scale technologies. Therefore, a novel energy-efficient TCAM macro design is proposed for IPv6 applications. The proposed TCAM employs the concept of architecture and circuit co-design. To achieve an energy-efficient TCAM architecture, a butterfly match-line scheme and a hierarchy search-line scheme are developed to reduce significantly both the search time and power consumption. The match-lines are also implemented using noise-tolerant XOR-based conditional

keepers to reduce not only the search time but also the power consumption. To reduce the increasing leakage power in advanced technologies, the proposed TCAM design utilizes two power gating techniques, namely super cut-off power gating and multi-mode data-retention power gating.

### 1.2.3 Network Interfaces

NIs, one of the building blocks in OCINs, is a major factor in the performance. In this dissertation, an efficient NI is proposed for the memory-centric OCIN to reduce the data blocking by a borrowing mechanism. By considering the borrowed memory blocks and p-MMU, the size of the output queue in NI can be dynamically scheduled. Additionally, the p-MMU can dynamically allocate the memory resources for buffering the blocking network data. Therefore, the proposed efficient NI can increase the performance of the memory-centric OCIN.

### 1.2.4 On-Demand Memory Sub-system

In this dissertation, a memory-centric on-chip data communication platform is presented for merging heterogeneous PEs, and applied to wireless video entertainment systems. In this platform, on-demand memory sub-system is developed for dynamically allocating memory resources and efficiently managing memory accesses. The contributions of on-demand memory sub-system are described as follows.

#### A. Buffer borrowing mechanism for NIs

In order to reduce the stall of PEs caused by network data blocking, a novel buffer borrowing mechanism is proposed to borrow the memory resources for buffering the blocking packets.

#### B. Adaptive cache control

For multi-task applications, different processor elements (PEs) may have different memory requirements during runtime. Therefore, the proposed c-MMU can support memory resource re-allocation by adaptive cache control scheme. Accordingly, the memory utilization of the system can be improved.

### **C. External Memory Interface for DDR3 DRAM**

DDR3 DRAM devices are utilized for supporting huge data storage recently. Therefore, an efficient external memory interface (EMI) is also designed to reschedule read/write commands for DDR3 DRAM that reduces both execution time and energy consumption.

### **D. Inter-Layer Pre-Fetch Scheme for Scalable Video Coding**

For wireless video entertainment systems, a scalable video coding (SVC) is utilized to adapt variations of wireless channels. Based on the data stream of SVC, an inter-layer pre-fetch scheme (IPS) is proposed to reduce the miss rate during frame decoding of SVC.

### **E. Efficient Address Translator (AT) for SVC**

The SVC data allocation in DDR3 DRAM is proposed using an efficient address translator. The translated addresses can improve the DRAM access efficiency while processing SVC.

## **1.3 Organization of This Dissertation**

The organization of this dissertation is depicted as follows. The related works of on-chip data communication are introduced in Chapter 2. In this chapter, the concept of on-chip data communication and previous works of the NoC/ OCIN are described. After presenting the related works of on-chip data communication, Chapter 3 presents

the self-calibrated energy-efficient and reliable channel design for OCINs using a self-calibrated voltage scaling technique with a SCG coding scheme. In the beginning of this chapter, previous reliable and low power coding schemes are analyzed. Then, the self-calibrated low power coding and voltage scaling channels are presented in the following sections.

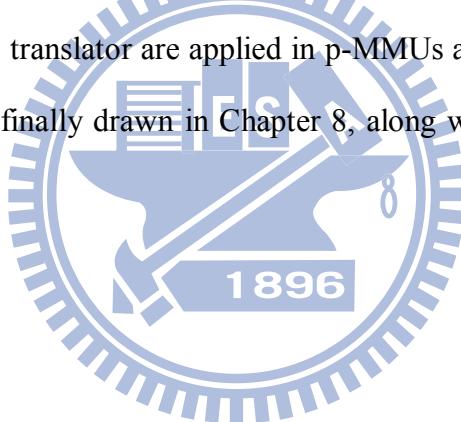
Chapter 4 presents the synchronous and asynchronous two-level FIFO buffers in routers for OCINs. The proposed two-level FIFO buffer architecture has a shared buffer mechanism allowing the output channels to share the centralized FIFO with sufficient buffer space. Different buffer architectures and different circuit implementations are analyzed and compared in the beginning of this chapter. Then, the concept of the proposed two-level FIFO buffer architecture is presented. The next section describes the behavior and circuit implementation of the data-link two-level FIFO buffer for the router. Consequently, the asynchronous and associated two-level FIFO buffer architectures are described in the following sections.

An adaptive congestion-aware routing algorithm is described in Chapter 5. In the first section of this chapter, the related works of routing algorithms are introduced and compared. The concept of the proposed routing algorithm for a router is presented in the following section. Then, the detail of the proposed adaptive congestion-aware routing algorithm and its implementation are both described. In addition, the quality-of-service arbitration mechanism is also be presented in next section.

And then, the implementation of routing tables in OCINs is presented in the first section of Chapter 6. Moreover, the implementation of routing tables is extended for network routers in IPv6 applications via a TCAM macro. In this chapter, the overall architecture of the TCAM macro design is introduced. Then, the following section introduces the proposed energy-efficient match-line schemes, which involve the

butterfly match-line and XOR-based conditional keeper. Next, the proposed don't-care-based hierarchy search-line scheme will be presented. Furthermore, the next section elucidates two power gating techniques for reducing leakage current.

Subsequently, Chapter 7 presents the design of the on-demand memory sub-system, including p-MMUs and a c-MMU. Buffer borrowing mechanism in p-MMUs and adaptive cache scheme in the c-MMU are proposed for optimizing the memory resources utilization dynamically. Additionally, for accessing the external memory, an efficient external memory interface is presented. Subsequently, a pre-fetch and DRAM data allocation schemes are described to improve the memory energy efficiency in wireless video entertainment systems. Therefore, a pre-fetch command generator and an address translator are applied in p-MMUs and c-MMU, respectively. Finally, conclusions are finally drawn in Chapter 8, along with recommendations for future research.



## ***Chapter 2:*** ***Survey of On-Chip Data Communication***

With development of System-on-Chip (SoC) and multimedia communication technologies, a great amount of data computing requirement increases rapidly. In addition, the communication bandwidth requirement between processor elements (PEs) and the memory bandwidth requirement are also increasing to maintain the system performance. Therefore, the aggregate communication bandwidth between the processing cores is in the GBytes/s range for many video applications. In the future, with the integration of many applications onto a single device and with increased processing speed of cores, the bandwidth demands will scale up to much larger values. Multi-core SoC architectures are emerging as appealing solutions for embedded multimedia applications [2.1]-[2.5]. In general, multi-core SoCs are composed of core processors, memories and some application-specific cores. Additionally, data communication among PEs is provided by advanced interconnect fabrics, such as high performance and efficient networks-on-chip (NoCs) [2.6]. NoC was investigated for dealing with the challenges of on-chip data communication caused by the increasing scale of next generation SoC designs. Furthermore, on-chip interconnection networks (OCINs) provide the micro-architecture and the building blocks for NoCs, including network interfaces (NIs), routers and link wires [2.7], [2.8]. In OCINs, PEs (including memory modules) communicate by sending packet to one another over the network instead of by sending wires over ad-hoc wiring structures [2.9]. In this chapter, the related works of on-chip data communication are given, including NoCs, OCINs and memory sub-systems. The organization of this chapter is as shown in Fig. 2.1.

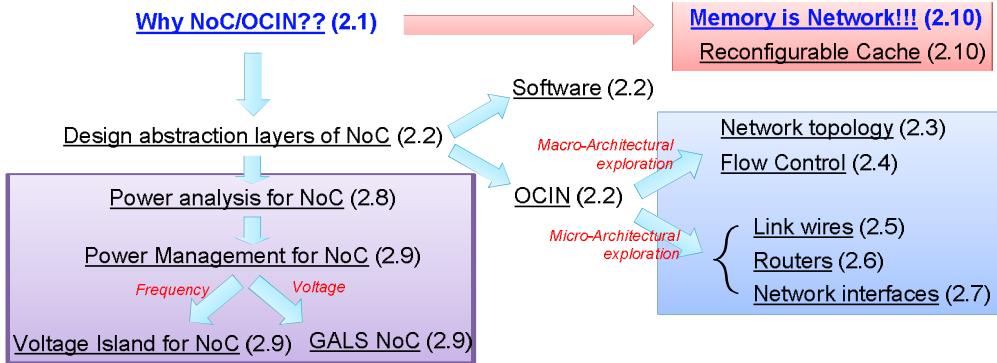


Fig. 2.1 The organization of Chapter 2.

## 2.1 Why NoC and OCIN?

Multi-core SoC designs provide the integrated solution in the communications, multimedia and consumer electronics. Moreover, SoC designs become increasingly complex, while the associated numbers of transistors grows exponentially. Most SoC will find their application within embedded systems, traditional figures of merit, such as performance, energy consumption and cost. However, modern SoC design is faced with a number of problems caused by the scale and complexity of the designs although on-chip bus platforms provide interfaces between PEs and a good verification environment as shown in Fig. 2.2.

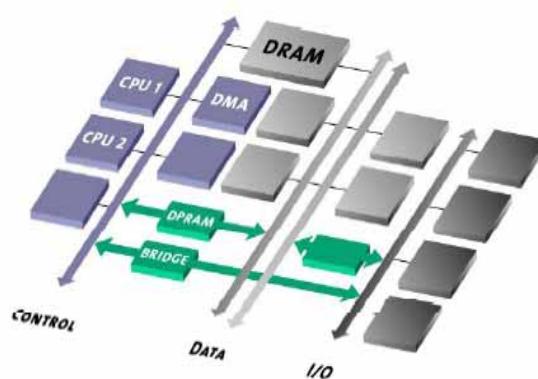


Fig. 2.2 A conventional on-chip bus platform. [2.10]

First, the complexity of on-chip bus platforms increases exponentially while the number of PEs increases linearly [2.10]. The shared bus architectures limit the

development factor for integration with increasing PEs. Existing bus architectures and techniques are proving to be non-scalable, unable to meet leading edge complexity and performance requirements. Second, the interconnect delay across the chip exceeds the average clock period of the IP blocks, especially in nano-scale technologies [2.11]. The ratio of global interconnect delay to average clock period will continue to grow. An interconnect channel design methodology for high performance ICs has proposed in [2.11], it devised a methodology to size the FIFOs in an interconnect channel containing one or more FIFOs connected in series and shows that the sizing of the FIFOs in the channel is a function of system parameters such as data production rate and communication rate, number of channel stages etc.

Third, in nano-scale technologies, increased coupling effect for interconnects not only aggravates the power-delay metrics but also deteriorates the signal integrity due to capacitive and inductive crosstalk noises [2.11]. Several options were proposed to reduce the inter-wire capacitances. The first option is to widen the pitch between bus lines. The second option is using P&R (place & route) tools to avoid routing of the bus lines side by side. However, the interconnect complexity and the routing time do not allow designers trying it to minimize the coupling capacitances. The third option is to change the geometrical shape of bus lines. But the disadvantage of this method is that the frank area will increase since the cross-sectional area of a bus line is fixed. The fourth technique is to add a shielding line (VDD/Ground) between two adjacent signal lines. The fifth option reduces the coupling power consumption via bus encoding schemes [2.12]-[2.16]. However, on-chip physical interconnections will present a limited factor for performance, reliability and energy consumption due to advanced technologies [2.17], [2.18]. Therefore, the encoding schemes for low power and reliability issues were proposed in [2.19], [2.20]. The designers must overcome

the challenge of noises to provide the function correct, reliable operation of the interacting components. A robust self-calibrating transmission scheme for interconnections is proposed in [2.21] and it examines some physical properties of on-chip interconnects, with the goal of achieving fast, reliable and low-energy communication.

Forth, both the system design and performance are limited by the complexity of the interconnection between the different modules and blocks into single clocked design. Different data transfer speeds are required, as well as parallel transmission. The traditional system buses may not be suitable for such a system since only one module can transmit at a time. Additionally, modern multi-core SoC designers assemble the system using ready virtual components which might not be easily adaptable to different clocking situations. The solution to above problems is a segmented bus design combined with the concept of the globally asynchronous locally synchronous (GALS) system architecture [2.22]-[2.24]. Asynchronous design can make the circuits resilient to delay variation.

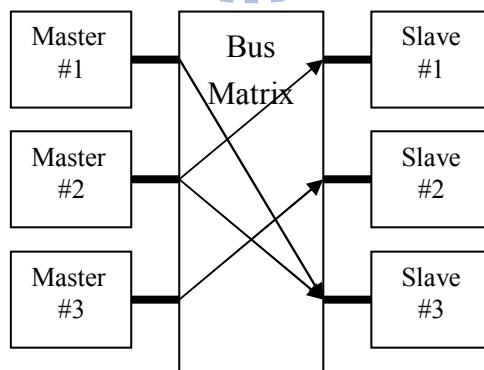


Fig. 2.3 Multi-layer bus architecture. [2.24]

For the above mentioned problems, new architectures for on-chip data communications were proposed to adapt the next multi-core SoC era. A multi-layer on-chip shared bus, as shown in Fig. 2.3, was proposed as an advised version of the

conventional on-chip bus platform to reduce the shared-medium channels [2.24]-[2.26]. Multi-layer on-chip buses enable parallel access paths between multiple masters and slaves by a bus matrix. However, multi-layer bus architectures are confused with complex wire routings inducing larger power consumption and interconnect delay associated with the increasing number of PEs.

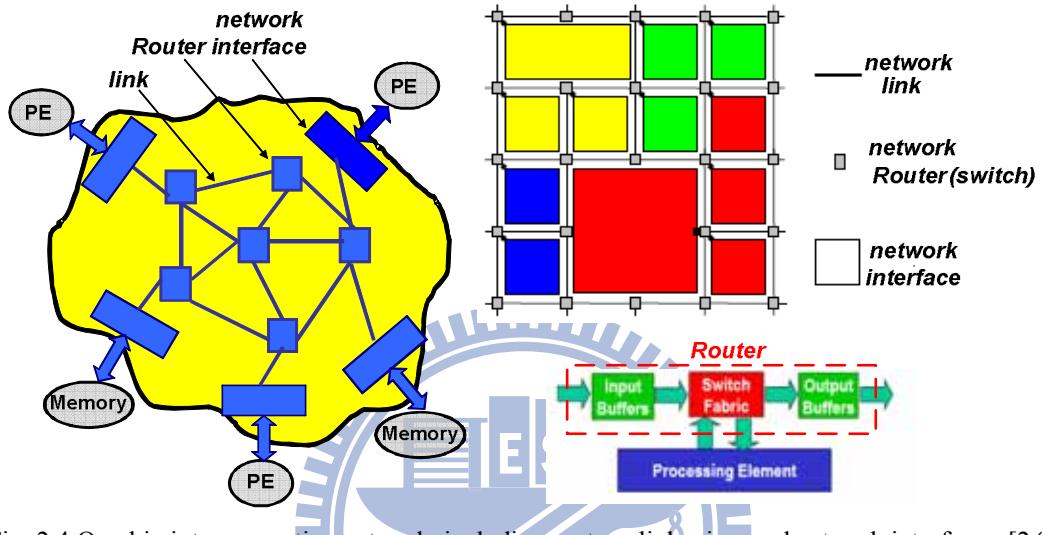


Fig. 2.4 On-chip interconnection network, including routers, link wires and network interfaces. [2.9]

OCIN architecture was proposed based on a scalable switch fabric network, which considers all the requirements of on-chip communications and traffic via routing packets [2.9]. Moreover, OCINs have a few distinctive characteristics, namely low communication latency, energy consumption constraints and design-time specialization. Fig. 2.4 presents the OCIN architecture that provides the building blocks and backbone for NoC platform. The motivation of establishing NoC platform is to achieve performance using a system perspective of communication. The core of NoC technology is the active switching fabric that manages multi-purpose data packets within complex, IP laden designs.

## 2.2 Design Abstraction Levels of NoC

The design of NoC is vast and complex. Therefore, considering on-chip data

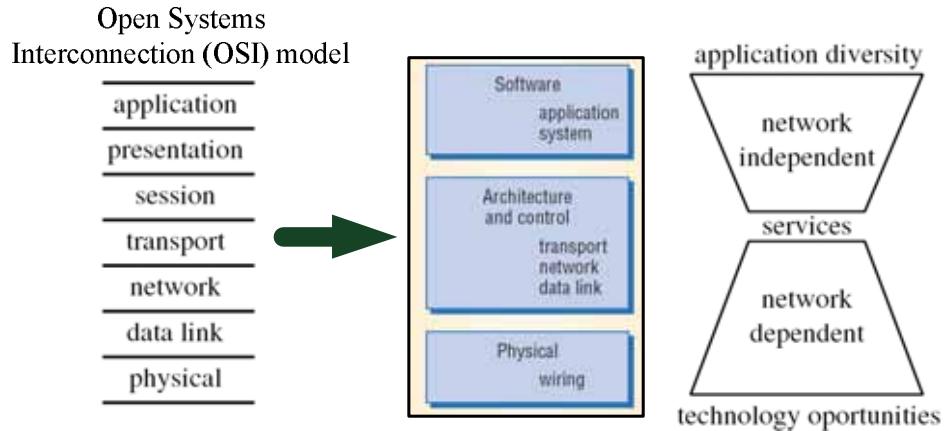


Fig. 2.5 The design abstraction layers of NoC [2.27]

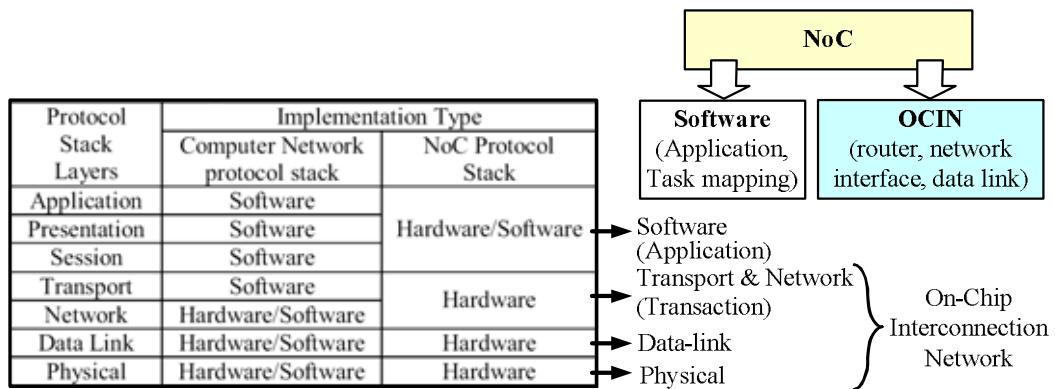
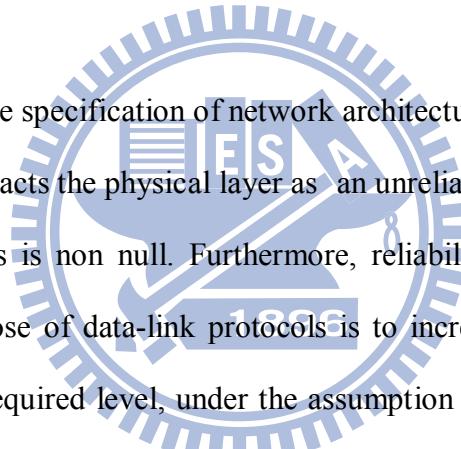


Fig. 2.6 The reduced NoC protocol stack. [2.29]

communication and the abstraction of NoC as a micro-network and analyzing the various levels of this micro-network stack bottom to up is as shown in. NoC models are typical organized starting from physical layer to software layer, in a fashion that resembles the Open Systems Interconnection (OSI) model as shown in Fig. 2.5 [2.27]-[2.28]. However, the OSI model stacks is resembled for a marco-network. For a micro-network, the model stack will be reduced to four layers, namely physical layer, data-link layer, network and transport layer (transaction layer) and software layer [2.29]. Fig. 2.6 shows the reduced NoC protocol stack, and the physical layer, data-link layer, and transaction layer present the design models for OCIN, which constructs the micro-architecture for NoC. Moreover, the research of OCIN can further be divided into micro-architectural innovations within the major components

and macro-architectural choices aiming to seamlessly merge the interconnection backbone with the remaining system modules [2.30].

NoC protocols are described bottom-up, starting from the physical layer up to the application layer. In the physical layer, link wires are the physical implementation of the communication channels. It is important to realize that a well-balanced design should not over design wires so that their behavior approaches an ideal one, because that the corresponding cost in performance, energy-efficiency and modularity may be too high. Physical layer design should find a compromise between competing quality metrics and provide a clean and complete abstraction of channel characteristics to layers above.



NoC design entails the specification of network architectures and control protocols. The data-link layer abstracts the physical layer as an unreliable digital link, where the probability of bit upsets is non null. Furthermore, reliability can be traded off for energy. The main purpose of data-link protocols is to increase the reliability of the link up to a minimum required level, under the assumption that the physical layer by itself is not sufficiently reliable. At the data link layer, error correction can be complemented by several packet-based error detection and recovery protocols. Several parameters in the protocols can be adjusted depending on the goal to achieve maximum performance at a specified residual error probability within given energy consumption bounds.

At the network and transport (transaction) layer, packet data transmission can be customized by the choice of switching and routing algorithms. The NoC designers establish the type of connection to its final destination. Switching and routing affect heavily performance and energy consumption. Robustness and fault tolerance will also be highly desirable. Algorithms deal with the decomposition of messages into

packets at the source and their assembly at destination. Packetization granularity is a critical design decision because the behavior of most network control algorithm is very sensitive to packet size. Packet size can be application specific in SoCs, as opposed to general network.

Software (application) layers comprise system and application software which includes PEs and network operating systems. The system software provides us with an abstraction of the underlying hardware platform. Moreover, policies implemented at the system software layer request either specific protocols or parameters at the lower layers to achieve the appropriate information flow. The hardware abstraction is coupled to the design of wrappers for processor cores which perform as network interfaces between PEs and NoC architecture.

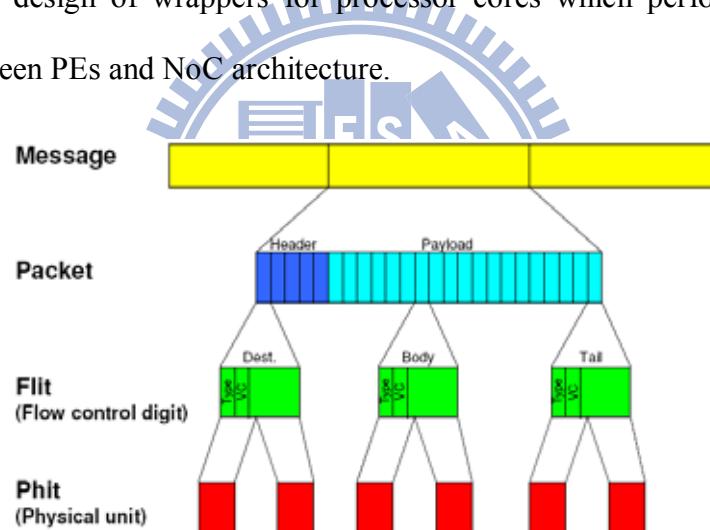


Fig. 2.7 Data abstraction. [2.7]

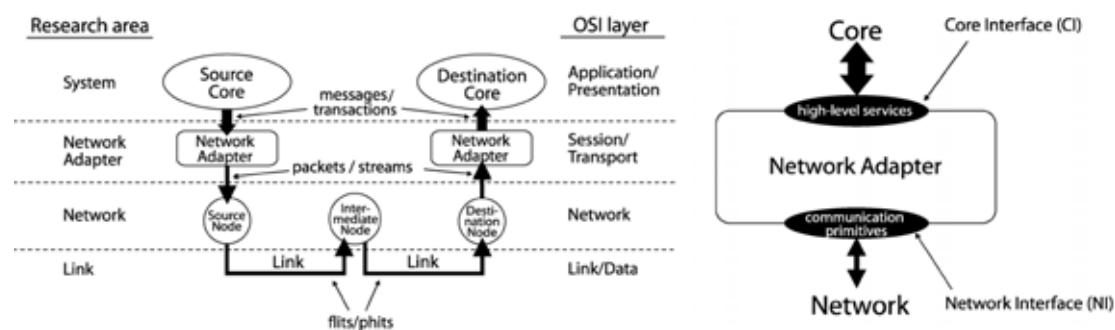


Fig. 2.8 NoC research areas versus OSI model based on the flow of data. [2.31]

The data stream can also be divided into 4 data abstraction layers as shown in Fig.

2.7, which are message, packet, flit and phit (physical transfer unit) [2.7]. Therefore, in addition to the reduced design abstraction layers, the spectrum of NoC research is also divided into four areas based on the flow of data, including system, network adapter, network and link [2.31]. The correspondence between these four areas and OSI models is as shown in Fig. 2.8. The network adapter provides a bridge between high-level services and communication primitives using core interfaces (CIs) and NIs.

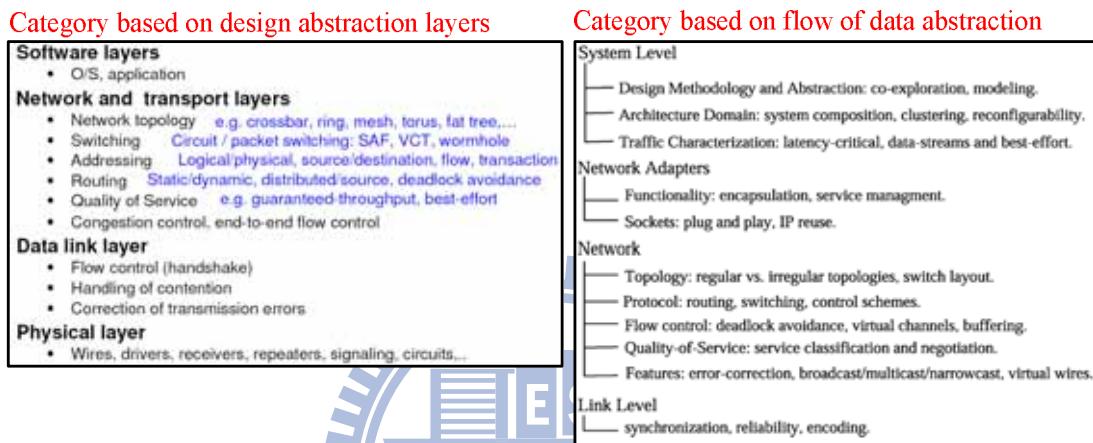


Fig. 2.9 NoC Research category based on design abstraction layers and flow of data abstraction. [2.31]

According to the design abstraction layers and flow of data, the NoC research topics can be categorized as shown in Fig. 2.9 [2.31], [2.32]. In the following sections, the research topics associated with OCINs are introduced, including both macro-architectural exploration (topology) and micro-architectural exploration (building blocks). Moreover, the research related to power analysis, voltage scaling and GALS of NoC is also described.

## 2.3 Network Topologies of OCINs

NoC platforms enable designing parallel systems resembling cellular structures including thousands of PEs. Such systems combined with multi-threaded computing can increase system efficiency for fine-grain parallel programs [2.33], [2.34]. Therefore, the OCIN architecture of NoC should be efficient for a huge amount of

PEs. A number of different OCINs have been proposed as shown in Fig. 2.10. Their origins can be traced back to the field of parallel computing. However, a different set of constraints exists when adapting these architectures to the multi-core SoC design paradigm.

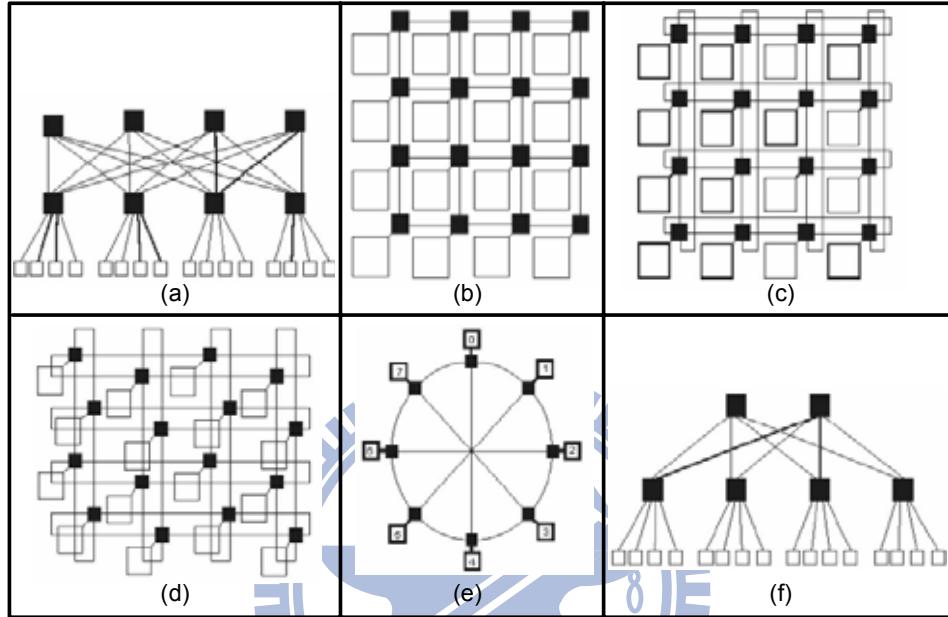


Fig. 2.10 Conventional network topologies of OCIN (a) SPIN (b) Mesh (c) Torus (d) Folded tours (e) Octagon (f) Butterfly Fat tree. [2.35]

A generic interconnect template was proposed which is called SPIN (Scalable, Programmable, Integrated Network) for on-chip packet switched interconnections as shown in Fig. 2.10(a), where a fat-tree architecture is used to interconnect PEs [2.35]. In this fat tree, every node has four children and the parent is replicated four times at any level of the tree. The functional PEs reside at the leaves and the switches reside at the vertices. A mesh-based (tile-based) OCIN architecture consists of an  $m \times n$  mesh of switches interconnecting computational resources (PEs) placed along with the switches, as shown in Fig. 2.10(b). Every switch (router), except those at the edges, is connected to four neighboring switches and one PE.

2D torus was proposed as an OCIN [2.36], as shown in Fig. 2.10(c). The Torus

architecture is basically the same as a regular mesh. The only difference is that the switches at the edges are connected to the switches at the opposite edge through wrap-around channels. Every switch has five ports, one connected to the local resource and the others connected to the closest neighboring switches. The long end-around connections can yield excessive delays. However, this can be avoided by folding the torus as Fig. 2.10(d) [2.37]. This renders to a more suitable VLSI. The OCTAGON MP-SoC architecture was proposed in [2.38]. Fig. 2.10(e) shows a basic octagon unit consisting of eight nodes and 12 bidirectional links. Each node is associated with a processing element and a switch. Communication between any pair of nodes takes at most two hops within the basic octagonal unit. For a system consisting of more than eight nodes, the octagon is extended to multidimensional space. This type of interconnection mechanism may significantly increase the wiring complexity. In a Butterfly Fat-Tree (BFT) architecture which is shown as Fig. 2.10(f), PEs are placed at the leaves and switches placed at the vertices [2.39]. A pair of coordinates is used to label each node. The number of switches in the butterfly fat tree architecture converges to a constant independent of the number of levels. Other high-radix topologies were also studied as OCIN architectures [2.40], [2.41]. However, the complexity of the switching circuits in high-radix topologies induces huge amount of area and power consumption.

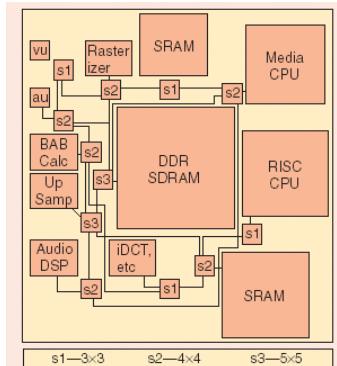


Fig. 2. 11 Xipies Architecture. [2.42]

A popular network topology of OCIN implementations is the two-dimensional mesh architecture as mentioned above, providing a regular topology and communications. Therefore, many advanced OCIN topologies are designed based on this mesh network. An advanced OCIN, called Xpipes as shown in Fig. 2. 11, targeting high performance and reliable communication for on-chip multi-processors was presented in [2.42]. Data links can be pipelined with a flexible number of stages to decouple link throughput from its length and to get arbitrary topologies. The I/O ports of each switch can be parameterized, and Xpipes is optimized from mesh-based OCIN architecture.

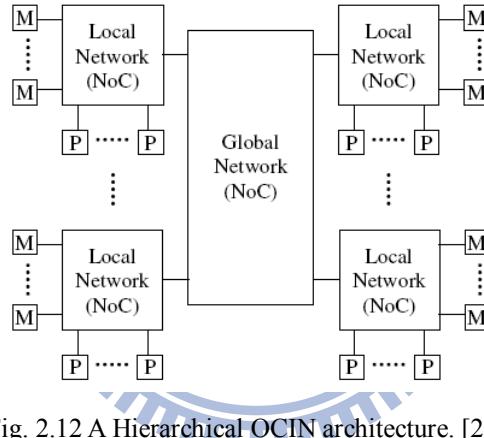


Fig. 2.12 A Hierarchical OCIN architecture. [2.43]

A hierarchical OCIN architecture was presented and constructed via local network and global network as shown in Fig. 2.12 [2.43]. The local network preserves the features of 2-D links network on chip, and the global network is designed as centralized crossbar. Other hierarchical OCIN or hybrid OCIN topologies were also proposed to adopt multiple PEs and heterogeneous systems [2.44]-[2.47]. Energy consumption and area of hierarchical OCIN architectures were analyzed as shown in Fig. 2.13 [2.48]. Fig. 2.13(a) shows the comparison result of the energy consumption under the uniform traffic. Although the mesh has short and regular length of links, it has more hop counts than the star thus the energy cost of the mesh is 40%-50% higher

than the star. Among the hierarchical topologies excluding the hierarchical point-to-point topology, the hierarchical star (locally star globally star or H-star) topology shows the lowest energy cost under any kinds of traffic. The network area cost including the area of switches, multiplexers/ demultiplexers, and links are also analyzed as shown in Fig. 2.13(b). The area of point-to-point topologies is skyrocketing as the increases because of their huge link wires interconnecting every PU pair. This is the major reason which makes the point-to-point topology impractical to implement. The area consumption of the hierarchical topologies is as small as bus topologies. Considering the energy and area cost together, the hierarchical star topology is the most energy-efficient and cost-effective topology in general.

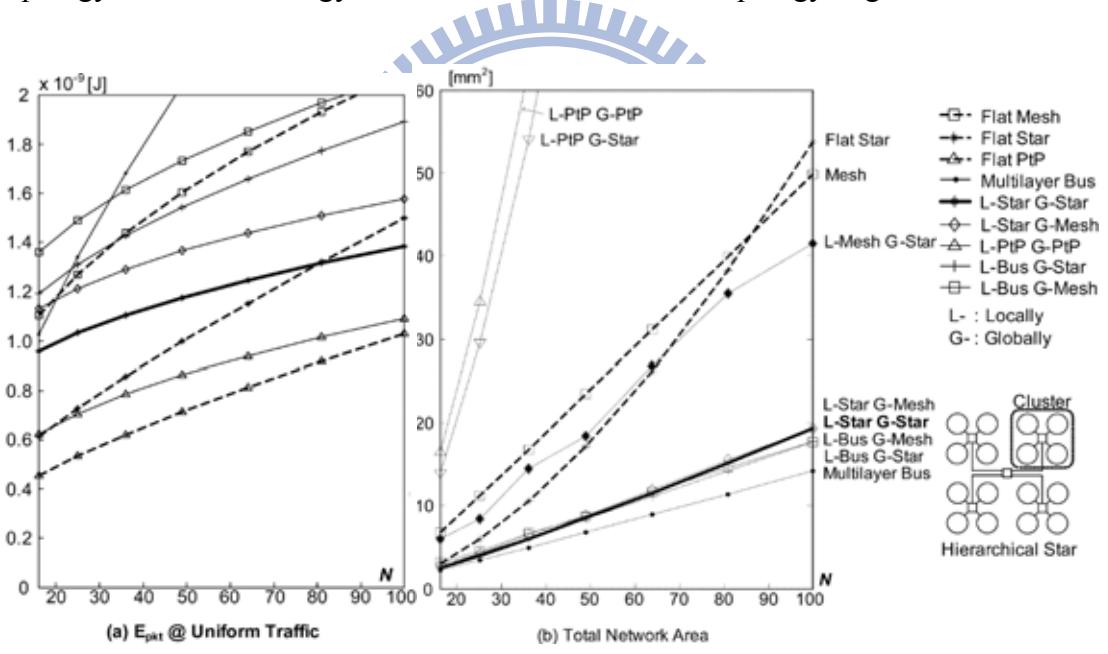


Fig. 2.13 (a) Energy consumption (b) network area according to a number of PEs. [2.48]

In order to achieve better performance, functionality and packaging density, through-silicon-via (TSV) three-dimensional (3D) ICs were proposed with multiple layers of active devices [2.49]. Additionally, TSV 3D-ICs allow for performance enhancements in the absence of scaling. The performance improvement arising from the architectural advantages of NoCs will be significantly enhanced if TSV 3D-ICs are adopted as the basic fabrication methodologies. Therefore, new topologies of TSV

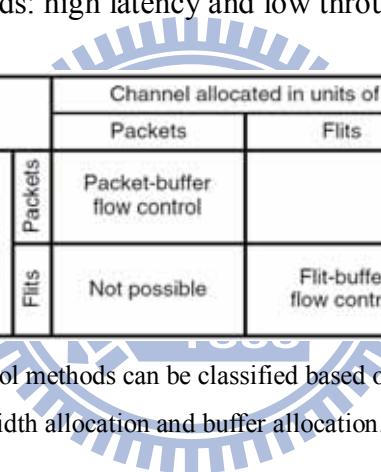
3-D network were also proposed for the future ICs [2.50]-[2.53].

## 2.4 Flow Control and Switching Technique for OCINs

Flow control determines how a network's resource such as channel bandwidth, buffer capacity, and control state, are allocated to packets traversing the network [2.6], [2.8]. Flow control is tightly coupled with buffer management algorithms that determine how messages are handled when blocked in the network. One can view flow control as either a problem of resource allocation or one of contention resolution. From the resource allocation perspective, resources in the form of channels, buffers, and state must be allocated to each packet as it advances from the source to the destination. Additionally, flow control provides a synchronization protocol for transmitting and receiving a unit of information. The unit of flow control refers to that portion of the message whose transfer must be synchronized. This unit is defined as the smallest unit of information whose transfer is requested by the sender and acknowledged by the receiver. The request/acknowledgement signaling is used to ensure successful transfer and the availability of buffer space at the receiver.

The switching techniques for OCINs determine when and how internal switches are set to connect router inputs to outputs and the time at which message components may be transferred along these paths. These switching techniques are coupled with flow control mechanisms for the synchronized transfer of units of information between routers and through routers in forwarding message through network. The switching techniques can be categorized via bufferless flow control and buffered flow control first [2.54]. The simplest flow-control mechanisms are bufferless, and rather than temporarily storing blocked packets, they either drop or misroute these packets. These forms of flow control use no buffering and simply act to allocated channel state

and bandwidth to competing packet. In these cases, the flow-control methods must perform an arbitration to decide which packet gets the channel it has requested. The arbitration method must also decide how to dispose of any packets that did not get their requested destination. Circuit switching is a form of bufferless flow control that operates by first allocating channels to form a circuit from source to destination and then sending one or more packets along this circuit [2.54]. When no further packets need to be sent, the circuit is deallocated. Circuit switching differs from dropping flow control in that if the request flit is blocked, it is held in place rather than dropped. However, circuit switching has two weaknesses that make it less attractive than buffered flow control methods: high latency and low throughput.



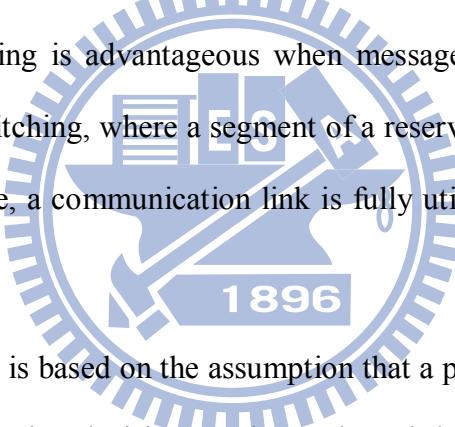
		Channel allocated in units of	
		Packets	Flits
Buffer allocated in units of	Packets	Packet-buffer flow control	
	Flits	Not possible	Flit-buffer flow control

Fig. 2.14 Buffered flow control methods can be classified based on their granularity of channel bandwidth allocation and buffer allocation. [2.54]

Adding buffers to OCINs results in significantly more efficient flow control, allowing the allocation of the second channel to be delayed without complications. Storing a flit (or a packet) in a buffer decouples allocation of the input channel to a flit from the allocation of the output channel to a flit. Adding a buffer prevents the waste of the channel bandwidth caused by dropping or misrouting packets or the idle time inherent in circuit switching. As a result, full channel utilization with buffered flow control can be realized via buffered flow control. Therefore, the buffered flow control mechanisms should allocate buffers considering channel bandwidth. Moreover, the buffered flow control mechanisms depend on the granularity of the buffer allocation and channel allocation as depicted in Fig. 2.14 [2.54].

### 2.4.1 Packet-Buffer Flow Control

The packet-buffer flow control channel allocates bandwidth and buffers in units of packets. A packet is completely buffered at each intermediate node before it is forwarded to the next node. This is the reason why this switching technique is also referred to as store-and forward (SAF) switching [2.7], [2.54]. The packet must be allocated two resources before it can be forwarded: a packet-sized buffer on the far side of the channel and exclusive use of the channel. Once the entire packet has arrived at a node and these two resources are acquired, the packet is forwarded to the next node. While waiting to acquire resources, if they are not immediately available, no channels are being held idle and only a single packet buffer on the current node is occupied. Packet switching is advantageous when messages are short and frequent [2.55]. Unlike circuit switching, where a segment of a reserved path may be idle for a significant period of time, a communication link is fully utilized when there are data to be transmitted.

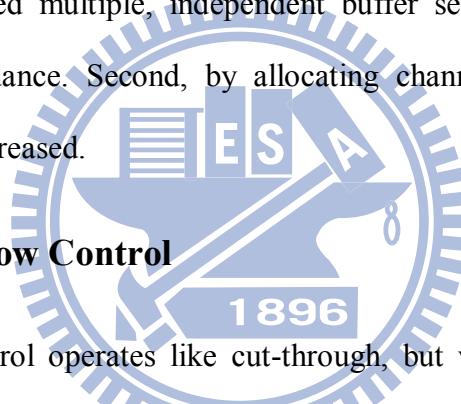


SAF packet switching is based on the assumption that a packet must be received in its entirety before any routing decision can be made and the packet forwarded to the destination. Rather than waiting for the entire packet to be received, the packet header can be examined as soon as it is received. The router can start forwarding the header and following data bytes as soon as routing decisions have been made and the output buffer is free. In fact the message does not even have to be buffered at the output and can cut through to the input of the next router before the complete packet has been received at the current router. This switching technique is referred to as virtual cut-through switching (VCT) [2.7], [2.54]. In the absence of blocking, the latency experienced by the header at each node is the routing latency and propagation delay through the router and along the physical channels. If the header is blocked on a

busy output channel, the complete message is buffered at the node. Thus, at high network loads, VCT switching behaves like SAF.

VCT flow control reduced the latency from the product of the hop count and the serialization latency, giving very high channel utilization by using buffers to decouple channel allocation. It also achieves very low latency by forwarding packets as soon as possible. However, the cut-through method, or any other packet-based method, has two serious shortcomings. First, by allocating buffers in units of packets, it makes very inefficient use of buffer storage. As we shall see, we can make much more effective use of storage by allocating buffers in units of flits. This is particularly important when we need multiple, independent buffer sets to reduce blocking or provide deadlock avoidance. Second, by allocating channels in units of packets, contention latency is increased.

#### 2.4.2 Flit-Buffer Flow Control

The watermark features the Texas A&M University seal, which is a circular emblem. The outer ring contains the text "TEXAS A&M UNIVERSITY". Inside the ring, there is a central figure, possibly a star or a more complex emblem, with the year "1896" prominently displayed at the bottom. The entire watermark is semi-transparent.

Wormhole flow control operates like cut-through, but with channel and buffers allocated to flits rather than packets [2.8]. In wormhole switching, the buffer requirements within the routers are substantially reduced over the requirement for VCT switching. The primary difference between wormhole switching and VCT switching is that, in the former, the unit of message flow control is a single flit and, as a consequence, small buffers can be used. Compared to cut-through flow control, wormhole flow control makes far more efficient use of buffer space, as only a small number of flit buffers are required per virtual channel. In contrast, cut-through flow control requires several packets of buffer space, which is typically at least an order of magnitude more storage than wormhole flow control. This savings in buffer space, however, comes at the expense of some throughput, since wormhole flow control may

block a channel mid-packet. Blocking may occur with wormhole flow control because the channel is owned by a packet, but buffers are allocated on a flit-by-flit basis.

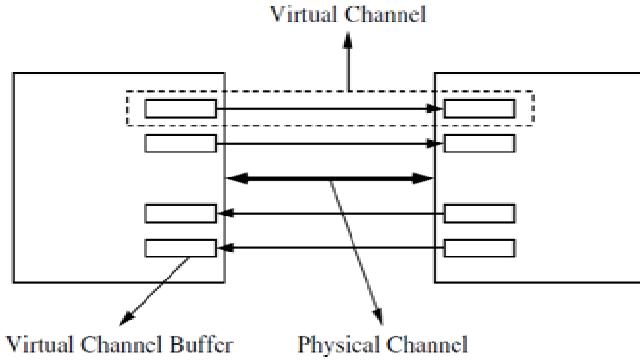


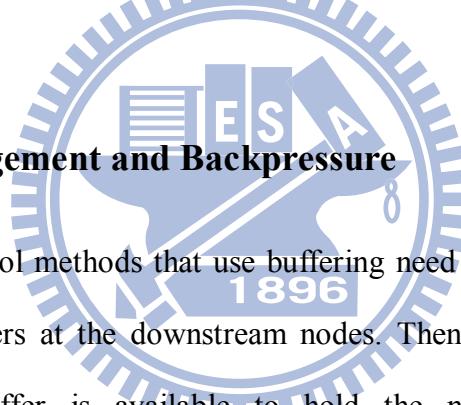
Fig. 2.15 Virtual channels. [2.8]

Once a message occupies a buffer for a channel, no other message can access the physical channel, even if the message is blocked. Alternatively, a physical channel may support several logical or virtual channels multiplexed across physical channel. Each unidirectional virtual channel is realized by an independently managed pair of message buffers as illustrated in Fig. 2.15 [2.8], [2.56]. Each message can share the physical channel on a flit-by-flit basis. The physical channel protocol must be able to distinguish between the virtual channels using the physical channel. Logically, each virtual channel operates as if each were using a distinct physical channel operating half the speed. Virtual channel were originally introduced to solve the problem of deadlock in wormhole-switched networks. Deadlock is a network state where no messages can advance because each message requires a channel occupied by another message. By allowing messages to share a physical channel, messages can make progress rather than remain blocked. Virtual channels can also be used to improve message latency and network throughout. Virtual-channel flow control decouples the allocation of channel state from channel bandwidth. This decoupling prevents a packet that acquires channel state and then blocks from holding channel bandwidth idle. This permits virtual-channel flow control to achieve substantially higher

throughput than wormhole flow control.

As in wormhole flow control, an arriving head flit must allocate a virtual channel, a downstream flit buffer, and channel bandwidth to advance. Subsequent body flits from the packet use the virtual channel allocated by the header and still must allocate a flit buffer and channel bandwidth. However, unlike wormhole flow control, these flits are not guaranteed access to channel bandwidth because other virtual channels may be competing to transmit flits of their packets across the same link. In fact, given the same total amount of buffer space, virtual-channel flow control also outperforms cut-through flow control because it is more efficient to allocate buffer space as multiple short virtual-channel flit buffers than as a single large cut-through packet buffer.

#### 2.4.3 Buffer Management and Backpressure



All of the flow control methods that use buffering need a means to communicate the availability of buffers at the downstream nodes. Then the upstream nodes can determine when a buffer is available to hold the next flit (or packet for store-and-forward or cut-through) to be transmitted. This type of buffer management provides backpressure by informing the upstream nodes when they must stop transmitting flits because all of the downstream flit buffers are full. Three types of low-level flow control mechanisms are in common use today to provide such backpressure[2.57]: credit-based [2.58], on/off, and ack/nack [2.59].

With credit-based flow control [2.60], [2.61], the upstream router keeps a count of the number of free flit buffers in each virtual channel downstream. Then, each time the upstream router forwards a flit, thus consuming a downstream buffer, it decrements the appropriate count. If the count reaches zero, all of the downstream

buffers are full and no further flits can be forwarded until a buffer becomes available. Once the downstream router forwards a flit and frees the associated buffer, it sends a credit to the upstream router, causing a buffer count to be incremented.

On/off flow control can greatly reduce the amount of upstream signaling in certain cases. With this method the upstream state is a single control bit that represents whether the upstream node is permitted to send (on) or not (off). A signal is sent upstream only when it is necessary to change this state. An off signal is sent when the control bit is on and the number of free buffers falls below the threshold  $X_{off}$ . If the control bit is off and the number of free buffers rises above the threshold  $X_{on}$ , an on signal is sent. With an adequate number of buffers, on/off flow control systems can operate with very little upstream signaling.

Ack/nack flow control reduces the minimum of this buffer vacancy time and the average vacancy time. Unfortunately there is no net gain because buffers are held for an additional waiting for an acknowledgment, making ack/nack flow control less efficient in its use of buffers than credit-based flow control. It is also inefficient in its use of bandwidth which it uses to send flits only to drop them when no buffer is available. With ack/nack flow control, there is no state kept in the upstream node to indicate buffer availability. The upstream node optimistically sends flits whenever they become available. If the downstream node has a buffer available, it accepts the flit and sends an acknowledge (ack) to the upstream node. If no buffers are available when the flit arrives, the downstream node drops the flit and sends a negative acknowledgment (nack). The upstream node holds onto each flit until it receives an ack. If it receives a nack, it retransmits the flit. Because of its buffer and bandwidth inefficiency, ack/nack flow control is rarely used. Rather, credit-based flow control is typically used in systems with small numbers of buffers, and on/off flow control is

employed in most systems that have large numbers of flit buffers.

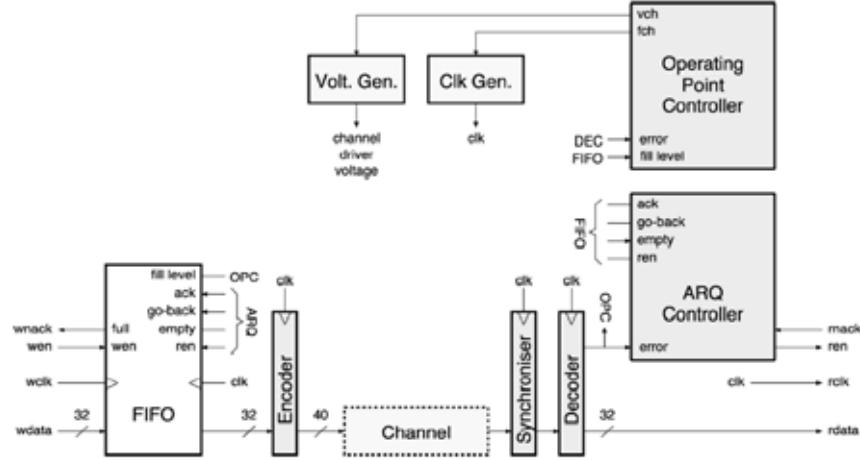


Fig. 2.16 A robust self-calibrating transmission scheme for OCIN links. [2.62]

## 2.5 Link Wires for OCINs

The interconnection is used to distributed clock and signals, and to provide power lines among PEs on a chip. Generally speaking, high performance signaling has two modes, which are voltage-mode signaling and current-mode signaling. Voltage-mode signaling uses current to charge the capacitive load of the link wire, using the level of voltage to decide the logic high or low. Current-mode signaling does not need to charge the node to full swing, and can sense the current direction and decide the logic. To simplify the design, voltage-mode signaling is usually adopted for link wires in OCINs [2.7].

The features of OCIN links consists of regularity, point-to-point (no fan-out tree), and well-defined current return path [2.32]. Additionally, the OCIN links can be optimized for noise/speed/power at bit-level and flit-level both. At flit-level, dynamic voltage scaling for link wires is presented to reduce energy consumption [2.62]. A variable frequency and swing technique as shown in Fig. 2.16 is proposed to trade off speed for energy, which is achieved by dynamic voltage-swing scaling and two

controllers, automatic repeat-request (ARQ) controller and operation-point controller [2.62]. ARQ controller decides which words to push through the channel, and operation-point controller is picking up the lowest frequency and voltage swing to meet constraint. This approach exploits a variable relation between operating frequency and voltage swing to find the best safe operating point in current environmental conditions, by monitoring the error rate.

For OCIN links, three critical issues should be addressed, which are delay, power and reliability. Coding theory is an effective solution to deal with these three challenges at flit-level. Low-power coding (LPC) schemes were presented to reduce the transition activity of transmission data in physical channels [2.63], [2.65]. Crosstalk avoidance coding (CAC) algorithm reduces the delay and the power consumption by forbidding specific transitions as to reduce the crosstalk effect [2.65]-[2.68]. Error control coding (ECC) guarantees error-free transmission where the coding algorithm has to provide a reliability boundary [2.69], [2.70]. The ECC is able to detect the error and re-transfer the data. Even more, some will correct the error in the receiver. LPC and CAC can be combined in an algorithm to their correlation properties. This is because that avoiding crosstalk between two adjacent lines will also reduce the power consumption by decreasing coupling effect. By modifying the structure of the data packets and reducing the coding-decoding operations can achieve energy reduction [2.67]. The method incorporating with CAC schemes in NoC data packets enable to save a significant energy. The schemes divides packets into fix-length flow control units (flits), the header contains routing information which can be decoding enables the switches to establish the path. The subsequent flits just follow the path and transmit whole flits in concept of pipeline. Only the header flits contain the control information and the body flits don't. If the packet structure can be

modified in such a way that coding/decoding is needed only at the source and destination nodes, then there will be no extra power dissipation arising out of the codec blocks in the intermediate nodes. Eventually this will help to reduce the overall communication energy dissipation.

Incorporating of different coding schemes have been investigated to increase system reliability. CAC algorithms reduce the worse-case switching capacitance of a wire by ensuring that a specific codeword transitions doesn't happen. However, OCIN links are sensitive to internal (power supply noise, crosstalk noise, inter-symbol interference) and external (electromagnetic interference, thermal noise, noise by alpha particles) noise sources due to lower supply voltage, smaller node capacitances, a decrease of wire spacing, the increasing role of coupling capacitances, the higher clock frequency. Therefore, CAC incorporates with forward error correction coding is a solution for a robust NoC system. Jointing CAC and single error correction (SEC) codes are proposed as duplicate-add-parity(DAP) [2.71], boundary shift code (BSC) [2.72] and modified dual rail (MDR) code [2.73].

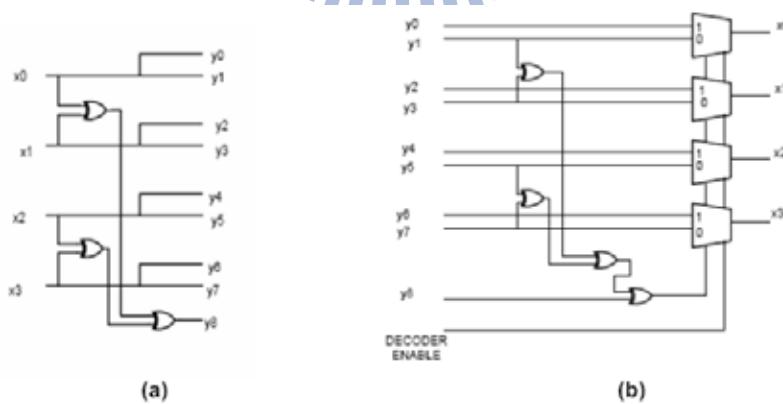


Fig. 2.17 Duplicate-add-parity(DAP) coding (a)Encoder (b)Decoder. [2.71]

Duplicate-add-parity encoder/decoder is as shown in Fig. 2.17. The encoder duplicates data( $x_0, x_1, x_2, x_3$ ) and generates ( $y_1, y_3, y_5, y_7$ ).  $y_8$  is a parity bit generated from  $x_0 \ x_1 \ x_2 \ x_3$  which means if data has odd “1”  $y_8=1$ , else (even “1”)  $y_8=0$ .

The Modified Dual Rail (MDR) code is similar to DAP code. In the MDR code, two copies of the parity bit are placed adjacent to the other codeword bits reduce for reducing crosstalk. The conventional error control code, such as Hamming code, requires redundant bits to detect and correct error bits. However, the power overhead of the redundant bits is significant. Therefore, BSC as shown in is generated by copying each bit, and an extra parity indicates the even/odd of inputs. Fig. 2.18 presents the implementation of BSC encoder and decoder. However, BSC has a disadvantage of large gate delay with the increasing code-width.

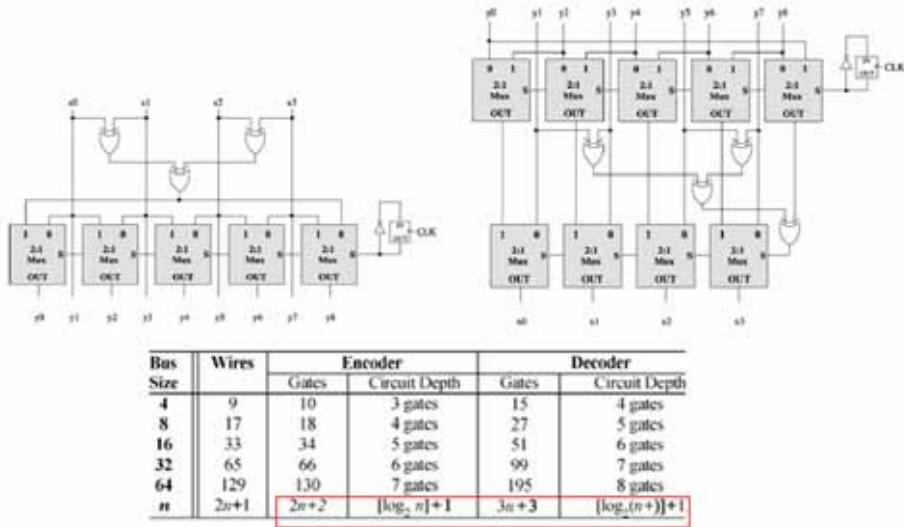


Fig. 2.18 Implementation of boundary-shift codes (BSC). [2.72]

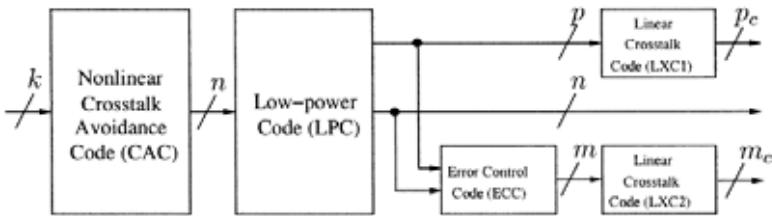


Fig. 2.19 A unified coding framework for link wires. [2.71]

A unified framework was proposed to combine LPC, CAC, and ECC for realizing high-speed, low-power and reliable links [2.71]. However, these codes also affect each other and reduce the efficiency. Therefore, the authors combine different codes by some rules and the coding flow is shown as Fig. 2.19. The rules are as follow:

- (1) CAC needs to be the outermost code.
- (2) LPC can follow CAC.
- (3) ECC needs to be systematic.
- (4) The additional information bits generated by LPC (p) and ECC (m) need to be encode through linear crosstalk code (LXC).

This framework derives a wide variety of joint codes which enables the trade-off between delay, power, reliability and area. In addition to coding and voltage scaling for link wires, a bi-direction channel design has been proposed to maximize the utilization of link wires based on the traffic conditions as shown in Fig. 2.20 [2.74], [2.75]. This scheme allows each communication channel to be dynamically self-configured to transmit flits in either direction.

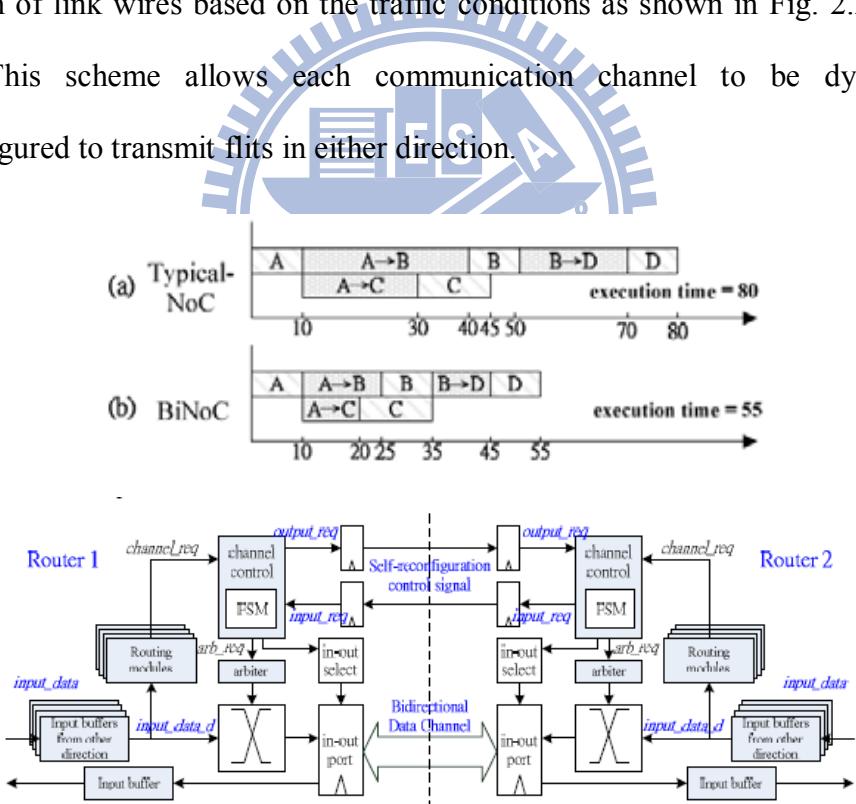


Fig. 2.20 A bi-direction channels to optimize the utilization of link wires. [2.74]

For realizing high-performance and low power links at bit-level, an energy recovery technique is adopted for OCIN links [2.76]. Dynamic voltage scaling for link wires is also presented to reduce energy consumption at bit-level via lookahead-based

adaptive voltage scheme [2.77]. Fig. 2.21 shows this adaptive voltage scheme based on the transition detection circuit. Based on the detection signal, the voltage control adjusts the voltage swing of this interconnect. Furthermore, wave-pipelining is a well-known technique for OCIN links at bit-level that increases the throughput of the interconnect [2.78].

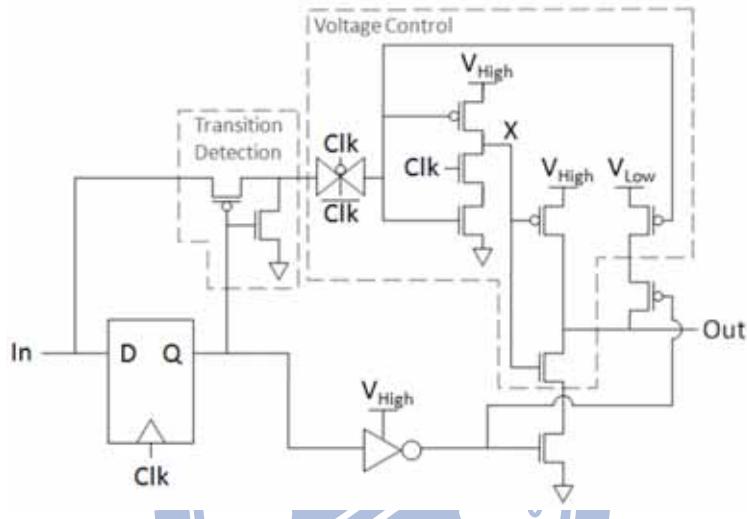


Fig. 2.21 Lookahead-based adaptive voltage scheme. [2.77]

## 2.6 Routers for OCINs

Routers (or called switching fabrics) are kernel components in OCIN to propagate the data information. The implementation of routers depends on the topology, flow control and protocol of OCIN. Fig. 2.22 presents the micro-architecture of a router for mesh-based OCINs. Generally, the router consists of five parts, including I/O port, link control unit (routing unit), buffering (queues), switching circuit and arbitration unit. The link control units (routing units) manage the data communication in the OCIN backbone, and the arbitration unit arbitrates contention data which are routed to the same channel. The OCIN former should avoid deadlock of the on-chip communications and traffic which are intruded by the bad policy routing algorithms. The details of each unit will be described in the following sections.

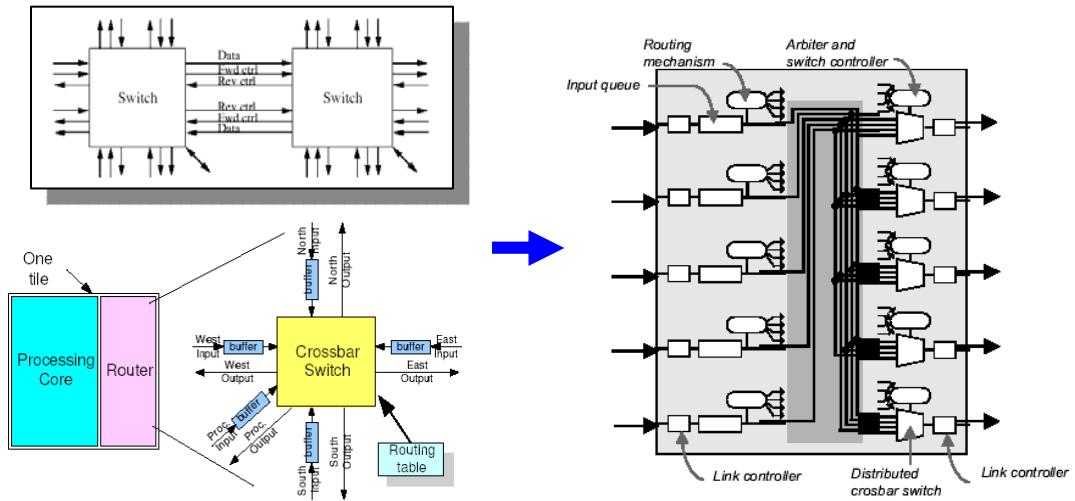


Fig. 2.22 Micro-architecture of a router for mesh-based OCINs. [2.9]

### 2.6.1 Routing Algorithm for Link Control

The design of link control unit highly relies on the topologies of OCINs. Most routing algorithms were developed for regular mesh OCINs. The routing schemes for mesh OCINs can be classified into several categories. The routing decision at every router can be static or dynamic. In static routing schemes (or called oblivious routing), the path is completely determined the source and destination address [2.79], [2.80]. The routing scheme does not consider the current load and the situation on the network. On the contrary, the dynamic routing scheme decides the path by not only source and destination address but also the dynamic network condition. The advantage of static routing scheme is its simplicity of design and hardware overhead. Dynamic routing schemes can use alternative paths which consider the network traffic situation but it may cost much hardware resource [2.81].

Furthermore, routing techniques can be classified according to where the routing information decisions are determined. In distributed routing, the routing decisions are decided in each router. Therefore, each packet carries its source and destination addresses, and the router according to the information to lookup the routing table or to

execute the routing function for the routing decisions. For example, XY routing, a very common routing in OCIN, the router compares the router address and the destination address, and the routing will first arrive the X coordinate, then arrive Y coordinate [2.82]-[2.85]. In source routing, the pre-computed routing tables are stored in the NIs. The routing decisions are determined according to the destination address and the routing table at the source router. Each packet carries its header including routing choices for each hop along its path. When the packet arrive a router, its next routing output is read from the header file. In comparison to distributed routing, source routing does not need extra routing table or function in the intermediate router [2.86]-[2.88]. On the other hand, it requires a source route header in the packet header and requires additional routing tables for each source. Moreover, the classification can be distinguished between minimal and non-minimal distance routing [2.89], [2.90]. Note that the minimal power routing is not equal to the minimal distance routing. Based on the above description of routing algorithm, a category of routing algorithms grouping is given as shown in Fig. 2.23 [2.91].

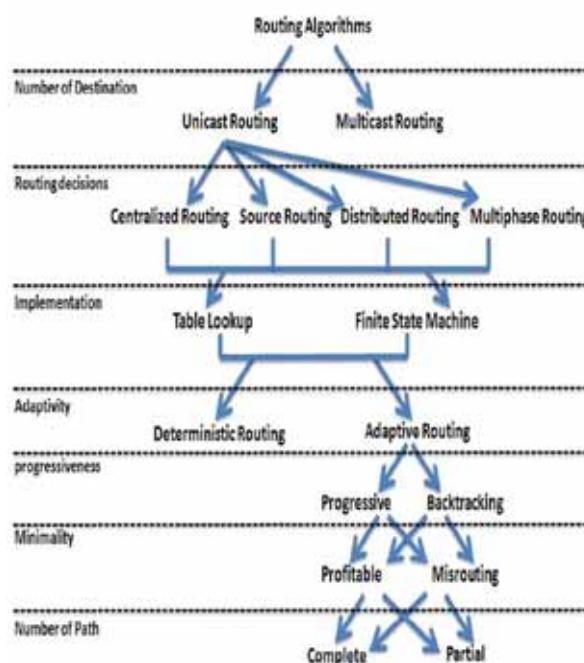


Fig. 2.23 A category of Routing Algorithms Grouping. [2.91]

The major constraint for any routing algorithm is assuring the freedom from deadlock. The definition of deadlock is that a packet is blocked at some intermediate resource and cannot reach its destination. Deadlock occurs when one or more packets in the network are blocked during an indefinite time, waiting for an event that can't happen. An example as depicted in Fig. 2.24 is a situation where four packets are routed in a circle between the routers in a square mesh. The packet in A1 router is allocated to west and another packet in A2 router is allocated to south, and so as A3 and A4. The four packets are already held by another packet and will never be released [2.92], [2.93].

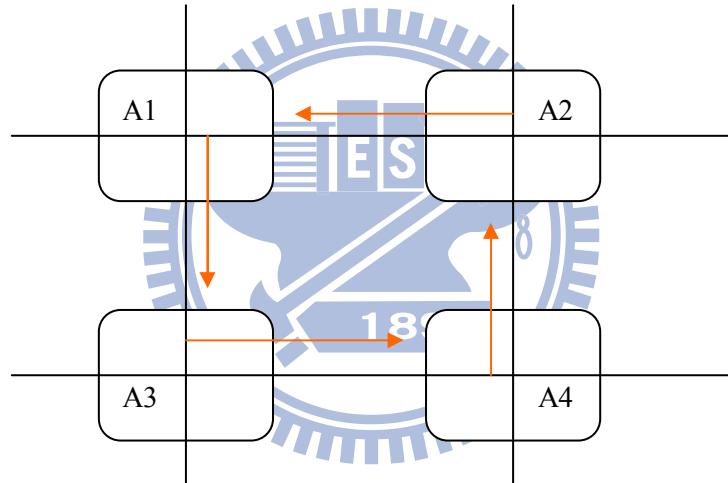


Fig. 2.24 An example of deadlock. [2.92]

The prominent strategy for dealing with deadlock problems is avoidance and most deadlock-free routing algorithms are deduced by the strategy. (1) Choose a particular routing algorithm. (2) Check whether this algorithm is deadlock free. (3) If need, add hardware resource or restrict routing to guarantee that this algorithm is deadlock free. Deadlock free can be analyzed via building a dependency graph, and this dependency graph cannot be cyclic. Another constrain for the routing algorithms is livelock. The definition of livelock is a packet enters a cyclic path dose not reach its destination. Livelock is only induced by dynamic routing algorithms.

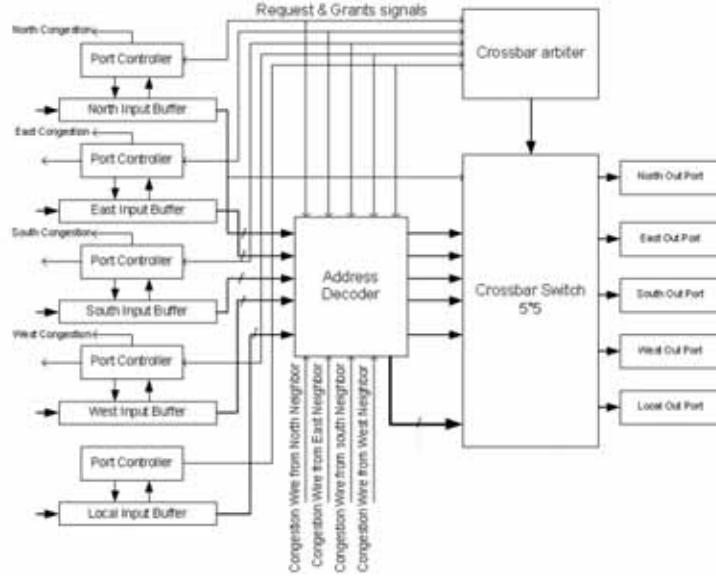


Fig. 2.25 A dynamic routing algorithm for avoiding hot spots. [2.90]

Several routing algorithms of the mesh-OCIN have been proposed for enhancing performance, and are described as follows. A dynamic routing algorithm for avoiding hot spots was proposed in [2.90] as shown in Fig. 2.25. Each input port has input buffer and a controller. The input controller receives packets from the link and requests the crossbar arbiter for the packet injection grant. Additionally, this input controller also detects whether the space of the input buffer is becoming full or empty via the full/empty status flag. The full/empty status flag indicates the congestion condition of the router.

Neighbors-on-Path (NoP) was presented as shown in Fig. 2.26 [2.94]. A header flit is transferred from the node (0,0) to the node (1,0). In the node (1,1), the score of next nodes (1,2) and (2,1) is computed. Additionally, the next nodes must be on minimal paths. The decision of next node is according to its score. For example, Fig. 2.26 shows that two nodes (1,2) and (2,1) check whether their neighbors are available or not. If available, mark a circle. Fig. 2.26(a) and Fig. 2.26(b) indicate that the scores are determined using the number of available nodes. The score of node (2,1) is 1, and node (1,2) is 2. Consequently, node (1,1) will select node (1,2) as the output.

Additionally, (1,3) and (0,2) also send the information of available to (1,2). Nevertheless, these two nodes are not on minimal paths with given destination (3,2) as shown in Fig 2.4 (c)-(d). In view of this, only two nodes (1,2) and (2,1) send the score to (1,1), and node (1,1) decides that the next node is (2,1).

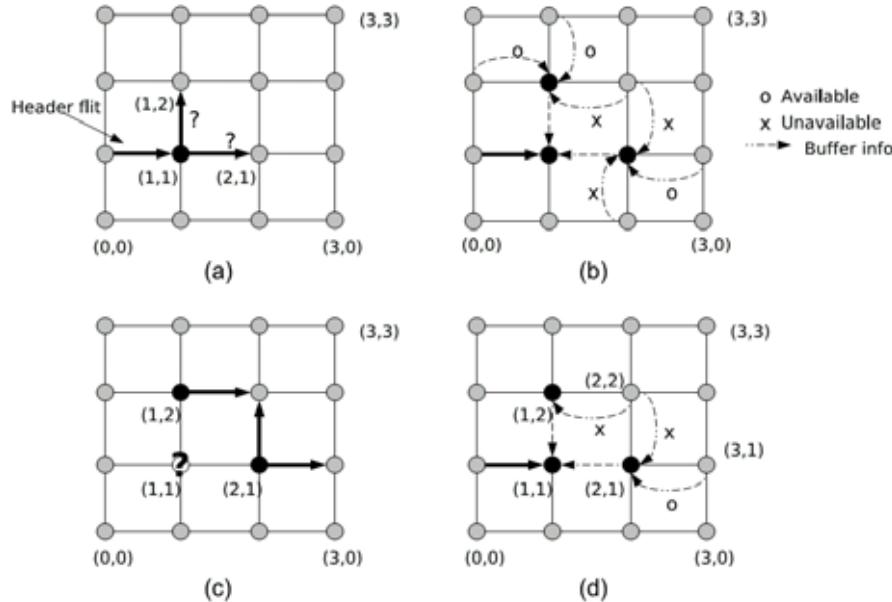


Fig. 2.26 Example for neighbors-on-path algorithm. [2.94]

Antnet algorithm uses a nature method to route the least latency way [2.95]. At first, every packet chooses the next node randomly with a forward ant for recording the paths. When the packet reaches the destination, send back a backward ant along the same path as the forward ant passed, and update the routing table of every router on the path (increase the priorities of directions of the router on the path). After a period, the less latency the path is, the more backward ants come back, so the directions of routers on the path will get higher priority due to backward ants. Therefore, more and more packets will follow the way which has less latency dynamically.

In recent years, research of routing algorithms focuses on routing for irregular topologies [2.96], [2.97], routing for hierarchical topologies [2.98], [2.99], fault

tolerant routing [2.100]-[2.102] and 3-D routing[2.103]-[2.105].

## 2.6.2 Switching Matrix (Crossbar) in Routers

The switching matrix (crossbar) in routers provides routing traversal paths between input ports and output ports based on the decisions of the routing controller. The area of a switch matrix is often the dominant area component of an on-chip router as the area is proportional to  $O(p^2w^2)$ , where  $p$  is the number of router ports and  $w$  is the datapath width. There are two major implementations of the switching matrix, a cross-point switch and a MUX-based switch [2.7].

The cross-point switch has pass transistors at each crossing junction of input and output wires. The capacitive loading of input driver is the junction capacitance of pass transistors on input and output wires and the wire capacitance itself. The voltage swing on the output wire is reduced to  $V_{dd}-V_{th}$  as to the threshold voltage drop. The fabric area is determined by the wiring area so that its area cost can be the minimum. However, this design is hard to be synthesized and sensitive to the noise.

The MUX-based switches use multiplexer for each output port. The power consumption, delay and area are all worse than cross-point switching, especially for large input/output. Nevertheless, the power consumption and delay will exponentially increase with the number of I/O ports no whether Mux-based switch or cross-point switch.

The use of dimension-sliced routers in OCINs was proposed to reduce crossbar switch area with minimal performance loss [2.106]. Based on this low-cost router, the cost of OCINs is reduced by partitioning the crossbar, prioritizing packets in flight to simplify arbitration, and reducing the amount of buffers. Additionally, Crossbar partial activation technique as shown in Fig. 2.27 was also proposed to reduce the power

consumption and delay with large input and output [2.48]. This approach divides the switching matrix into multiple segments to reduce the capacitance.

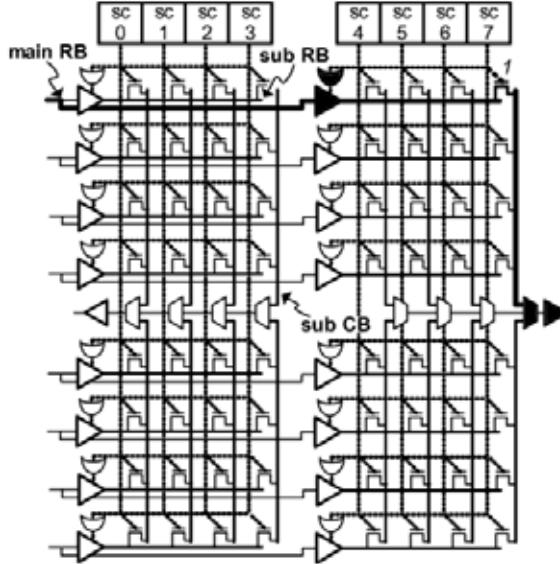


Fig. 2.27 Crossbar partial activation technique. [2.48]

### 2.6.3 Arbitration Unit in Routers

For arbitrating the output conflicts, an arbitration unit is adopted for each output channel. The latency of the arbitration unit increases rapidly with the increasing size of the switching matrix. TDMA [2.107]-[2.109] and round-robin scheduling algorithm [2.110], [2.111] are widely utilized since the characteristics of simple implementation and fairness. Moreover, a pseudo-LRU algorithm was proposed for achieving smaller area and latency than those of the round-robin algorithm [2.112]. Accordingly, an input contention-aware arbitration algorithm was also proposed to achieve high performance by considering the traffic of neighbor nodes [2.112]-[2.114].

### 2.6.4 Queuing Buffer in Routers

Queuing buffer is used in routers or NIs to store un-routed data. Buffer allow for local storage of data that cannot be immediately routed. Unfortunately, queuing

buffers have a high cost in terms of area and power consumption, and thus OCIN implementations strive with limited buffer sizes. If the design lacks sufficient buffer space, on the contrary, the buffers may fill up too fast while over-provisioning of buffers clearly is a waste of scarce area resources. Additionally, insufficient buffer size is a factor to induce head-of-line blocking problems. When the head data of a virtual channel could not be routed and data behind the head data are occupied queuing buffers, head-of-line blocking problems will influence the performance of the network. Some approaches have been proposed to optimize the location and size of buffers. Application-specific buffer space allocation is a system-level buffer planning algorithm to customize the router design [2.115]. Additionally, dynamic virtual channel regulators were proposed to decrease the buffer size without performance overhead [2.113], [2.114], [2.116]-[2.118]. Other approaches were proposed to define the buffer model and buffer constraint in NoC systems [2.119], [2.120]. The details of buffer implementation and buffer analysis will be described in Chapter 4.

## 2.7 Network Interfaces (NIs) for OCINs

NI is usually denoted as the glue logic necessary to adapt PEs to OCIN, implementing the network end-to-end flow control [2.7]. Additionally, NI is designed as a bridge between a process element and a router. A generic template for the NI architecture consists of a front-end sub-module and a back-end sub-module. The network front-end implements a standardized point-to-point protocol allowing core reuse across several OCINs. This solution allows core developers to focus on developing functions of PEs. For the back-end sub-module, data packetization and routing-related functions are essential tasks and tightly interrelated. In fact, source routing implementation requires routing table lookups at the sender NI and integration of routing information in packet headers. The NI also provides data-link layer service

via back-end. Primarily, communication reliability has to be ensured by means of proper error-detection strategies and effective error recovery techniques.

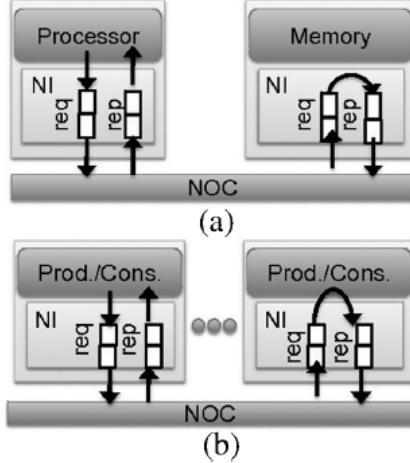


Fig. 2.28 Network interfaces connecting cores to the NoC and possible message dependencies in (a) shared-memory and (b) message-passing communication paradigms. [2.121]

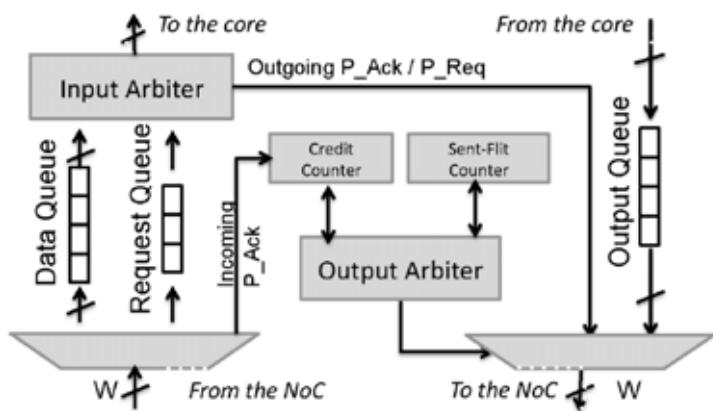


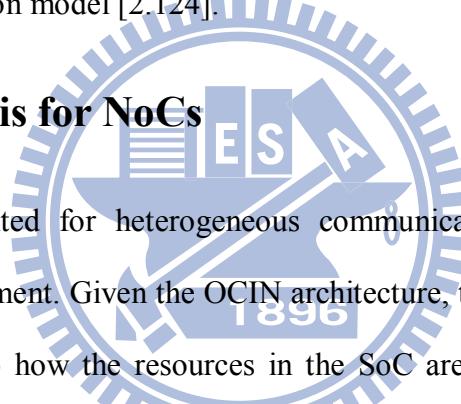
Fig. 2.29 Block diagram of network interface supporting the CTC protocol. [2.121]

NI is a major factor in the performance of OCINs, and implements the communication protocols of OCINs. NI is implemented based on the message dependencies in shared-memory and message-passing communication paradigms as shown in Fig. 2.28. Therefore, a NI micro-architecture was proposed for the connection-then-credit (CTC) flow control protocol among both communication paradigms as presented in Fig. 2.29 [2.121]. CTC is based on the classic end-to-end credit-based flow control protocol but differs from it because it uses a network

interface microarchitecture where a single credit counter and a single input data queue are shared among all possible communications. This architectural simplification reduces the area occupation of the network interfaces and increases their design reuse.

Accordingly, an efficient NI was investigated to offer guaranteed services, shared-memory abstraction and flexible network configuration [2.122]. These two approaches increase design reuse and allow flexible instantiations in different design constraints. In addition to meet these real-time constraints, a communication and configuration controller was developed to manage reconfiguration data-flows in NIs [2.123]. Furthermore, a high-speed NI was proposed to support the serial-link packet-based transmission model [2.124].

## 2.8 Power Analysis for NoCs



OCINs are well suited for heterogeneous communication among PEs in the multi-core SoC environment. Given the OCIN architecture, the system designer needs to make decisions as to how the resources in the SoC are interconnected and how communication in the system is routed. These decisions affect the power and performance of the system being developed. However, total energy reduction is not a sufficient objective to address other aspects of design constraints. The energy for global communication does not scale down with technology shrinking. This makes energy more and more dominant in communications. Reaching those goals will be crucial to the whole semiconductor industry in the next future, in order to face with the escalating range of signal integrity and physical wiring issues, who are making the target IC reliability harder and exponentially expensive to achieve. Indeed, some high level models for functional/performance system simulations Therefore, four layers of power consumption in OCIN architecture should be addressed for power analysis in

NoC as shown in Fig. 2.30, including system level, transaction level, word level, bit level [2.125]. The major difference between system level and transaction level is that the system level contains how to connect the resources and routers and the transaction level is dealing with the hot traffic and the communications between two routers. The bit level model can be divided into four parts as shown in Fig. 2.30, and the difficult part for analyzing is wiring delay. Bit-level power models for OCIN have been utilized in [2.126]-[2.128]. In word level, using the thermal time constant (TTC) can determine the number of flits that are present in the network during this time window and the consequent energy consumption of each of these flits. Flit traversal in the network can be broken down into a sequence of operations buffer reading, switch traversal, external link traversal and buffer writing [2.125], [2.129].

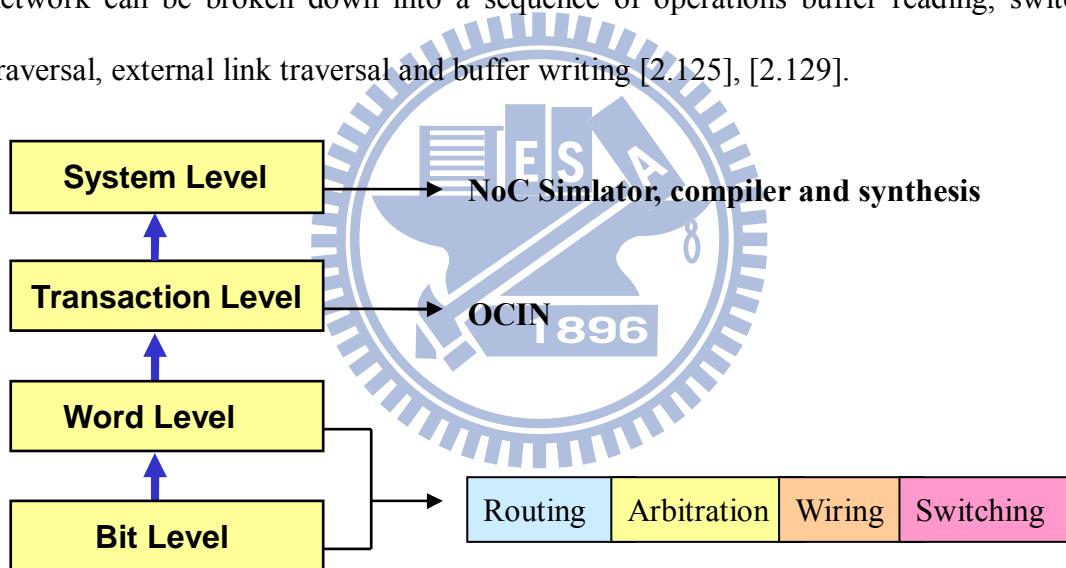


Fig. 2.30 Four levels for power analysis of OCINs

By not providing actual placements of the communicating cores, the power models for network on chip provide the system designer with that degree of freedom and by controlling the bandwidth allocated to the communication and controlling the peak power consumption [2.129],[2.130]. The consequential tradeoff was the decrease in throughput and increased latency. The power models are currently developing a low cost, online power management strategy that would improve the throughput and reduce the latency. Therefore, power analysis and power exploitation were proposed

to develop power model database [2.131]-[2.134] as shown in Fig. 2.31. Moreover, the power model database can be adopted for NoC simulator, NoC compiler and NoC synthesis as shown in Fig. 2.32.

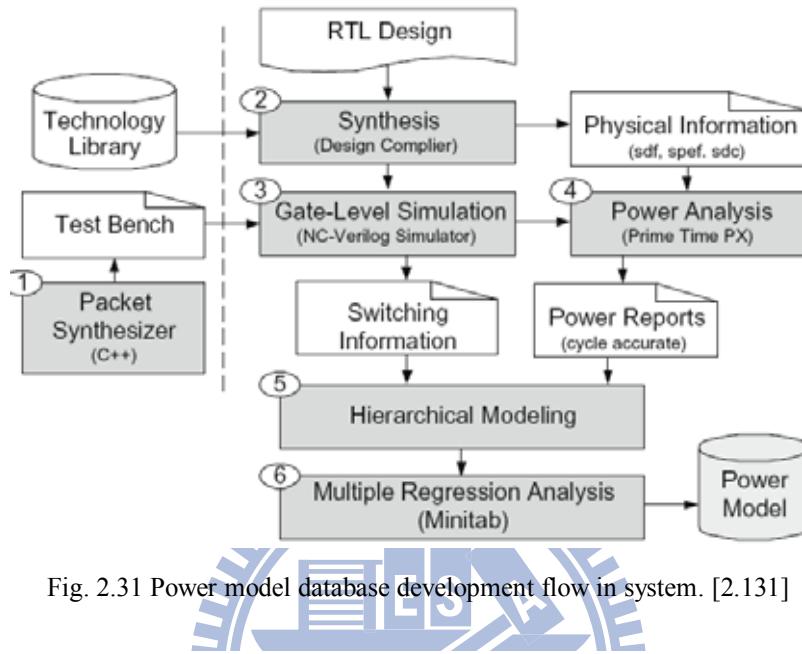


Fig. 2.31 Power model database development flow in system. [2.131]

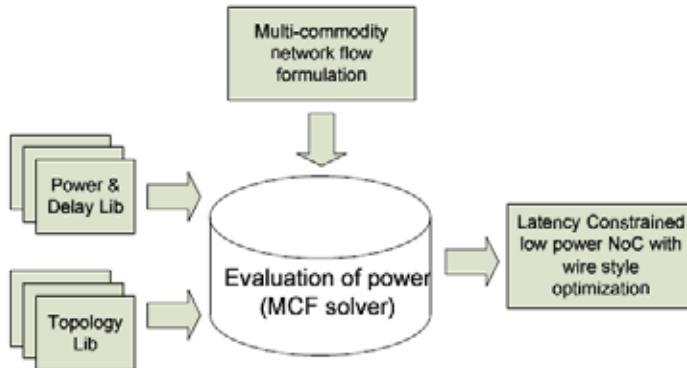


Fig. 2.32 Design Flow of NoC Synthesis. [2.134]

## 2.9 Power Management for NoCs

OCINs have a much higher bandwidth due to multiple concurrent connections. OCINs have regular structure, so the design of wires can be fully optimized and as a result their properties are more predictable. Overall performance and scalability increase since the networking resources are shared. Scheduling of traffic on shared resources prevents latency increases on critical signals. Networking model decouples

the communication layers so that design and synthesis of each layer is simpler and can be done separately. In addition, decoupling enables easier management of power consumption and performance at the level of communicating PEs. Network-centric power management utilizes interaction with the other system cores regarding the power and the quality of service (QoS) needs. Therefore, a methodology for managing power consumption of NOCs was proposed in [2.136] and a power management problem is formulated for the first time using closed-loop control concepts via an estimator and a controller as presented in Fig. 2.33. The estimator is capable of very fast and accurate tracking of changes in the system parameters. The main task of the estimator is to observe the system behavior and based on that to estimate the parameters needed for optimization and control. The quality of power management decisions strongly depends on estimator's ability to track changes of critical parameters at runtime. NOC power management requires estimation of workload characteristics, core parameters, and buffering behavior.

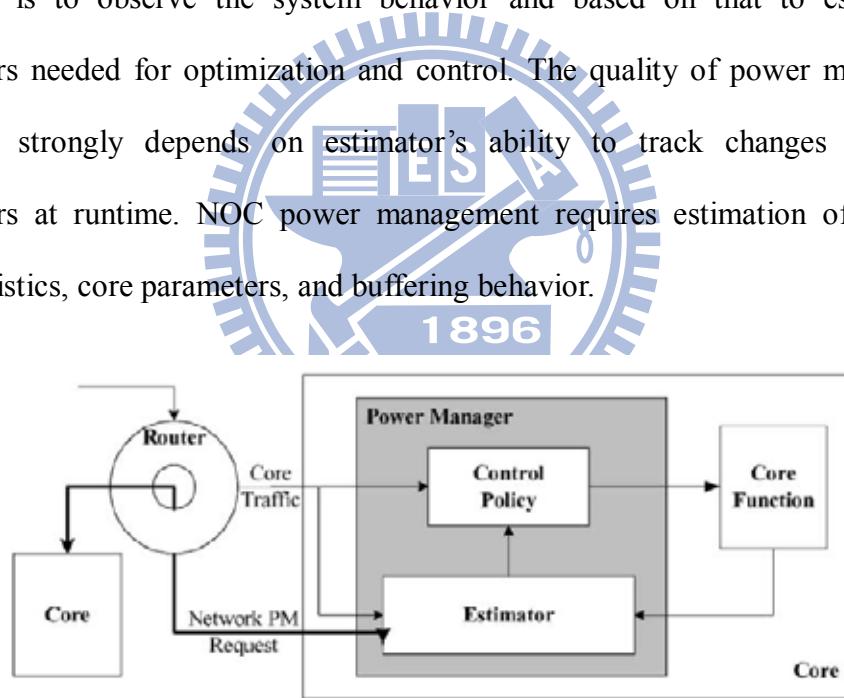


Fig. 2.33 A methodology for managing power consumption of NOCs via the estimator. [2.136]

Given the continuing transistor shrinking and component parallelization, run-time monitoring of NoC will become essential as always-optimal design paradigm and has been acknowledged to achieve better power efficiency among other benefits [2.137], [2.138]. It is important to consider NoC power while designing techniques for chip level power management is important. Especially monitoring and actively controlling

the peak power consumption is very important from the perspective of chip health; as peak power directly related to the chip temperature. A Scalable Adaptable Peak Power management technique (SAPP) has been proposed to address the problem [2.139]. DVFS (dynamic voltage and frequency scaling), usually reduces both the power and energy with lowered supply voltage and frequency. However, randomly controlled DVFS may result in network congestion, which will result in highly increased latency so that the energy-latency product rises significantly, and the static power may increase as a result of long queuing in the buffers [2.140], [2.141].

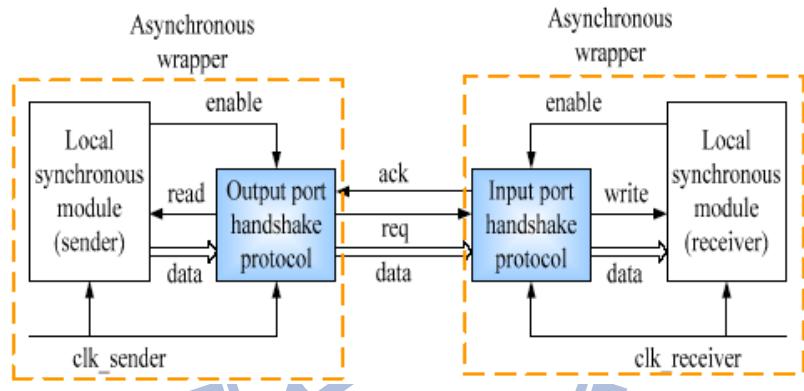


Fig. 2.34 Point-to-point GALS architecture. [2.142]

While scaling voltage and frequency, signal synchronization becomes one of the design challenge. Therefore, GALS design can provide data synchronization for varying frequencies of frequency islands. In GALS NoC, each PE is a synchronous block which has its own local clock called locally synchronous module (LS module) as shown in Fig. 2.34 [2.142]. The communication protocol between PEs is asynchronous that makes use of request and acknowledge signals for the handshaking, and is controlled by port controller. The purpose of a wrapper is to communicate with other LS modules, which is composed of a LS module, a local clock generator and a port controller. The main advantage of GALS is that no additional hardware for clock phase detection is needed. In this GALS NoC, however, the global wires will span multiple clock domains, and synchronization failures between different clock domains

will be rare but unavoidable events. The other main drawback of GALS systems is because of the handshake protocol, the high latency issue will degrade the average data throughput compared with synchronous based design.

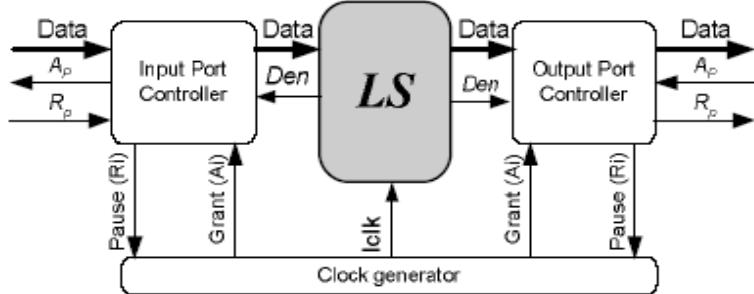


Fig. 2.35 GALS systems based on plausible clocking. [2.143]

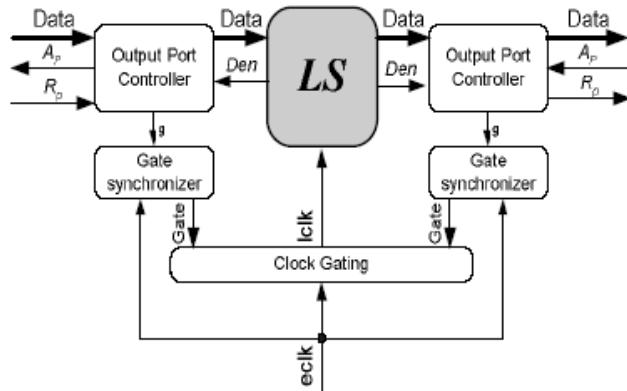


Fig. 2.36 GALS systems based on clock gating. [2.143]

Most of the proposed on-chip clock generators for GALS systems are based on plausible clocking which is based on on-chip ring oscillators as presented in Fig. 2.35 [2.143]. Pausible clock generator is the traditional type of the clock controller. The generator generates the local clock for the LS module and the clock will be stretched or paused by the control signal. Other methods for generating local clock pulse are introduced that is based on clock gating as shown in Fig. 2.36 [2.144]-[2.146]. Moreover, synchronizer or MUTEX cannot be absent for clock gating technique for mixed clock system [2.147]-[2.150]. A synchronization scheme for clock-based GALS wrapper has been proposed, which is called locally delayed latching (LDL) synchronization [2.145]. This method allows for GALS inter-modular communication

and synchronization without the need for clock pausing as shown in Fig. 2.37. In the locally delay latching (LDL) scheme, the clock of the LS module is never stopped. The asynchronous controller controls both the latch and  $Y_1$ , and the local clock signal of the module  $Y$  is uninterrupted. A port request is accepted only during the low phase of  $Y$ , latching the incoming data ( $L^+$ ) and delaying  $Y_1^+$  when needed. When there is a conflict between port and clock ( $R2^+$  and  $Y^+$ ), the MUTEX element will arbitrate them and decide which will be granted. In the case of  $R2^+$  wins over  $Y^+$ , the asynchronous controller is granted ( $R3^+$ ). The controller employs an latch matched delay  $Do \rightarrow Di$  to open the latch and  $L^+$  sets *valid* to logic high, indicating a new ready data inside the data latch. Once latching is completed, the controller deasserts  $R2^+$  and along with  $valid^+$  releases the MUTEX. The next incoming data transfer request is blocked by C-element until the data is sampled by REG1. *Valid* is released with the next rising edge of  $Y_1$ , also enabling the arbitration of next data transfer request. Thus there is no clock pausing for local clock, and the metastability failure caused by clock pausing could be avoided successfully.

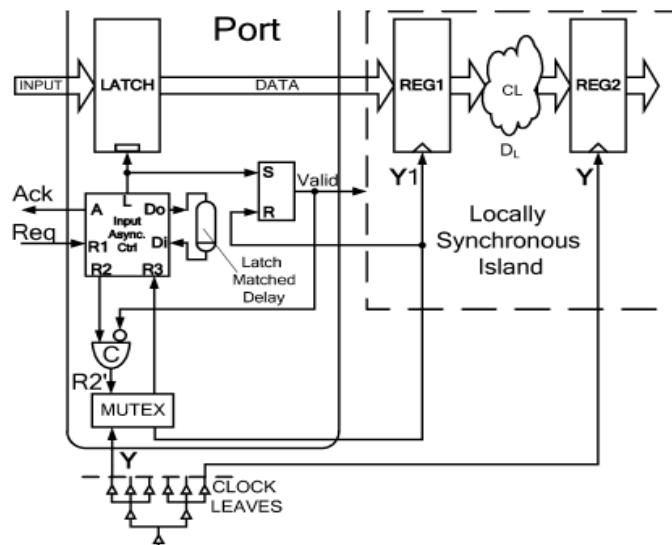


Fig. 2.37 Locally delayed latching (LDL) synchronization. [2.145]

Another design issue for GALS NoCs is the interface between wrappers, such as

mixed-timing FIFO and self-timed ring. Low latency mixed-timing FIFO interface designs for mixed-timing systems have been proposed [2.151]-[2.153]. First, they have low latency, which means the delay time from the first sent data to the first received data. Second, the depth of the FIFO and the width of data bus are scalable with few circuit modifications. A self-timed ring architecture used for GALS multi-point communication was proposed as shown in Fig. 2.38 [2.154]. shows the bypass architecture of the self-timed ring, the ring transceiver in the gray part of the figure is composed of two parts: a *router* that decides where the incoming data packet has to go and an *arbiter* that decides which request to pass (incoming request from the preceding ring transceiver, or request from the host circuitry that wants to feed a data packet into a ring).

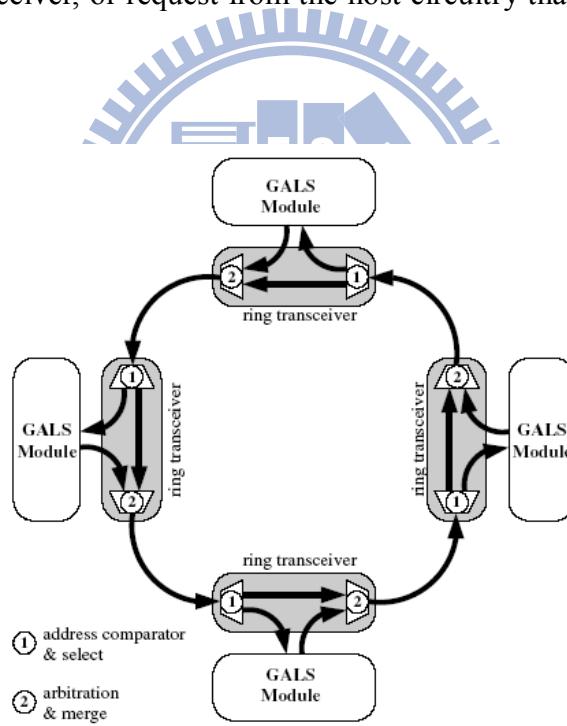


Fig. 2.38 Bypass architecture of the self-timed ring. [2.154]

In recent years, GALS NoCs are implemented with voltage islands to achieve energy optimization via optimal voltage-frequency island partition [2.155], [2.156]. NoC architecture combined with a GALS paradigm is a natural enabler for DVFS mechanisms. Therefore, asynchronous power aware and adaptive NoCs have been proposed via GALS technique with voltage islands [2.157]-[2.159]. Fig. 2.39 presents

the architecture of GALS NoC unit in a voltage island via  $V_{dd}$  hopping technique. The Local Power Manager (LPM) is in charge of handling the unit's power modes and integrated into NI. This manager contains a set of programmable registers, which can be programmed through the NoC, to define the unit power mode, to configure the programmable delay line, and to configure and control the power supply unit. Additionally, Synchronization and communication between the NoC router and the synchronous units is done using a pausable clock mechanism called SAS (Synchronous-to-Asynchronous and Asynchronous-to-Synchronous interfaces).

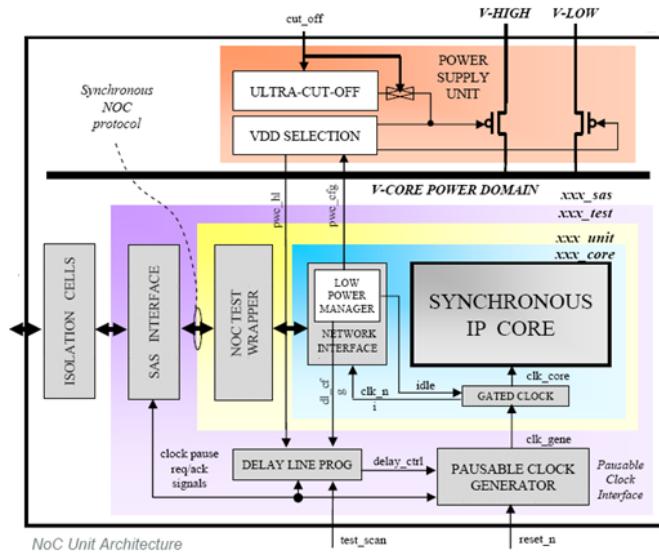
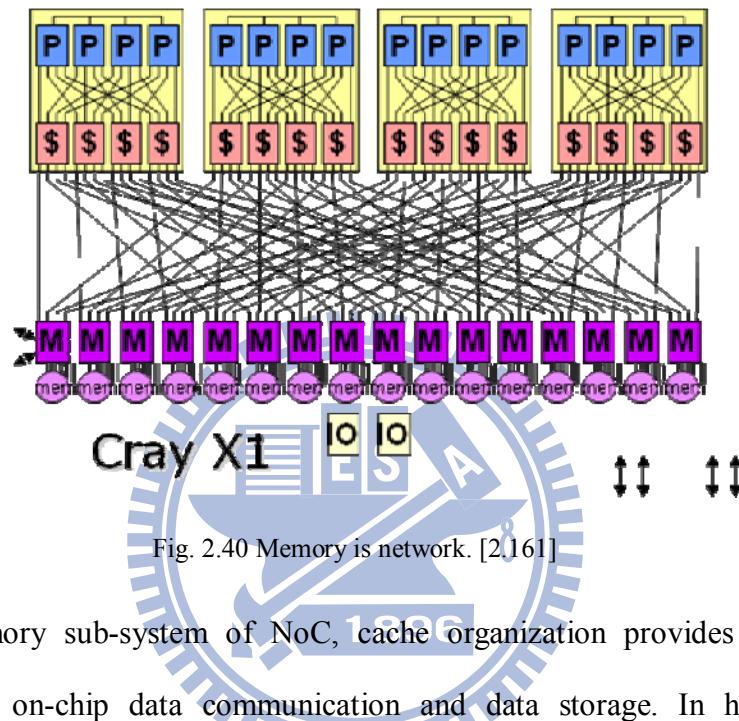


Fig. 2.39 Architecture of GALS NoC unit in a voltage island. [2.157]

## 2.10 Memory is Network!!!

With chip capacities going well beyond the billion transistor mark, on one hand large amounts of the die are occupied by memory resources and on the other hand many complex applications being mapped to these chips are also memory-intensive. In such instances, memories dominate all the axes of traditional design constraints, including, but not limited to performance, area (cost), and power/energy. All of these trends make the case for a memory-aware NoC design methodology [2.160]. NoC provides a parallel computation to enhance system performance. However, the system

performance is still bottlenecked by memory sub-system, and the network is just a path to PEs and memory (on-chip and off-chip). Therefore, memory is network from software's view as shown in [2.161]. According, a memory-centric NoC has been proposed to manage producer-consumer data transactions between the tasks in the case of task-level pipelines [2.162].



In the memory sub-system of NoC, cache organization provides high access bandwidth for on-chip data communication and data storage. In heterogeneous multi-core SoC designs, reconfigurable cache techniques can improve the overall performance via varying memory accesses of PEs during runtime. The best configuration of the cache on a system can be distinct from different application characteristics and design constraints [2.163]. Since no cache organization can fulfill the requirements of all applications [2.164], one way to overcome this problem is to create reconfiguration capabilities in the cache. Reconfigurable caches require some additional mechanisms that enable the on-chip SRAM cache to be dynamically partitioned and reused for other PEs. The aspects of the cache organization can be categorized according to different partitioning method, data consistency process, reconfiguration policy and the reconfigurable cache level [2.165].

## 2.10.1 Cache Partition Methods

In order to resize the cache size, the SRAM partition mechanism is a critical challenge to design a reconfigurable cache. Several partition methods have been presented for reconfigurable caches, including associativity-based partitioning, overlapped wide-tag partitioning and molecular-based partitioning.

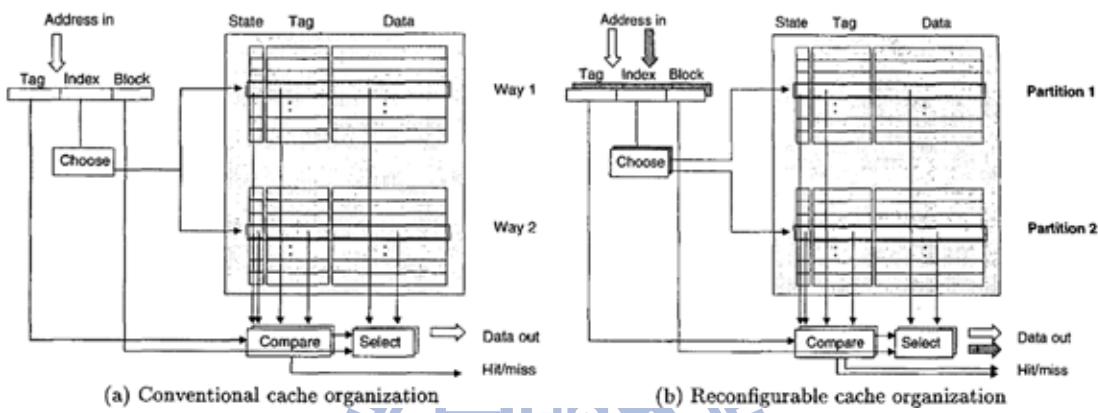


Fig. 2.41 Associativity-based partitioning organization for reconfigurable caches. [2.165]

The associativity-based partitioning divides the reconfigurable cache into partitions at the granularity of ways in the traditional cache [2.165]. Fig. 2.41 presents the associativity-based partitioning cache. This partitioning approach has several advantages. First, the organization only requires few changes to the current set-associative cache organization. Second, different requests which address different partitions can be isolated from each other. However, the drawback of this organization is that the number and granularity of the partitions are limited by the associativity of the cache. Additionally, a selective cache ways method for on-demand cache resource allocation was proposed in [2.166]. This technique disables a subset of the ways in the set associative cache to reduce energy consumption. Moreover, a highly configurable cache architecture for embedded systems was proposed via a way concatenation technique [2.167]. The basic principle of the way concatenation technique is also based on the associativity-based partitioning. Hence, the configurable cache can be

configured by software to be direct-mapped, two-way or four-way set associative.

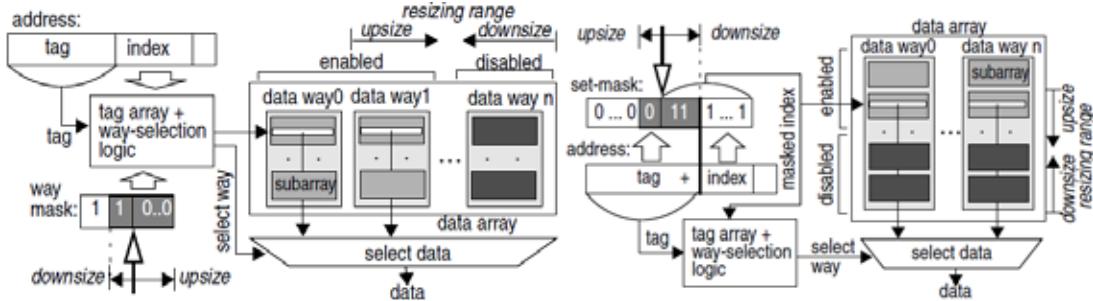


Fig. 2.42 A selective-ways organization and a selective-sets organization. [2.169]

The overlapped wide-tag partitioning increases the tag array bit size to support the maximum tag bit variation with various partition sizes [2.165]. The size of partition can potentially be any size. However, the size would be limited to be powers of two to have simpler implementation generally. The main drawback of this partitioning is that the data in all blocks are required to be flushed while the resizing occurs because the mapping of the address has been changed. Accordingly, an I-cache design has been proposed that the cache size can dynamically be changed, and the cache partitioning method of resizing is similar to the overlapped wide-tag partitioning [2.168]. Moreover, a hybrid selective-sets-and-ways cache organization was proposed to enhance the configuration flexibility [2.169]. Fig. 2.42 shows the basic structures of selective-ways and selective-sets resizable caches. In addition, a heterogeneous cache for QoS was presented via the set partitioning technique [2.170].

In many partitioning approaches, the cache SRAM is divided into several individual sub-caches and categorized as molecular-based partitioning. The separated caches can dynamically be reorganized according to different application requirements. Molecular Caches composed of many small and reconfigurable building blocks has been proposed and called Molecules [2.171]. The design can dynamically adjust the configuration of the cache capacity, set-associativity, and line size. The

cache accessed by a processor is an aggregation of molecules. The Molecular caches support selective enablement of molecules according to different application requirements so that the dynamic power dissipation can be reduced. Fig. 2.43 shows the cache access method. Each molecule is configured with the Application Space Identifier (ASID) which uniquely identifies a running application. Before any cache operation is performed on the molecules, an ASID match is performed to see if the molecule is eligible to perform the operation.

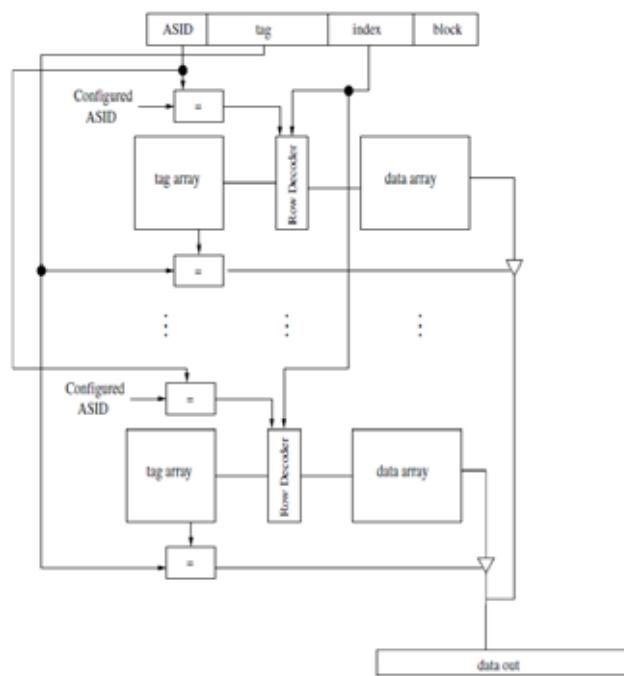


Fig. 2.43 Cache access method in Molecules. [2.171]

A bank-aware dynamic cache partitioning for multi-core architectures was proposed in [2.172] via molecular-based partitioning. A typical allocation in the bank-aware dynamic cache partitioning is as shown in Fig. 2.44. According to different memory resource requirements, the L2 cache banks are separated into eight parts for eight cores. In addition to the above two reconfigurable cache designs, the sub-caches can also be implemented as heterogeneous caches via a combination of overlapped wide-tag partition and molecular-base partition [2.170].

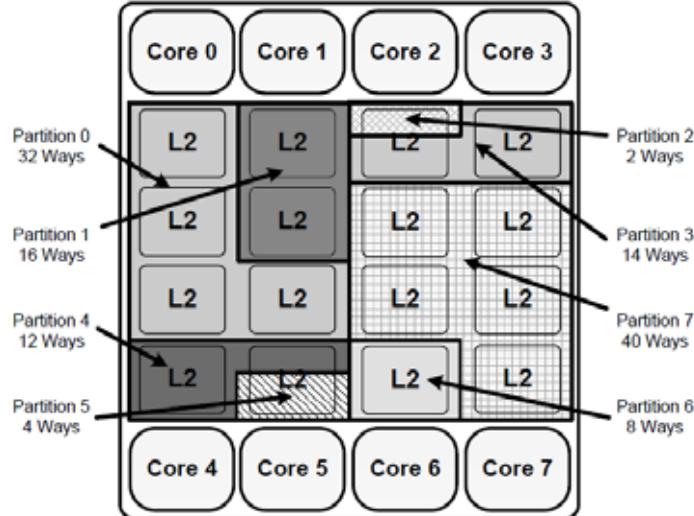


Fig. 2.44 An example of typical CMP cache partitioning. [2.172]

### 2.10.2 Data Consistency in Reconfigurable Cache

Another critical design challenge is data consistency after resizing the cache. Data consistent mechanisms for reconfigurable are required to ensure that the data belonging to a particular processor element resides only in the partition associated with that particular activity [2.165]. Two common approaches are generally utilized for the data consistency, namely cache scrubbing and lazy transitioning.

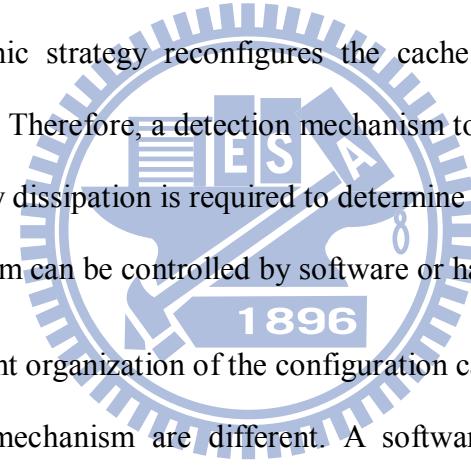
During the reconfiguration process, cache scrubbing scheme moves all valid data to the new partition blocks or lower levels of memory that requires examining all the locations of the cache to check their validity [2.165]. Hence, cache-scrubbing induces huge power overhead since great amount of data accesses. However, the overhead can be ignored if the reconfiguration is infrequent.

When the reconfiguration occurs frequently, lazy transitioning scheme is adopted. Based on the lazy transitioning scheme, the data is lazily moved into the correct partition blocks until the data is accessed [2.166]. Therefore, additional cache-line information is required to indicate the use of the corresponding cache line.

Additionally, if a miss occurs in the appropriate partition, other partitions should be checked because the data may laze in other partitions. The lazy transitioning scheme can avoid high overhead without moving large amounts of data during the reconfiguration. However, this scheme increases the contention between the partition blocks.

### 2.10.3 Reconfiguration Policy and Detection

Detection mechanisms and reconfiguration policies determine the reconfiguration of the reconfigurable cache. The cache reconfiguring strategy can be static or dynamic. In the static strategy, the cache resizing is determined prior to the execution. On the other hand, the dynamic strategy reconfigures the cache organization during the runtime of applications. Therefore, a detection mechanism to dynamically monitor the performance and energy dissipation is required to determine the reconfiguration types. The detection mechanism can be controlled by software or hardware.



According to different organization of the configuration caches, the reconfiguration policy and detection mechanism are different. A software-visible register, called Cache Way Select Register (SWSR), is adopted to enable/disable the particular ways [2.166]. The SWSR was written and read by specific pre-defined instructions. The Performance Degradation Threshold (PDT) measured the performance degradation relative to a cache with all ways enabled. The Mattson's stack distance algorithm and the concept of Marginal Utility, which originated from economic theory, are used to be the assignment policy in bank-aware cache partitioning [2.172]. A basic Block Vectors (BBV)-based tuning technique has been proposed to trace the loop characteristics of the program in the runtime that dynamically learns the configuration type by holding the previous CPI value [2.163].

The aspects of the configurable cache can be categorized according to different partitioning method, data consistency process, reconfiguration policy and the reconfigurable cache level. Table 2.1 lists the summaries of the reconfigurable caches.

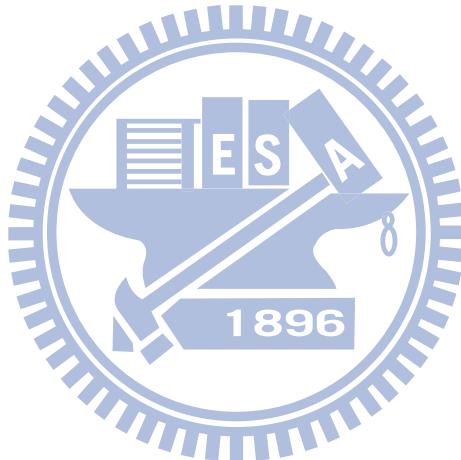
Table 2.1 Related work of reconfigurable caches

	Partitioning mechanism	Data consistency	Detection mechanism	Reconfigurable cache level	Application
[2.163]	Molecular-based	Cache scrubbing	Hardware controlled; Dynamic strategy	L2	General purpose
[2.165]	Associativity-based	Cache scrubbing	Software controlled	L1	Media processing
[2.166]	Associativity-based	Lazy transitioning	Software controlled; Dynamic strategy	L1	General purpose
[2.167]	Associativity-based	N/A	Software controlled; Static strategy	L1	Embedded System
[2.168]	Overlapped wide-tag	Cache scrubbing	Software controlled, Static/Dynamic strategy	L1 I-cache	General purpose
[2.169]	Associativity-based Overlapped wide-tag	Cache scrubbing	Software controlled Static/Dynamic strategy	L1	General purpose
[2.170]	Overlapped wide-tag Molecular-based	Cache scrubbing	Software controlled dynamic strategy	Shared cache	Multi-core Network-intensive applications.
[2.171]	Molecular-based	Cache scrubbing	Software controlled; Dynamic strategy	L2	General purpose multi-core
[2.172]	Molecular-based	N/A	Software controlled Dynamic strategy	L2	General purpose multi-core

## 2.11 Summary

Based on the survey of on-chip data communication, previous works can build a comprehensive on-chip data communication platform. However, they did not emphasize the connection between memory access and data communication. Additionally, energy-efficiency is also a critical design issue for multi-core SoCs. Therefore, in this dissertation, an energy-efficient memory-centric on-chip data communication platform is proposed to deal with the increasing data communication

and data storage for heterogeneous multi-core SoC designs that consists of a memory-centric OCIN and an on-demand memory sub-system. The memory-centric OCIN provides the micro-architecture for data communication based on the building blocks, including link wires, routers and NIs. In this dissertation, all building blocks are analyzed and developed to realize energy-efficient multi-core SoCs. Additionally, the on-demand memory sub-system enhances memory bandwidth and reduces the total execution time of the whole system via the centralized MMU and private MMUs. Moreover, the NI provides a bridge between the on-demand memory sub-system, memory-centric OCIN and heterogeneous PEs.



# ***Chapter 3: Energy-Efficient and Reliable Channels for OCINs***

In this chapter, energy-efficient and reliable channels are provided for on-chip interconnection networks (OCINs) using a self-calibrated voltage scaling technique with self-corrected green (SCG) coding scheme. This self-calibrated low-power coding and voltage scaling technique increases reliability and reduces energy consumption simultaneously. The SCG coding is a joint bus and error correction coding scheme that provides a reliable mechanism for channels. In addition, it achieves a significant reduction in energy consumption via a joint triplication bus power model for crosstalk avoidance. Based on SCG coding scheme, the proposed self-calibrated voltage scaling technique adjusts voltage swing for energy reduction. Furthermore, this technique tolerates timing variations. Based on UMC 65nm CMOS technology, the proposed channels reduces energy consumption by nearly 28.3% compared with that for un-coded channels at the lowest voltage. This approach makes the channels of OCINs tolerant of transient malfunctions and realizes energy efficiency.

## **3.1 Background**

OCINs provide the building blocks and the micro-architecture for NoCs [3.1], [3.2]. However, some physical effects in nano-scale technology unfortunately degrade the performance and reliability of OCINs. Moreover, channels in OCINs dominate the overall power consumption [3.3], [3.4]. On-chip physical interconnections will comprise a limiting factor for performance and energy consumption. For on-chip

interconnections, three critical issues, delay, power and reliability must be addressed. For the delay issue, propagation decreased by coupling capacitances. For long global lines, discharging large capacitances takes considerable time. For the power issue, power dissipation increases due to both parasitic and coupling capacitances. Finally, the reliability issue for on-chip interconnections will be degraded due to noise. In advanced technologies, circuits and interconnects degrade further due to noise with decreasing operating voltages. Furthermore, increasing coupling noise, the soft-error rate, and bouncing noise also decrease the reliability of circuits. Thus, self-calibrated circuitry has become essential for near-future interconnection architecture designs.

To achieve low latency and reliable and low energy on-chip communication, energy efficiency is the primary challenge for current OCIN designs with nano-scale effects. First, coupling capacitance increases significantly in nano-scale technology. Second, decreasing operating voltage makes the interconnection susceptible to noise increasingly. Due to crosstalk noise, the coupling effect not only aggravates the power-delay metrics but also deteriorates the signal integrity. Many techniques have been developed to reduce the coupling capacitance effect using bus encoding schemes [3.5]-[3.13]. Bus encoding is an elegant and effective technique for eliminating the crosstalk effect, and provides a reliability bound for on-chip interconnects. Moreover, in order to provide a reliability bound for on-chip interconnects, forward error correction (FEC) and automatic repeat request (ARQ) techniques are widely used in NoC [3.1], [3.14]. Additionally, a joint error correction coding and bus coding technique is an effective solution to resolve delay, power and reliability. Encoding schemes for low power and reliability issues were proposed in [3.15]-[3.20]. The designers increased reliability for on-chip interconnections. Moreover, robust self-calibrating transmission schemes were proposed in [3.14], [3.21]-[3.23], which

examined some physical properties of on-chip interconnects, with the goal of achieving fast, reliable and low-energy communication.

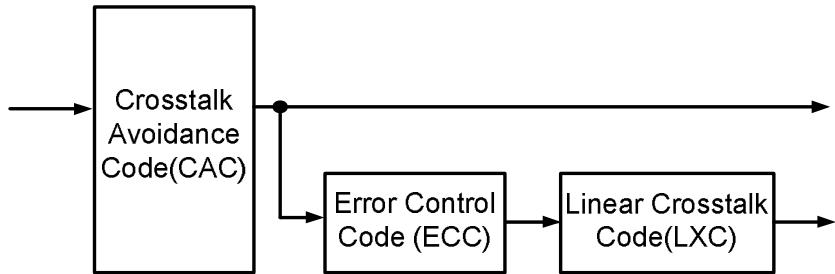


Fig. 3.1 A unified framework for joint crosstalk avoidance code and error correction code.

Incorporating of different coding schemes has been investigated to increase system reliability and to reduce energy dissipation. The crosstalk avoidance codes incorporate with forward error correction coding is a solution to provide the low power and reliable on-chip interconnection. Therefore, Duplicate-add-parity (DAP) [3.15], Modified Dual Rail (MDR) [3.18], Boundary Shift Code (BSC) [3.17], [3.18] and Hamming Codes [3.15] are the forward error correction coding to increase the reliability of interconnections. A unified framework of coding with crosstalk avoidance codes (CAC), error control codes (ECC) and linear crosstalk codes (LXC) was proposed in [3.15], [3.16]. It provides practical codes to solve delay, power, and reliability problems jointly as shown in Fig. 3.1. CAC avoids specific code patterns or code transitions to reduce delay and power consumption by decreasing crosstalk effect. ECC is able to detect and correct the error bits. However, the parity bits of CAC cannot be modified. In order to reduce the coupling effect of parity bits, LXC is applied without destroying the parity bits. Other approaches are based on the unified framework to improve the ability of error correction and to address signal integrity in OCINs [3.15]-[3.20].

CACs are designed to improve the signal integrity and to reduce the coupling

effect. The purpose of CAC is to reduce the worst-case switching patterns, which are forbidden overlap condition (FOC), forbidden transition condition (FTC) and forbidden pattern condition (FPC) [3.15]. FOC represents a codeword transition from 010 to 101 or from 101 to 010. In addition, FTC represents a codeword transition from 01 to 10 or from 10 to 01, and FPC represents a codeword having 010 or 101 patterns. In order to reduce or avoid the worst-case switching patterns, many coding schemes are proposed to be directed against the three conditions [3.20]. Forbidden Overlap Code provides a 5-bit codeword for a 4-bit dataword to eliminate FOC. And Forbidden Pattern Code is also a 5-bit codeword for a 4-bit dataword to avoid FPC in codeword. Additionally, Forbidden Transition Codes provides a 4-bit codeword for a 3-bit dataword to prevent FTC. However, these three coding schemes do not satisfy the forbidden adjacent boundary pattern condition, which is defined as two adjacent bit boundaries in the codes cannot both be of 01-type and 10-type. Hence, One Lambda Codes is proposed not only to avoid FTC and FPC but also to satisfy the forbidden adjacent boundary pattern condition [3.20]. However, it needs an 8-bit codeword to transfer a 4-bit dataword.

Joint coding schemes based on the unified framework as shown in Fig. 1 provide better communication performance. However, these schemes just combine different kinds of codes directly, since the intrinsic qualities of CACs and ECCs are mutually exclusive, except for duplicating codes (DAP, MDR and BSC) [3.15], [3.18]. In DAP coding, nevertheless, the critical path of the priority bit is much longer than others. Moreover, CAC must be a code that does not modify the parity bits in any way as decoding of ECC has to occur before any other decoding in the receiver. In order to reduce the coupling effect of the parity bits, the linear crosstalk code could be applied without destroying the parity bits.

## 3.2 Self-Calibrated Low Power and Energy-Efficient Channel Design

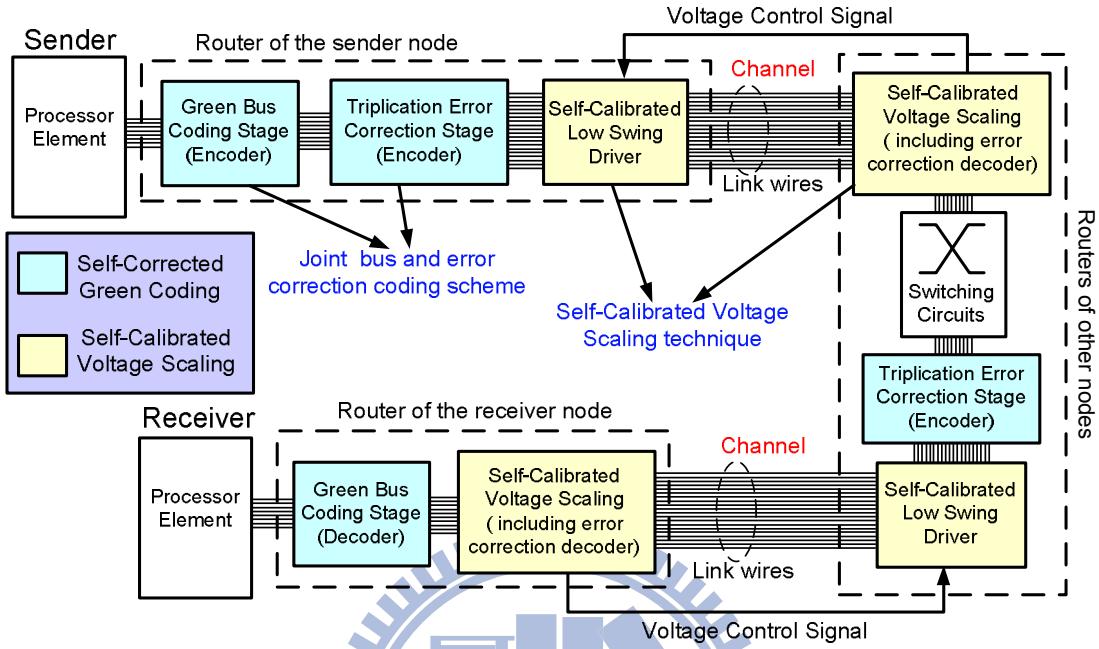


Fig. 3.2 Self-calibrated energy-efficient and reliable channels for on-chip interconnection networks with self-corrected green (SCG) coding scheme and self-calibrated voltage scaling technique.

The self-calibrated energy-efficient and reliable channels are developed using a self-calibrated voltage scaling technique and a joint bus/error correction coding scheme, which is called the SCG coding scheme. Fig. 3.2 shows the block diagrams of the proposed channels for OCINs. The SCG coding scheme reduces coupling effects and has a rapid correction ability that reduces the physical transfer unit size in routers. The self-calibrated voltage scaling technique achieves the optimal operating voltage for link wires in channels according to the SCG coding scheme. Additionally, the proposed technique overcomes increasing variation in advanced technologies and facilitates the energy-efficient on-chip data communication. Therefore, the proposed self-calibrated low power coding and voltage scaling realizes energy-efficient and reliable channels for OCINs.

The SCG coding scheme is a joint bus and error correction coding scheme that

provides low energy and high reliability channels for OCINs. The SCG coding scheme is constructed in two stages, the green bus coding stage and the triplication error correction coding stage. In routers, an un-decoded code increases the area and energy dissipation of switching circuits by large physical transfer unit sizes. Therefore, the error correction code should be decoded in routers to reduce power dissipation and the area of switching circuits and buffers. The triplication error correction coding stage achieves rapid correction to reduce the physical transfer unit size in routers via a self-corrected mechanism at the bit level. To efficiently reduce the coupling effect, the green bus coding stage is developed using the joint triplication bus power model, which depends upon the characteristics of triplication error correction coding. The SCG coding can avoid the FOC and FPC, and reduce the FTC to achieve the power saving of channels. The bit-width in the self-calibrated low power coding and voltage scaling varies. The green bus coding encodes packets in accordance with a 4-to-5 codec. To increase the reliability of channels, the triplication error correction stage increases bit-width from  $k$ -bit to  $3k$ -bit. Although the SCG coding increases link wires in channels, on-chip wires are cheap and plentiful with the increasing metal layers in advanced technologies [29, 30].

Designers can tradeoff between power consumption and reliability by reducing the operating voltage as the error correction coding increases the reliability of channels. Therefore, the operating voltage of the link wires in channels is adjusted according to the SCG coding scheme using a self-calibrated voltage-scaling technique. This technique detects error conditions of channels in the triplication error correction stage, and thus feeds the control signals back to the low swing drivers and adjusts the operating voltage of the link wires. The self-calibrated voltage scaling technique determines the optimal operating point to trade off between energy consumption and

reliability. The SCG coding scheme and self-calibrated voltage scaling technique are described in the following sections.

### 3.3 Self-Corrected Green (SGC) Coding Scheme

This section describes the SCG coding scheme, a joint bus and error correction coding scheme. This proposed scheme generates low energy and reliable channels for advanced technologies. The SCG coding scheme is constructed via two stages, the green bus coding stage and triplication error correction coding stage. The green bus coding has the advantages of shorter delay for error correction coding, greater energy reduction and smaller area than other approaches. The green bus coding is developed using the joint triplication bus power model to achieve additional energy reductions for triplication error correction coding.

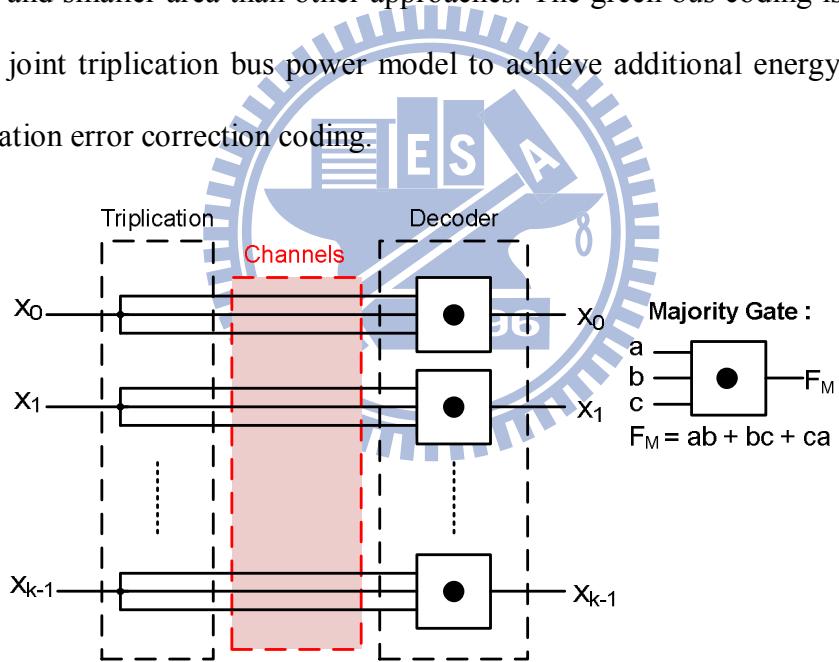


Fig. 3.3 Triplication error correction stage of SCG coding scheme.

#### 3.3.1 Triplication Error Correction Stage

The triplication error correction coding scheme as shown in Fig. 3.3 is a single error correcting code by triplicating each bit. Based on information theory, a code set with a hamming distance of  $h$  has an  $h-1$  error-detect ability and a  $[(h-1)/2]$  error-correction ability. For triplication error correction coding, the hamming distance

of each bit is 3. Therefore, each bit can be corrected individually when no more than one error bit exist in the three triplicated bits, which are defined as a triplication set. The error bit can be corrected by a majority gate. Fig. 3.3 also shows the function of the majority gate. Compared with other error correction mechanisms, the critical delay of the decoder is a constant delay of a majority gate and significantly smaller than that of other approaches [3.14]-[3.20]. Restated, the triplication error correction coding has rapid correction ability via self-correction mechanism at the bit level. Therefore, triplication error correction coding is more suitable to OCINs because data can be decoded and encoded in each router using the small delay of triplication correction coding.

Additionally, one advantage of incorporating error correction mechanisms in an OCIN data stream is that the supply voltage of channels can be reduced without compromising the system reliability. Reducing supply voltage,  $V_{dd}$ , increases bit error probability. To simplify error sources, we assume bit error probability,  $\varepsilon$ , is as Eq. (3.1) when a Gaussian distributed noise voltage,  $V_N$ , with variance  $\sigma_N^2$  is added to the signal waveform.

$$\varepsilon = Q\left(\frac{V_{dd}}{2\sigma_n}\right) \quad (3.1)$$

where  $Q(x)$  is given as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{y^2}{2}} dy \quad (3.2)$$

Each triplication set can be error-free if and only if no error transmission exists or just 1-bit error transmission exists. For each triplication set,  $P_{1\text{-bit correct}}$  is given as

$$P_{1\text{-bit correct}} = (1-\varepsilon)^3 + \binom{3}{1} \varepsilon (1-\varepsilon)^2 \quad (3.3)$$

For  $k$ -bits data, transmission is error-free if and only if all  $k$  triplication sets are

correct. Thus,  $P_k$  bits correct is given by

$$P_{k \text{ bits correct}} = \prod_{i=1}^k P_{i\text{-bit correct}} = (1 - 3\epsilon^2 + 2\epsilon^3)^k \quad (3.4)$$

Hence, word-error probability is

$$P_{\text{triplication}} = 1 - (1 - 3\epsilon^2 + 2\epsilon^3)^k \quad (3.5)$$

For a small probability of bit error,  $\epsilon$ , Eq. (3.5) is simplified to

$$P_{\text{triplication}} \approx 3k\epsilon^2 - 2k\epsilon^3 \quad (3.6)$$

By contrast, word-error probability is much smaller than that in the Hamming code and Duplicate-add-parity (DAP) [3.15], [3.16] which are directed to  $k^2\epsilon^2$ . Triplication error correction coding can avoid the FOC and FPC which increase energy dissipation via the coupling effect.

Because error-correction coding increases the reliability of on-chip interconnections, designers can tradeoff between power consumption and reliability by reducing operating voltage. In simplifying the cumulative effect of noise sources, the noise model on interconnects assumes Gaussian distributed noise with voltage  $V_N$  and variance  $\sigma_N^2$  is added to the signal. In addition, we assume errors on different link lines are independent. The bit error probability,  $\epsilon$ , is given in Eqs. (3.1) and (3.2), where  $V_{dd}$  is signal voltage swing. With given the same  $\sigma_N^2$ , bit error probability increasing as signal voltage swing decreases. However, some specific error control/correct coding schemes can decrease signal voltage swing, and guarantee the reliability of interconnections, if and only if the following equation is satisfied:

$$P_{\text{uncode}}(\epsilon) \geq P_{\text{ecc}}(\hat{\epsilon}) \quad (3.7)$$

where  $\varepsilon$  is bit error probability with full swing voltage of 1.0 V, and  $\hat{\varepsilon}$  is bit error probability with a lower swing voltage. To obtain the lowest supply voltage for specific error correction coding under the same level of reliability of the un-coded code, supply voltage can be revised as

$$\hat{V}_{dd} = V_{dd} \frac{Q^{-1}(\hat{\varepsilon})}{Q^{-1}(\varepsilon)}, P_{uncode}(\varepsilon) = P_{ecc}(\hat{\varepsilon}) \quad (3.8)$$

The inverse function of the Gaussian distributed function is also called a probit function  $\Phi(x)$ . The probit function has been proved that the function does not have primary primitive. To solve the problems, this work first approximates the bit error probability by varying voltage swing. By integrating from  $100 - V_{dd}/2$ , the integral range on the x-axis is divided into  $0.0001(v)$  segments, and each segment can produce a trapezoid. The areas of all trapezoids are then summed, which is the approximation of bit error probability. Therefore, the lowest voltage swing for a specific error correction coding that satisfies Eq. (3.8) can be obtained.

When an un-coded code is operated at full swing supply voltage (1.0v), different levels of bit error probability,  $\varepsilon$ , can be obtained by altering the variance of the Gaussian distributed function. Fig. 3.4(a) and Fig. 3.4 (b) show the voltages of specific error correction coding versus different un-coded word error rates with  $k = 8$  and  $k = 32$ , respectively. Factor,  $k$ , is bit-width. If bit error probability of an un-code word,  $\varepsilon$ , is  $10^{-20}$ , the specific voltage of Hamming code [3.15], Duplication-Add-Parity code [3.15], [3.16], joint crosstalk avoidance and triple-error-correction code (JTEC) [3.19] and the proposed SCG code are 0.705V, 0.710V, 0.579V, and 0.696V, respectively. The JTEC code uses a double error correction coding stage to enhance error correction and obtains lower voltages. However, delay and area overheads of the JTEC are much worse than those of other

approaches. Compared to other ECC codes, the proposed SCG code has better characteristics in that the lowest supply voltage increases slowly when the un-coded word error rate increases.

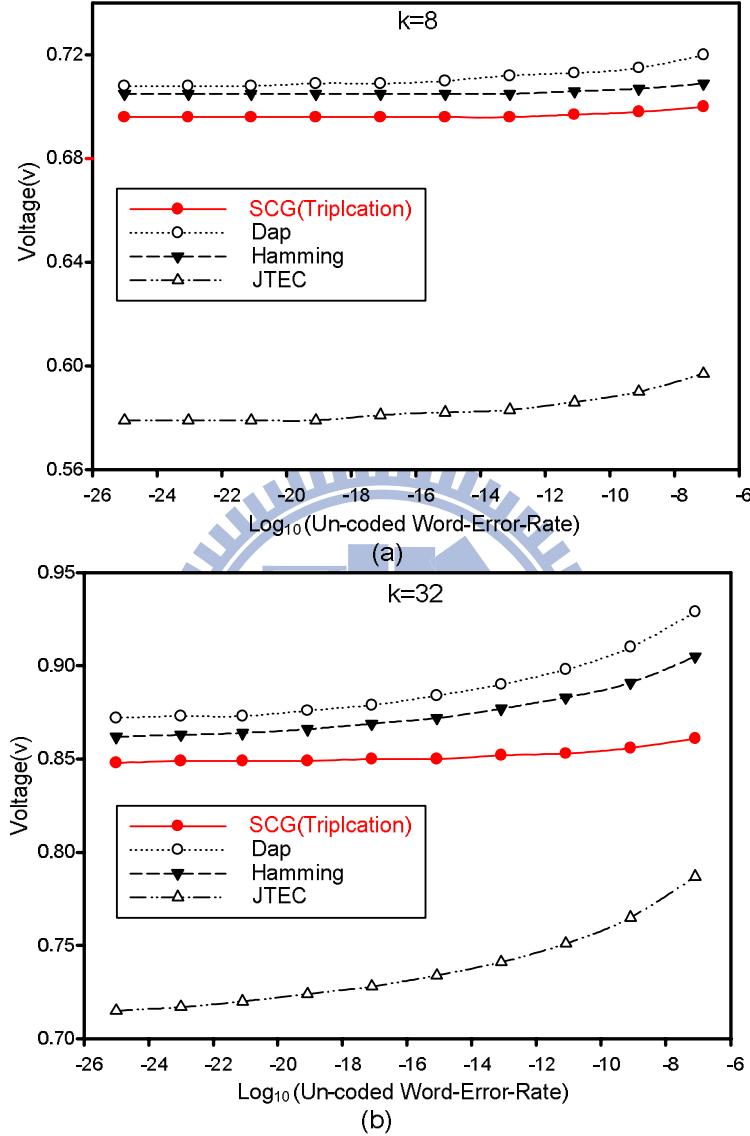


Fig. 3.4 The corresponding voltages of specific error correction coding versus different un-coded word-error- rate with (a)  $k = 8$  (b)  $k = 32$ .

### 3.3.2 Joint Triplcation Bus Power Model

Although triplcation error correction coding can avoid many forbidden conditions, some power-hungry transition patterns cannot be eliminated entirely. These patterns are mainly generated by the FTC and self-switching activity. The FTC can be satisfied

when a bit pattern does not have a transition from 01 to 10 or from 10 to 01. This work modified the RLC cyclic bus model in [3.26] by considering loading capacitances and coupling capacitances. Fig. 3.5(a) shows the modified model with a four-bit bus, where  $C_{11}$  means the loading capacitance of line 1 and the  $C_{12}$  is the coupling capacitance between line 1 and line 2. Moreover, the bus lines are parallel and coplanar. Most of the electrical field is trapped between adjacent lines and the ground. Fig. 3.5(b) shows an approximate bus power model that ignores the parasitic capacitances between nonadjacent lines.

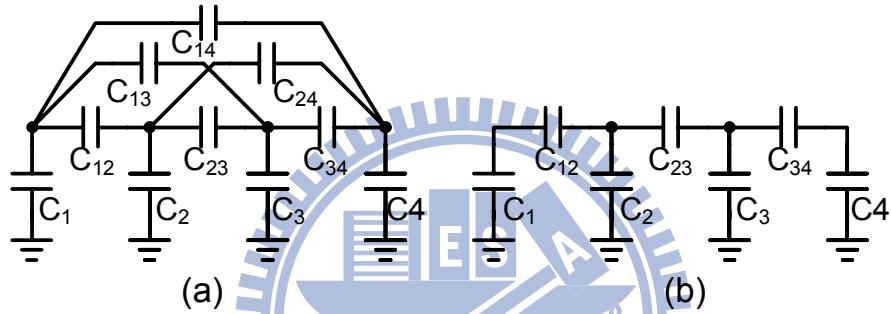


Fig. 3.5 (a) Bus Model for 4 bits (b) The approximate bus power model.

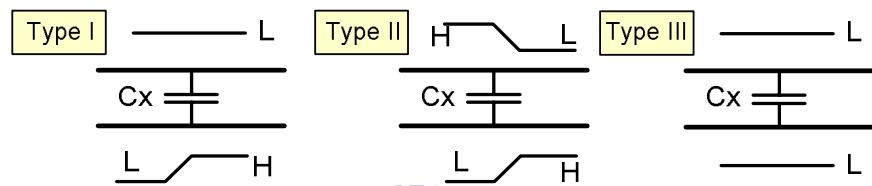
We assume all grounded capacitors have the same value without considering the fringing effect of boundary lines. Because of fringing capacitors are much smaller than loading and coupling capacitors, even for the wide buses. Therefore, this work utilized a joint triplication bus model to implement the bus coding stage to further reduce energy consumption. For a 4-bit triplication bus, the capacitance matrix  $C^t$  can be expressed as:

$$C^t = \begin{bmatrix} 3 + \lambda & -\lambda & 0 & 0 \\ -\lambda & 3 + 2\lambda & -\lambda & 0 \\ 0 & -\lambda & 3 + 2\lambda & -\lambda \\ 0 & 0 & -\lambda & 3 + \lambda \end{bmatrix} C_L, \lambda = \frac{C_x}{C_L} \quad (3.9)$$

The parameter,  $\lambda$ , is defined as the ratio of coupling capacitance,  $C_x$ , to loading capacitance,  $C_L$ . Therefore, the  $\lambda$  parameter depends on the technology, the specific

geometry, the metal layer and bus shielding.  $\lambda$  has some important properties; for example, the parameter  $\lambda$  typically increases with technology scaling. For instance, the value of  $\lambda$  is between 6 and 10, depending on the metal layer for standard 65nm CMOS technology and the minimum distance between wires. The parameter  $\lambda$  should be much large in advanced technologies. Additionally, the coefficient of loading capacitances is 3 for the three triplicated bits.

Static transitions:



Dynamic transitions:

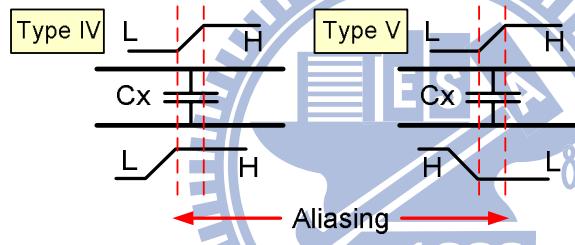


Fig. 3.6 Five transition types for two adjacent wires.

Five transition states exist between two adjacent lines, four of which are described in [3.27]. These five types can be separated into two cases. The first case is static transitions, including type I (single line switching) type II (two lines switching in opposite directions) and type III (no switching or two lines switching in the same direction) as shown in Fig. 3.6. The other case is dynamic transitions which inculde type IV and type V with signal aliasing for type II and type III, respectively. The static transition is defined as two adjacent lines switching at the same time without noise or different delays. The dynamic transition means that the two adjacent lines may be misaligned.

The power consumption formula is shown in Eq. (3.10), where  $E$  and  $P$  are energy

and power density, respectively;  $f$  and  $V$  ( $V_{DD}$ ) are frequency and voltage (voltage supply), respectively.  $B_i$  is the current voltage level (1 or 0) for line  $i$ , and  $B_i^{-1}$  is the previous voltage level for the line  $i$ .

$$E = (V^f)^T C^t (V^f - V^i) \\ P = f * V_{DD}^2 * \sum_i \sum_j C^t \left\{ (B_i - B_i^{-1}) * (B_j - B_j^{-1}) \right\} \quad (3.10)$$

Power density,  $P$ , can be transferred into Eq. (3.11).

$$P = f * C_L * V_{DD}^2 * \left\{ \begin{array}{l} 3(B_1 - B_1^{-1})^2 + 3(B_2 - B_2^{-1})^2 + 3(B_3 - B_3^{-1})^2 \\ + 3(B_4 - B_4^{-1})^2 + \lambda [(B_1 - B_1^{-1}) - (B_2 - B_2^{-1})]^2 \\ + \lambda [(B_2 - B_2^{-1}) - (B_3 - B_3^{-1})]^2 \\ + \lambda [(B_3 - B_3^{-1}) - (B_4 - B_4^{-1})]^2 \end{array} \right\} \quad (3.11)$$

The items in Eq. (3.11) are defined and identified as follows.

$$(B_i - B_i^{-1})^2 = B_i \oplus B_i^{-1} = r_i \\ [(B_i - B_i^{-1}) - (B_j - B_j^{-1})]^2 = r_i \oplus r_j + 4 \times d_{ij} \\ \text{where } d_{ij} = \overline{B}_i B_i^{-1} \overline{B}_j B_j^{-1} \cup B_i \overline{B}_i^{-1} \overline{B}_j B_j^{-1} \quad (3.12)$$

$$P = f \times C_L \times V_{DD}^2 \times \alpha \\ \alpha = 3(r_1 + r_2 + r_3 + r_4) + \lambda(r_1 \oplus r_2 + r_2 \oplus r_3 + r_3 \oplus r_4) \\ + 4\lambda(d_{12} + d_{23} + d_{34}) \quad (3.13)$$

The  $r_i$  means that a switch of line  $i$  exists and is not concerned with the direction of change and adjacent lines. This item,  $r_i$ , only considers loading capacitances. The meaning of  $r_i \oplus r_j$  is that only one line is changing between two lines of  $i$  and  $j$ . Additionally,  $d_{ij}$  indicates that two lines change in opposite directions. Moreover, compared with the other two definitions,  $r_i$  and  $r_i \oplus r_j$ , the voltage difference across the coupling capacitance is double and when squared it factor 4 for  $d_{ij}$ . Using Eq. (3.12), the power formula can be obtained as Eq. (3.13) with the parameter of  $\lambda$ . The term  $\alpha$

is the coefficient of coupling effects and switching activities.

### 3.3.3 Green Bus Coding Stage for Crosstalk Avoidance

The purpose of the green bus coding stage is to minimize the value of  $\alpha$  in Eq. (3.13) by encoding signals when  $\lambda > 2$ . Fig. 3.7 shows design flow of green bus coding. First a triplication capacitance matrix is established using the RLC cyclic model. Then the power formula with coefficient  $\alpha$  is derived, where  $\alpha$  represents the switching factor by considering coupling capacitances. The green bus coding stage only affects coefficient  $\alpha$ . Furthermore, the codeword minimizes the value of  $\alpha$  and maps the codeword to the dataword. Depending on the mapping between the codeword and dataword, the green bus coding stage can be implemented.

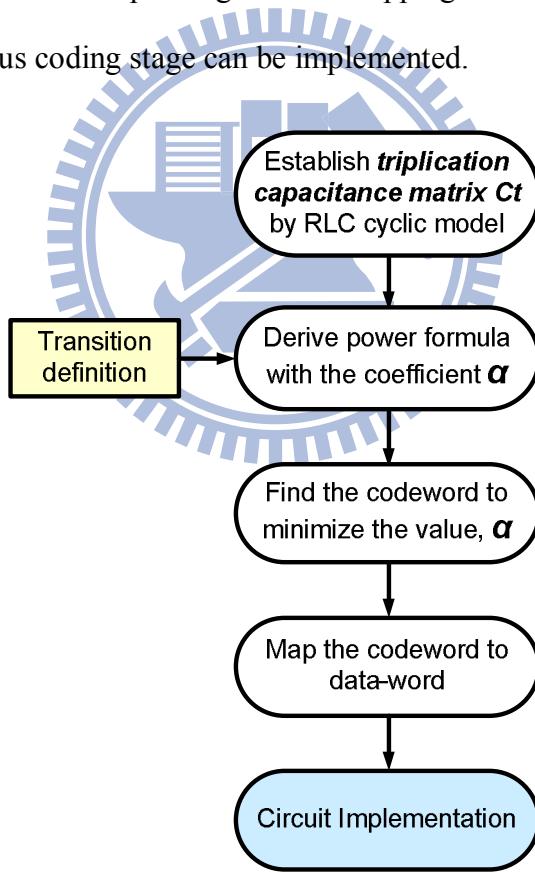


Fig. 3.7 The design flow of the green bus coding stage.

According to the design flow of the green bus coding stage, the modified switching activity,  $\alpha$ , should be minimized. Therefore, to convert the 4-bit dataword into a 5-bit codeword, a 32x32 transition state table is established by calculating  $\alpha$ . Thus, 16

transition patterns are selected with minimal values of  $\alpha$  as the codeword to eliminate crosstalk. The green bus coding chooses a 4:5 code to minimize  $\alpha$  depending on the energy saving bound and the latency of codec. In a data bus, the bit-width of a data is usually a multiple of 4. Therefore, the energy-saving bound of 4:5 to 4:8 codes are between 40% to 55% from the energy-saving bound analysis of [3.28]. However, the latency of the codec will increase significantly as the size of a codeword increases.

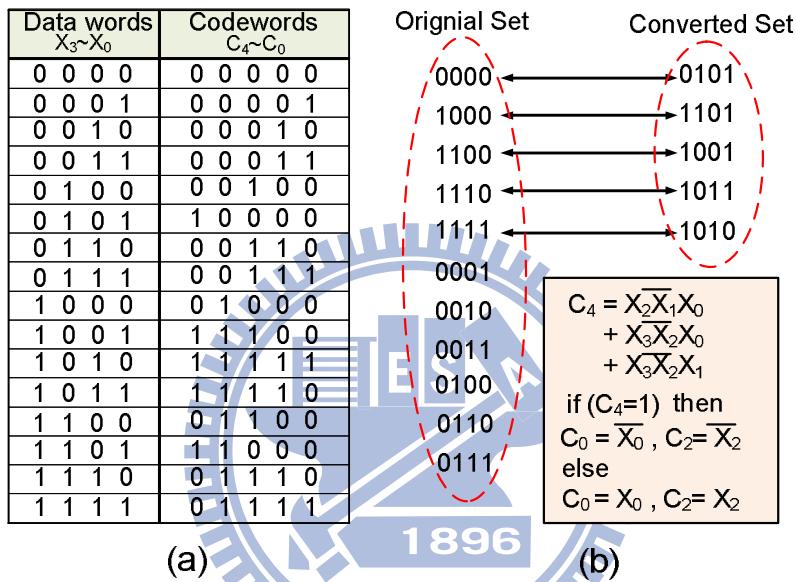


Fig. 3.8 (a) The mapping table between 4-bit dataword and 5-bit codeword of the green bus coding stage (b) The two sets and Boolean expression of the green bus coding stage.

Fig. 3.8(a) shows the relationships between the 4-bit dataword and 5-bit codeword. According to the relationships, the data-word can be grouped into two sets, the original set and converted set as shown in Fig. 3.8(b). When transmitted data are in the converted set, the green bus coding stage converts the data into the original set via one-on-one mapping. Meanwhile, the converted bit,  $c4$ , will be asserted, and  $c0$  and  $c2$  will be inverted and mapped to the original set. Notably,  $X1$  and  $X2$  will always not be modified.

Fig. 3.9 shows the circuit implementation of green bus coding, including the *encoder* and *decoder*. The circuitry of green bus coding is more simple and effective

than other approaches using the joint triplication bus model. An extra shielding line to reduce the coupling effect is not needed between two adjacent 5-bit codewords because the boundary data of the 5-bit codeword are set to roughly 0. Table 3.1 shows the comparisons between green bus coding and increasing wire spacing when  $\lambda = 8$ . Although increasing wire spacing can achieve more energy reduction than green bus coding, it has great amount of area overhead. Additionally, the energy-delay product (EDP) of green bus coding is smaller than that of double wire spacing.

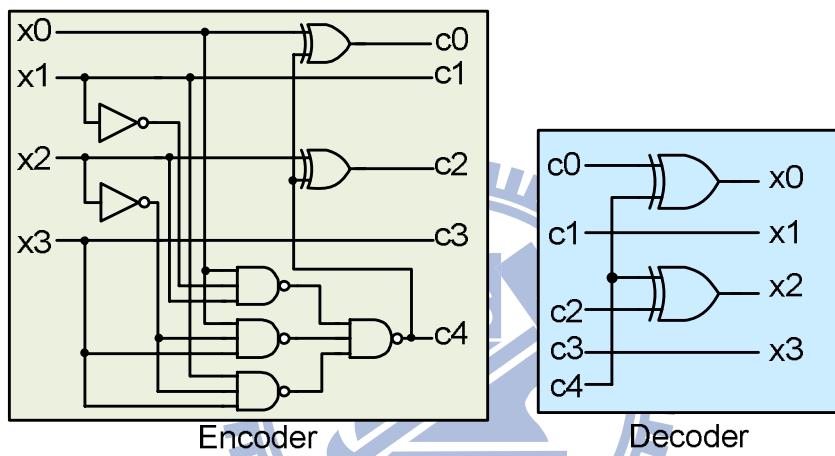


Fig. 3.9 The encoder and decoder of green bus coding stage.

Table 3.1 Comparisons between green bus coding and increasing wire spacing.

(4-bit, $\lambda=8$ )	Area overhead	Energy Reduction	Delay Reduction	EDP Reduction
Green Bus Coding	19%	19.7%	49.5%	59.3%
Double Spacing	23%	23.2%	30.1%	46.4%
Quadruple Spacing	129%	37.3%	39.2%	61.9%

The proposed green bus coding stage has the following properties:

- (1) The encoded bit always equals the data bit at certain bit positions, where  $y_1 = x_1$  and  $y_3 = x_3$ .
- (2) By focusing on the joint bus and error correction coding scheme, the SCG coding scheme can avoid FOC and FPC and reduce FTC to further reduce power consumption.

- (3) Adding extra shielding lines to reduce the coupling effect between two adjacent codeword with increasing coding bits is unnecessary.
- (4) According to the delay model and energy model given by [3.28], the energy dissipation and critical delay are reduced from  $(1+1.5\lambda)CV^2$  to  $(1.18+1.17\lambda)CV^2$  and  $(1+4\lambda)\tau_0$  to  $(1+2\lambda)\tau_0$  via the green bus coding, respectively.  $\tau_0$  is defined as the delay of a crosstalk-free wire.

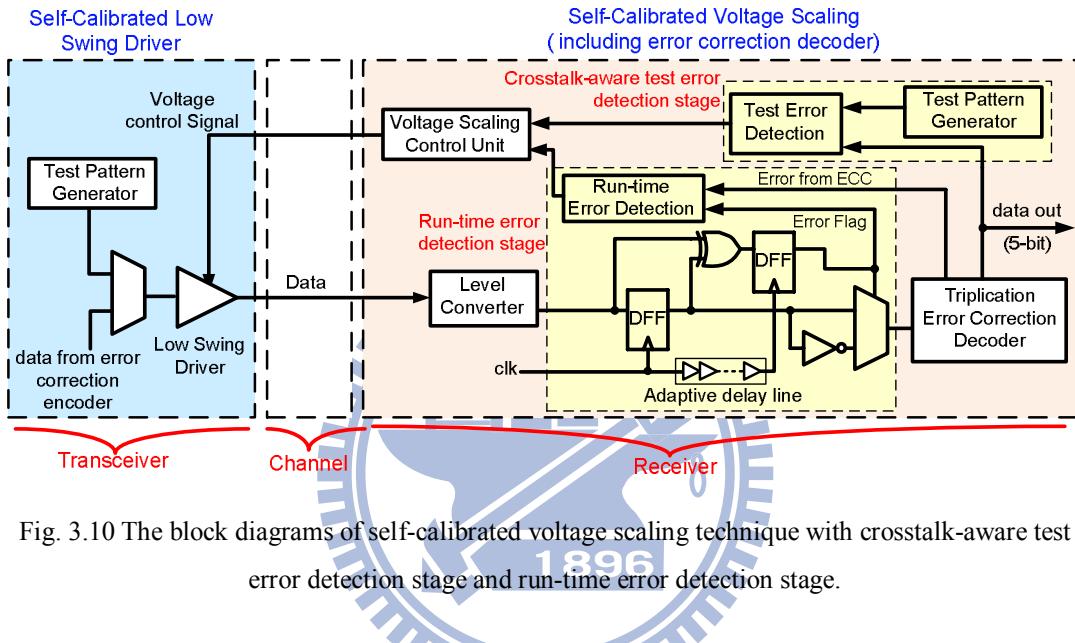


Fig. 3.10 The block diagrams of self-calibrated voltage scaling technique with crosstalk-aware test error detection stage and run-time error detection stage.

### 3.4 Self-Calibrated Voltage Scaling Technique

The proposed self-calibrated voltage scaling technique is applied to reduce the operating voltage of channels for energy reduction and ensure the reliability based on the SCG coding scheme. The self-calibrated voltage scaling technique will identify the optimal operating voltage to trade off between energy consumption and reliability for the self-calibrated circuitry. Fig. 3.10 presents the block diagrams of the self-calibrated voltage scaling technique. This technique is constructed by comprising low swing drivers, level converters, voltage scaling control unit, crosstalk-aware test error detection stage and run-time error detection stage. Depending on the detections about the two error detection stages, the voltage control unit adjusts voltage swing

levels of the link wires. The crosstalk-aware test error detection stage detects errors by maximal aggressor fault (MAF) test patterns in the test mode. The run-time error detection stage detects errors using the double sampling data checking technique and the adaptive delay line. Moreover, the self-calibrated voltage scaling technique is tolerant of timing variations by the adaptive timing borrowing technique. In response to detected errors, the self-calibrated voltage scaling technique can reduce voltage swing for energy reduction and guarantee the reliability is still in the confidence interval simultaneously.

Based on the SCG coding scheme, the triplication error correction coding stage can correct errors for link wires. The SCG coding scheme allows for reductions in signal voltage swing and, at the same time, achieves the same word error rate of un-coded link wires. When the bit error rate is in the range from  $10^{-20}$  to  $10^{-10}$ , a 0.7V signal swing for link wires can maintain the same reliability with the un-coded code at 1.0V as shown in Fig. 3.4. Therefore, a low swing driver and level converter are implemented with three voltage levels as shown in Fig. 11, which are high voltage ( $HV = V_{dd}$ ), middle voltage ( $MV = V_{dd} - V_t$ ) and low voltage ( $LV = V_{dd} - 2V_t$ ). The PMOS diodes are utilized to produce low swing voltages as shown in Fig. 3.11(a) by low- $V_t$  PMOS. In UMC 65nm CMOS technology, the threshold voltage of normal- $V_t$  and low- $V_t$  PMOS are 0.25v and 0.15v, respectively. Therefore, the voltage level will be two levels by normal- $V_t$  device. In order to realize the lowest voltage, 0.7v, low- $V_t$  PMOS and three voltage levels are selected. Three control signals, S0–S2, determine the voltage swing of link wires, and Fig. 3.11(a) shows the relationships between control signals and voltages. Based on the different voltages, the low swing driver and level converter can be implemented as shown in Fig. 3.11(b) and Fig. 3.11(c), respectively. Therefore, the timing overhead of switching voltage can be in one cycle.

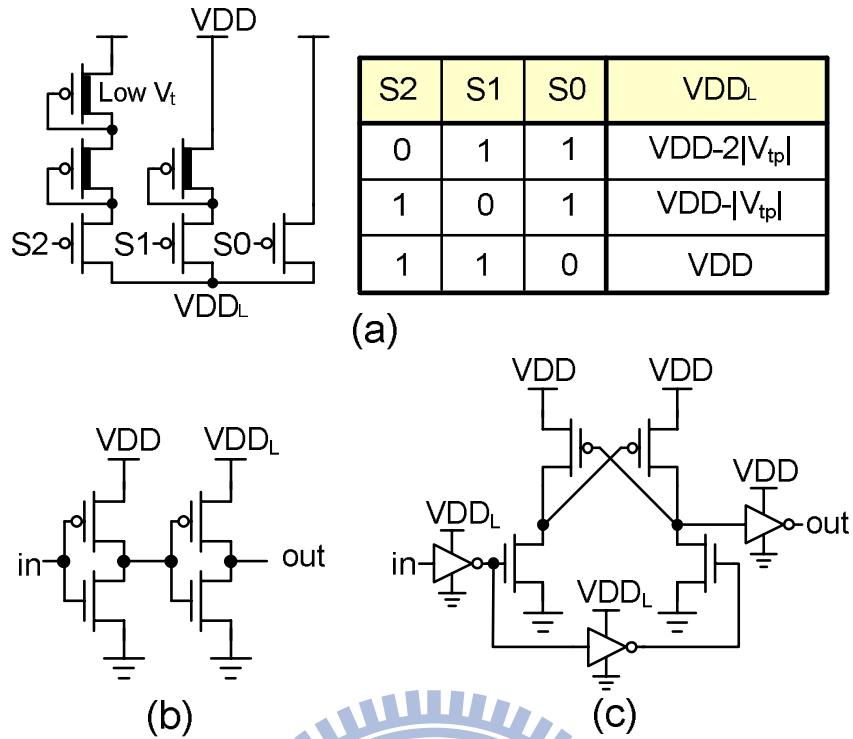


Fig. 3.11 (a) Low swing voltages (b) Low swing driver (c) Level converter.

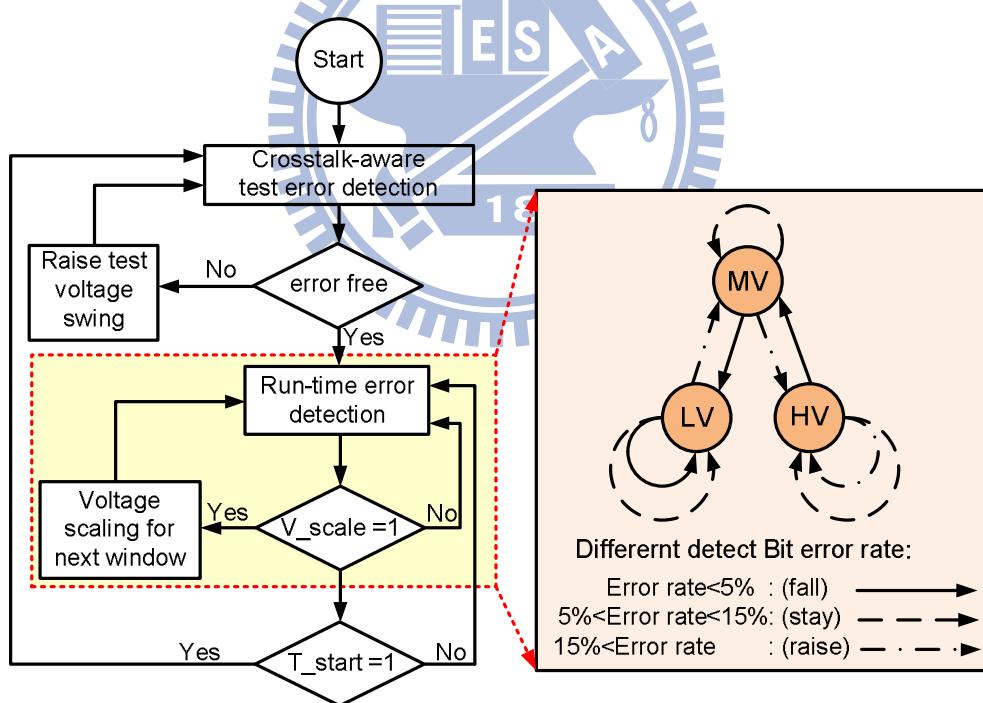


Fig. 3.12 The control policy of self-calibrated voltage scaling technique.

Fig. 3.12 shows the control policy and voltage state diagram of the self-calibrated voltage scaling technique. Therefore, the crosstalk-aware test error detection stage is triggered by  $T_{start}$ , and crosstalk-aware test vectors are generated. Test results are

compared by the test error detector. Initially, the crosstalk-aware test vectors are transmitted at the lowest voltage level of 0.7V. In terms of error correction coding, the error should be zero by the test error detector. If the error detector detects errors, the test vectors will be transferred again with a relatively higher voltage (0.85V or 1V). The initial voltage swing of link wires is determined until the test result is free of errors. When the test is finished,  $T_{\text{finish}}$  and the run-time error-detection stage will be activated. After the crosstalk-aware test error detection stage, the run-time error detection stage raises  $V_{\text{scale}}$  to trigger a scaling mechanism within every  $N$  clock cycles window. Based on the error rate, the voltage control unit can further increase or decrease the signal voltage swing during run-time. But the voltage in the run-time error detection stage cannot be lower than the voltage level determined by the crosstalk-aware test error detection stage. The error rate is defined as the ratio of total transmission data in one window to error data. If the error rate is less than 5%, signal voltage swing is reduced one level or kept at the lowest safe signal. However, if the bit error rate is larger than 5% but less than 15%, the signal voltage swing level is the same as that for the previous window. If the error rate is larger than 15%, signal voltage swing is increased one level or kept at the highest signal swing level. The range of bit error rate detection depends on properties of SCG coding scheme. If uncoded input data are random, the probability of the forbidden pattern condition (two adjacent lines switch in opposite directions, e.g.  $\uparrow\downarrow$  or  $\downarrow\uparrow$ ) of the coding scheme is roughly 15%. Additionally, the 5-bit voltage scaling control unit can determine 5% and 15 % error rate by an 8-bit adder in 256 cycles.

### 3.4.1. Crosstalk-Aware Test Error Detection Stage

The crosstalk-aware test error detection stage is composed of a test pattern generator (TPG), a test error detector (TED), and a control unit that generates the

control voltages for the low swing driver. The crosstalk-aware test error detection stage is triggered by  $T_{start}$ , and then generates crosstalk-aware test vectors. Conventional test pattern generators, such as the Linear Feedback Shift Register (LFSR) [3.29], [3.30], generate pseudo-random pattern sequences. By changing the feedback polynomial of the LFSR, the LFSR generates different subsets of the maximum-length LFSR (maximum  $2^n - 1$  patterns when the LFSR tests  $n$ -bits data with primitive polynomials). However, test patterns generated by the LFSR based TPG are complicated and require a long test time to achieve high error coverage. Hence, a better self-test methodology is needed to achieve low hardware overhead, fast test time, and high error coverage.

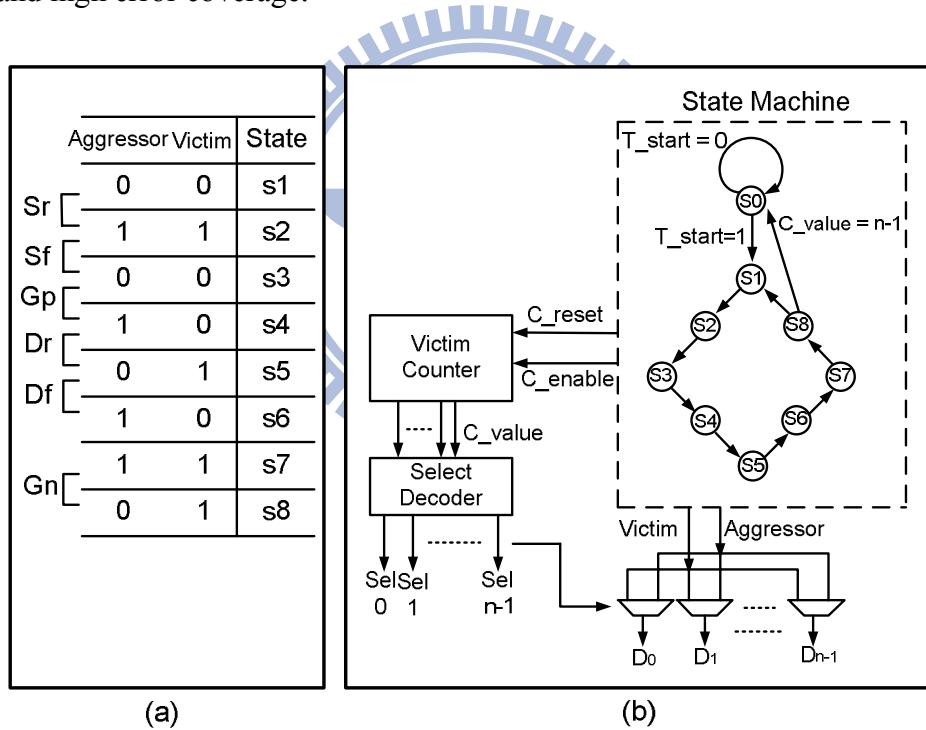


Fig. 3.13 MAF based test pattern generator (a) 8 states complete 6 faults test of MAF model (b) Hardware implementation.

Depending on test vectors, therefore, the test error detector can detect error data following error correction coding. The crosstalk-aware test vectors are generated by a test pattern generator with the maximal aggressor fault (MAF) model as shown in Fig. 3.13 [3.31]. The MAF-based test patterns are a simple pattern stream that represents

six different crosstalk effects: rising speed-up ( $S_r$ ), falling speed-up ( $S_f$ ), rising delay ( $D_r$ ), falling delay ( $D_f$ ), positive glitch ( $G_p$ ), and negative glitch ( $G_n$ ). For test wires with  $n$ -bits, one victim line and  $n-1$  aggressor lines exist. All aggressor lines switch simultaneously to generate speed-up, delay, or glitch error on the victim line. The MAF test vectors can achieve high error coverage. Additionally, the MAF-based test can be considered as an aggressive test that covers other pattern transition cases. To test  $n$ -bit on-chip interconnects, six fault models must be tested on each line. Therefore, testing  $n$ -bit needs  $6n$  test pattern transitions to complete an MAF-based test.

The test pattern generator of the MAF-based self-test methodology is implemented by the finite state machine (FSM). The FSM needs at minimum 8 cycles to complete six faults tests on one victim line, indicating that the test pattern generator requires  $8n$  cycles to complete an  $n$ -bit MAF test. Test time is much shorter than that of the Linear Feedback Shift Register. The FSM, which is triggered by  $T_{start}$  signal, generates the values of the victim line and the aggressor line, counter reset ( $C_{reset}$ ) and counter enable ( $C_{enable}$ ). After each circle (states S1–S8) of the FSM,  $C_{enable}$  triggers the victim counter. The decoder and output 2-to-1 mux are selected to ensure that the data bit ( $D_i$ ) selects the correct value (victim or aggressor value) during the test. When the value of the victim counter ( $C_{value}$ ) is equal to  $n-1$  in the S8 state, the test is finished and returns to the S0 state.

### 3.4.2. Run-Time Error Detection Stage

The run-time error detection stage detects timing variations of link wires. Timing delay variations of on-chip interconnections are due crosstalk noise, process variations, temperature variations and other noises. To overcome timing error, the

master-slave flip-flop (MSFF) [3.32] and double sampling data checking technique [3.33] have been proposed to detect timing errors. The MSFF contains a master flip-flop and a slave flip-flop, both of which operate at the same frequency. However, the slave flip-flop is positively triggered by a delay clock ( $\Delta t$ ) which is proportion to master flip-flop. We assume the data captured by the slave flip-flop is correct. The data captured by the master flip-flop and the slave flip-flop are compared using an XOR gate; an error-flag is generated when the two data are not identical. When an error occurs, the control circuit stalls pipeline data flow for 1 clock and the slave flip-flop resends correct data to the master flip-flop. The principle of the double sampling data checking technique is similar to that of the MSFF.

The timing delay variation of on-chip interconnects affects the design on  $\Delta t$ . The different propagation delay on the on-chip interconnection caused by crosstalk is due to different pattern transients. For the increasing timing variation of on-chip interconnections, detecting timing error is difficult for various voltage levels. However, the MSFF and double sampling data checking technique are limited by the clock period and fixed delay line, respectively. Therefore, the run-time error detection stage is constructed using the adaptive timing borrowing technique as shown in Fig. 3.10. The adaptive timing borrowing technique modifies the double sampling data checking technique with the adaptive delay line. In addition, the adaptive timing borrowing technique also has correction ability via a multiplexer. The modified double sampling data checking technique with the adaptive delay line has the adaptive timing borrowing ability to borrow timing from the next clock period.

Fig.3.14 presents analytical results for timing constraints. To ensure that functionality of the modified double sampling data checking technique is correct, time interval  $\Delta t$  must be set appropriately, and each pipeline stages must be considered. If

the delay between DFF1 and DFF2 exceeds 1 clock cycle, error sampling data of DFF1 are induced. The maximum data path delay can be extended to 1 clock cycle plus time interval  $\Delta t$ , as in Eq. (3.14), where  $t_{DFF}$  is the Clock to Q delay of the D Flip-Flop, and  $t_d$  is the data path delay (from the input of the low swing driver to the output of the level converter),  $t_{XOR}$  is the XOR propagation delay, and  $t_{setup}$  is the setup time of the D Flip-Flop.

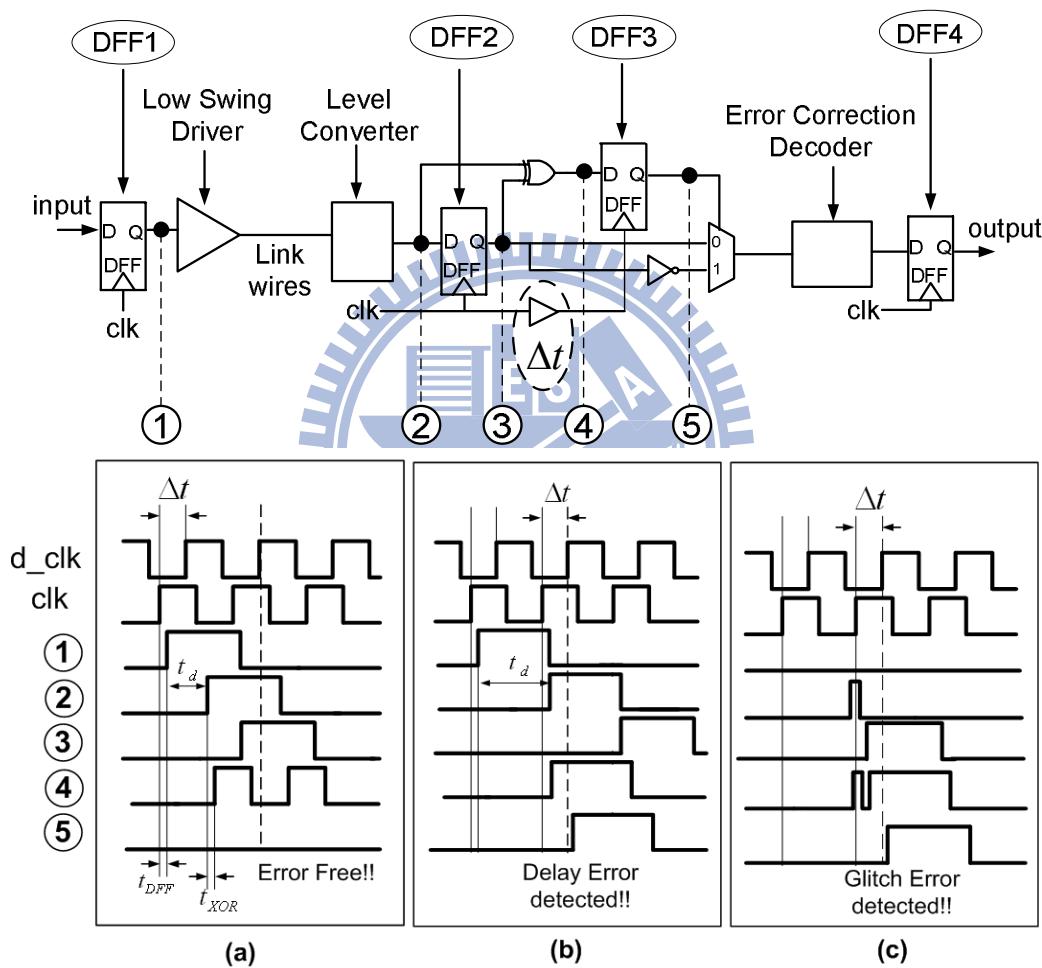


Fig.3.14 Modified double sampling data checking circuit and Wwaveforms (a) error-free (b) delay error (c) glitch error.

$$t_{DFF1} + t_d + t_{XOR} + t_{setup3} < \tau_{clk} + \Delta t \quad (3.14)$$

DFF3 samples the comparison signal, which compares sampling data before DFF2 and after DFF2. In addition, DFF3 must sample the comparison signal before next

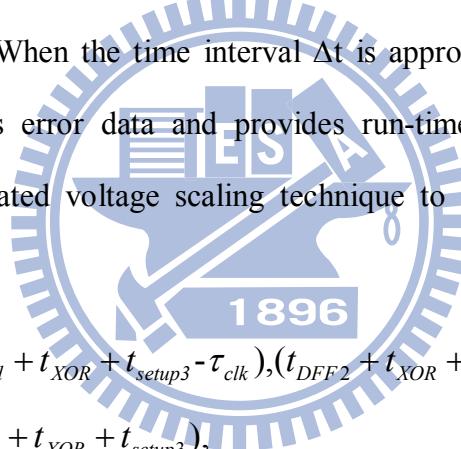
datum arrives. Therefore,  $\Delta t$  should be satisfied as in Eq. (3.15).

$$t_{DFF2} + t_{XOR} + t_{setup3} < \Delta t < t_{DFF1} + t_d + t_{XOR} + t_{setup3} \quad (3.15)$$

Additionally, the pipeline stages after the double sampling data checking stage must satisfy basic constraints, as in Eq. (3.16), to avoid the excessive timing borrowing.

$$\Delta t + t_{DFF3} + t_{MUX} + t_{Decoder} + t_{setup4} < \tau_{clk} \quad (3.16)$$

Eq. (3.14) and (3.15) are the timing conditions that avoid error detections. Eq. (3.16) is the timing condition that prevents setup timing violation of the sequential circuitry. According to Eqs. (3.14)–(3.16), the upper and lower bounds of time interval  $\Delta t$  is derived by Eq. (3.17). When the time interval  $\Delta t$  is appropriate, the run-time error detection stage corrects error data and provides run-time error rate information, allowing the self-calibrated voltage scaling technique to adjust the voltage swing levels of link wires.



$$\begin{aligned} & \text{Max}\{(t_{DFF1} + t_d + t_{XOR} + t_{setup3} - \tau_{clk}), (t_{DFF2} + t_{XOR} + t_{setup3})\} \\ & < \Delta t < \\ & \text{Min}\{(t_{DFF1} + t_d + t_{XOR} + t_{setup3}), \\ & (\tau_{clk} - t_{DFF3} + t_{MUX} + t_{Decoder} + t_{setup4})\} \end{aligned} \quad (3.17)$$

If Eq. (3.14) is not be satisfied, a type I statistical error occurs. The double sampling data checking technique cannot detect true errors, and suppose that the sampling data would be correct. On the other hand, if Eq. (3.15) is not be satisfied, the type II statistical error occurs. The double sampling data checking technique then misjudges and asserts an error flag when the transferred data is correct.

Timing delay variation is caused by the crosstalk effect, process variation, width variation, voltage variation. In view of increasing timing variation, the adaptive delay line is an effective solution that satisfies these conditions. Furthermore, data path

delay  $t_d$  is affected significantly by operating voltages and input vectors. Therefore, the adaptive delay line can generate three time intervals  $\Delta t$  for different signal voltage levels to satisfy the timing condition in Eq. (3.17); thus, the adaptive delay line can be implemented by a digital control delay line with MUXs. Adjust the time interval  $\Delta t$  guarantees the functionality of double sampling data checking technique with different voltage swing levels and process variations.

### 3.5 Simulation Results

This section presents simulation results demonstrating the improvement in energy and reliability via the SCG coding scheme and the self-calibrated voltage scaling technique. All simulation results are based on UMC 65nm 1P9M CMOS technology. For OCINs, the metal layers can be categorized into upper-level, middle-level and lower-level, respectively. In most cases [3.34]-[3.36], the upper-level metal layers are routed for power grids and global clock distribution via low resistance metals. Additionally, the lower-level metal layers are routed for local resources. Therefore, the characteristics of link wires between inter-processor elements are set as metal-6 with a minimum width and spacing of  $0.10\mu\text{m}$  in UMC 65nm 1P9M CMOS technology. Simulation results include analysis of different error-correction coding schemes, energy-delay product (EDP) of different joint coding schemes, energy saving of SCG coding in an 8x8 mesh network, process-variation timing analysis, and analysis of the self-calibrated voltage scaling technique.

Table 3.2 lists different combinations of joint coding schemes, such as the Hamming Code (HC), FTC+HC, FOC+HC and Boundary Shift Code (BSC) in [3.18], One Lumbda Code (OLC)+HC and DAP+shielding (DSAP) in [3.20], JTEC in [3.19], the proposed SCG coding scheme. Additionally, Table 3.2 summarizes different joint

coding schemes for 8-bit link wires, which consist of the physical transfer unit size in channels and routers.

Table 3.2 Different combinations of joint coding schemes

Category	Coding Scheme	Crosstalk Avoidance Coding	Error Correction Coding	Linear Crosstalk Coding	Phit Size (wire)	Phit Size (Router)
ECC	Hamming	-	Hamming	-	12	12
ECCx2	JTEC	Duplication	Hamming +Parity	-	25	25
CAC + ECC	FTC-HC	FTC	Hamming	Shielding	21	21
	FOC-HC	FOC	Hamming	-	16	16
	OLC-HC	OLC	Hamming	Shielding	34	34
	BSC	Duplication	Parity	-	17	17
	DAP	Duplication	Parity	-	17	8
	DSAP	Duplication	Parity	Shielding	25	8
	SCG Green/Triplication	Green	Triplication	-	30	10

Table 3.3 Summaries of different joint coding schemes for 8-bit link wires.

Coding Scheme	Link Wires (8-bit)			Codec		
	Delay ( $\tau_0$ )	Avg. Energy ( $CV_{dd}^2$ )	Lowest $V_{dd}$ (V)	Area ( $\mu m^2$ )	Delay (ns)	Power ( $\mu W$ )
Hamming	$1+4\lambda$	$3.00+5.50\lambda$	0.705	253.3	0.73	190.9
JTEC	$1+2\lambda$	$6.25+4.00\lambda$	0.579	512.2	0.93	311.1
FTC-HC	$1+2\lambda$	$3.38+4.77\lambda$	0.705	465.5	0.83	253.2
FOC-HC	$1+3\lambda$	$3.19+5.14\lambda$	0.705	421.3	0.59	250.1
OLC-HC	$1+\lambda$	$6.76+4.91\lambda$	0.710	961.6	0.62	321.3
BSC	$1+2\lambda$	$4.13+3.81\lambda$	0.710	488.4	0.73	207.6
DAP	$1+2\lambda$	$4.25+4.00\lambda$	0.710	146.3	0.35	68.8
DSAP	$1+\lambda$	$4.25+4.00\lambda$	0.710	149.2	0.35	68.9
SCG Green/Triplication	$1+2\lambda$	$7.05+2.77\lambda$	0.696	266.3	0.28/0.09	103.0/41.5

The maximum delay and average energy of link wires, the corresponding lowest supply voltage of different coding scheme are list in Table 3.3. Table 3.3 also summarizes the codec of different approaches, including the corresponding codec area, power and latency. The lowest supply voltages are theoretical values from Fig. 3.5 when  $\epsilon = 10^{-20}$ . The JTEC uses double error correction coding to enhance error correction. However, codec overhead and energy dissipation are much worse than

those of other approaches. Although the JTEC can reduce the supply voltage to the lowest point at the same uncoded word-error-rate, the latency is larger than others due to long chains of XOR gates. Furthermore, the lowest voltage of JTEC increases rapidly as bit error rate increases.

Except for the SCG coding, DAP and DSAP, the critical delays of other codec are larger than 0.5ns. Consequently, these codec are not appropriate for integration into high speed routers. Therefore, the physical transfer unit sizes in routers of these codec are bigger than that of proposed coding scheme; thus network area and energy consumption increase. The delay of green coding stage and triplication error correction stage are 0.28ns and 0.09ns, respectively. And the power consumption of triplication error correction stage is only  $41.5\mu\text{W}$ . Hence, the proposed SCG coding scheme has the smallest codec overhead. Additionally, the green bus coding stage is only integrated in the sender node and receiver node.

The delay and energy of link wires are calculated via the delay model and energy model given by [3.28], where  $\tau_0$  is defined as the delay of a crosstalk-free wire. The proposed SCG coding scheme achieves the most energy reduction by reducing coupling effects on link wire, and avoids the FOC and FPC by the triplication error correction coding stage. Additionally, the SCG coding scheme can reduce the FTC and self-switching activities using the green bus coding stage depending on the joint triplication power model. Although the triplication error correction stage triplicates transferred data and increases the physical transfer unit size on link wires, it also enhances data reliability and avoids the worst crosstalk patterns. Moreover, the delay can be reduced from  $(1+4\lambda)\tau_0$  to  $(1+2\lambda)\tau_0$ .

Fig. 3.15(a) shows the energy-delay product (EDP) reduction compared to un-coded code under different  $\lambda$  values. Coefficient  $\lambda$  is defined as the ratio between

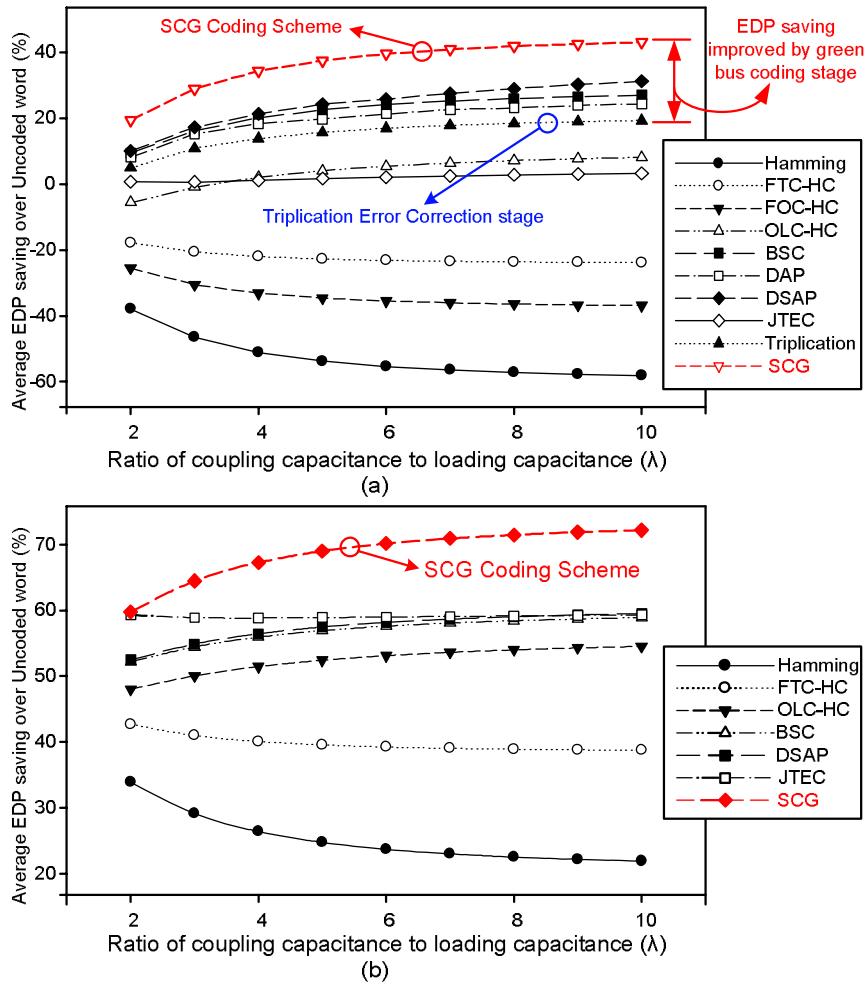


Fig. 3.15 The energy-delay product (EDP) reduction to un-coded code under different values of  $\lambda$  with  
 (a) full swing signal (b) the lowest swing signal.

coupling capacitance of two adjacent lines and loading capacitance. The proposed SCG coding achieves the highest EDP reduction regardless of the value of  $\lambda$ . Through the tradeoff between reliability and power consumption, the signal swing levels of specific codes can be reduced further to the lowest values based on the error correction abilities. The lowest signal swing guarantees the same level of word error rate as that of the un-coded code. Fig. 3.15(b) shows the energy reduction compared to un-coded code under different  $\lambda$  values and the lowest signal swing level. Simulation results indicate that the proposed SCG coding realizes more EDP saving than other joint coding schemes. When  $\lambda$  equals 4 with a full swing signal (1.0v), the

SCG coding scheme can achieve a 34.34% EDP reduction compared to un-coded word and a 56.54% EDP reduction relative to that achieved by traditional Hamming codes. The coding schemes can further increase EDP savings at the lowest operating voltages. In Fig. 3.15(b), the proposed SCG coding achieves a 67.29% EDP saving relative to that achieved by the un-code word when  $\lambda$  is 4 and operating voltage is 0.69v.

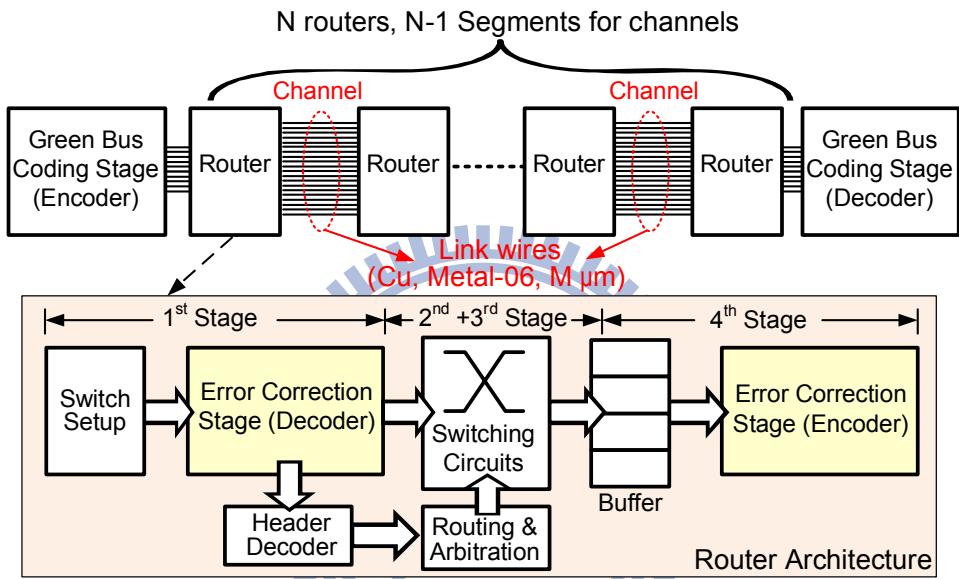


Fig. 3.16 Simulation environment setup with different number of routers (N) and different lengths (M) of link wires.

The proposed SCG coding is also simulated with different lengths of link wires. Fig. 3.16 shows the simulation environment setup with different number of routers (N) and various lengths (M) of link wires. The green bus coding stage is only integrated in the routers of the sender node and receiver node. The architecture of the routers is set as 5input/output ports with 4-stage pipeline for mesh interconnection networks. The first stage includes switch setup, error correction decoder and header decoder. The second stage and third stage are routing traversal and arbitration, respectively. The final stage is error correction encoder and link wires. The length of link wires is set as  $M \mu m$  of metal-6 with a minimum width and spacing of  $0.10 \mu m$ . The clock frequency

is as high as 1GHz. Fig. 3.17 illustrates energy reduction with different number of routers (N), different lengths (M) under the normal voltage (1.0v) and lowest voltage (0.7v). According to some NoC chips [3.34]-[3.36], the length of link wires is set from 200 $\mu$ m to 1800 $\mu$ m. The energy reduction increases while the length of link wires increases. Additionally, both reducing coupling effect and supply voltage can achieve significant energy saving by the SCG coding scheme.

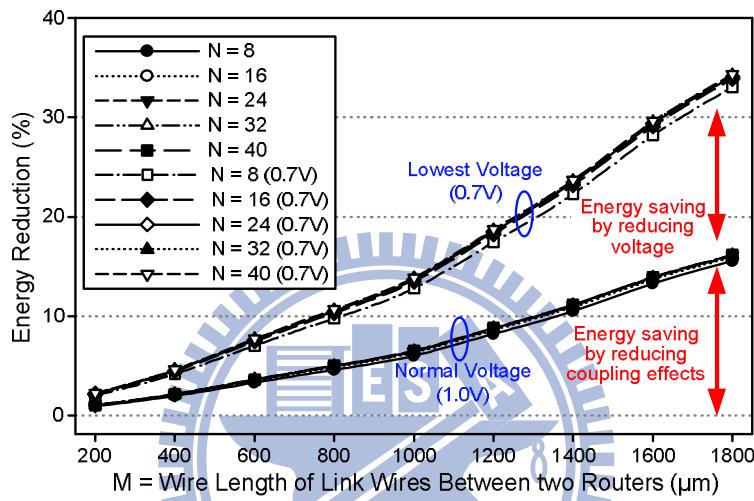


Fig. 3.17 Energy reduction under different lengths of link wires and different number of routers.

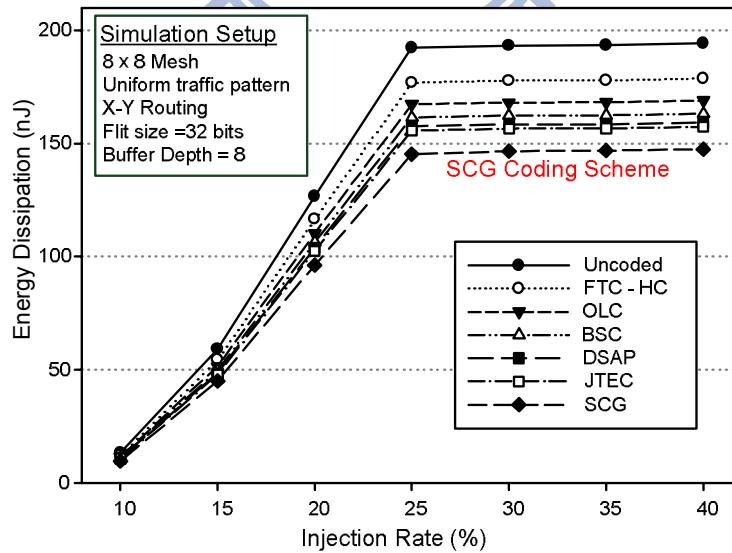


Fig. 3.18 Energy dissipation of an 8x8 mesh-NoC with different joint CAC and ECC coding schemes.

Fig. 3.18 shows the energy dissipation of an 8x8 mesh interconnection network with different joint CAC and ECC coding schemes under their lowest supply voltages.

The simulation environment is set as an  $8 \times 8$  mesh topology with uniform random patterns. The routing and arbitration algorithms are XY routing and round-robin, and The FIFO depth of each output buffer is 8 flits. The size of each flit size is 32 bits. The length of link wires is set as  $800\mu\text{m}$  of metal-6 with a minimum width and spacing of  $0.10\mu\text{m}$ . The clock frequency is as high as 1GHz. In order to reach 1GHz, the 32-bit un-coded data is divided into four 8-bit groups for different joint CAC and ECC coding schemes. The proposed SCG coding scheme can realize the most energy saving compared to other joint CAC and ECC coding schemes.

The self-calibrated voltage scaling technique is designed and simulated with the SCG scheme based on UMC 65nm CMOS technology. The length of link wires is set as  $800\mu\text{m}$  of metal-6 with a minimum width and spacing of  $0.10\mu\text{m}$ . The clock frequency is as high as 1GHz. Therefore, the timing of link wires should be analyzed with different voltage levels and process variations. The different transient patterns must also be considered. This analysis can help designers implement the adaptive delay line and guarantee correct function of the double sampling data check mechanism. The modified double sampling data checking circuit provides error information for the self-calibrated voltage scaling mechanism during run-time. However, the time interval,  $\Delta t$ , must satisfy the constraint discussed in Section V. The data path delay,  $t_d$ , is clearly affected by voltages (swing levels of link wires) and input data vectors. Additionally, PVT (process, voltage and temperature) variation affects both devices and on-chip wires. Therefore, the delays of link wires are analyzed using Monte-Carlo simulations of PVT variation at different voltage levels.

Fig. 3.19(a)–(c) show the data path delay,  $t_d$ , of rising speed-up(Sr) case, falling speed-up (Sf) case, rising delay (Dr) case, falling delay (Df) case, normal rising(Nr) case and normal falling (Nf) case under high voltage (1.0v), medium voltage (0.85v)

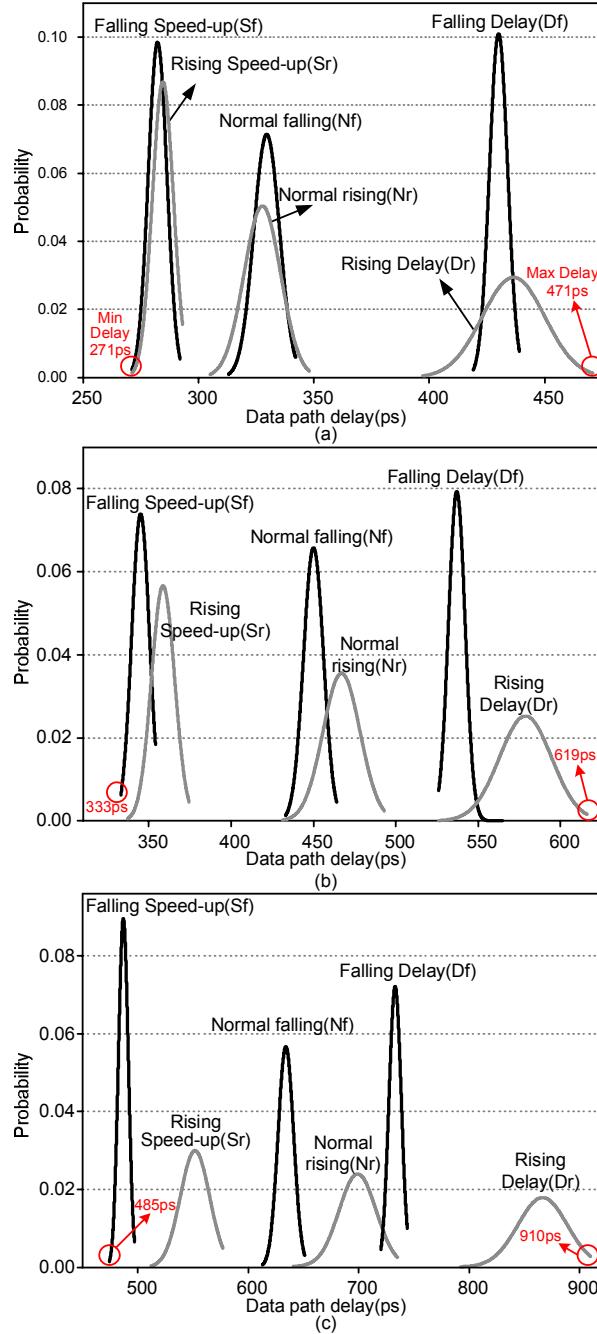


Fig. 3.19 The data path delay( $t_d$ ) distributions of rising speed-up, falling speed-up, rising delay, falling delay, normal rising and normal falling cases under (a) high voltage (1.0v) (b) medium voltage (0.85v) (c) low voltage (0.7v).

and low voltage (0.7v), respectively. The supply voltages have a 15% variation in  $3\sigma$  range and the mean are 1.0V, 0.85v, 0.7v. The maximum value and minimum value of  $t_d$  occur in the Dr case and Sf case. The maximum and minimums value under 0.7V, 0.85V and 1V are 910/485 (ps), 619/333 (ps), and 471/271 (ps), respectively.

According to Eqs. (3.12)–(3.15), the upper bounds of  $\Delta t$  under 0.7V, 0.85V and 1V are about 485ps, 333ps, and 271ps, respectively. Operating voltage obviously influences the timing interval. Therefore, the adaptive delay line can generate three time intervals,  $\Delta t$ , for different signal voltage levels: 450ps, 300 (ps), and 200 (ps), which are 45%, 30% and 20% of a clock period. Therefore, the adaptive delay line can be designed using a digital control delay line. Adjustments to the time interval guarantees functionality of double sampling data checking technique at different voltage swing levels and process variations. Nevertheless, analysis indicates that timing delay variation on link wires is much smaller under high operating voltage. In other words, if the error rate detected by the double sampling data checking technique increases, the control unit will increase the voltage to narrow the timing variation and enhance reliability.

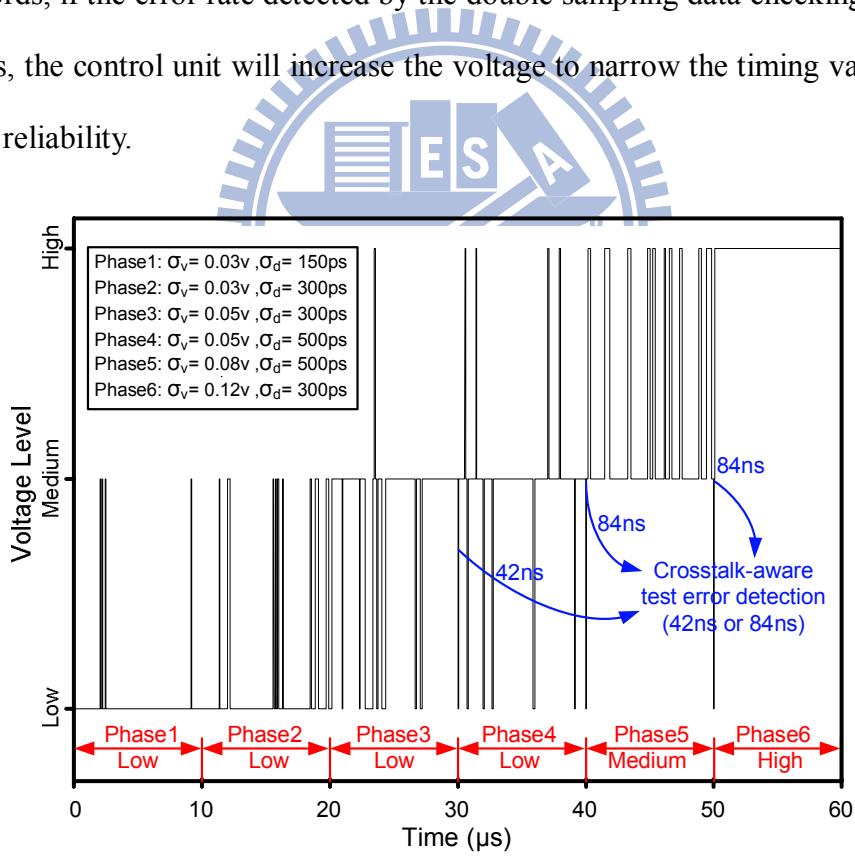


Fig. 3.20 Voltage levels of the self-calibrated voltage scaling technique under six phases with different noise distributions and timing variations.

Fig. 3.20 illustrates the adaptive voltage by the self-calibrated voltage scaling technique under six phases with different noise distributions and timing variations.

The noise distributions ( $\sigma_v$ ) and timing variations( $\sigma_d$ ) are distributed in  $|3\sigma|$  range. The timing variations may caused by process variation, temperature variation, large current density and coupling effect. The control policy of the proposed self-calibrated voltage scaling technique is well-described in Section V. The test time of the crosstalk-aware test error detection stage is 42 cycles (40 cycles for testing, 2 cycles for feedback and adjusting voltage) or 84 cycles. In phase1-4, the initial voltage level is the lowest voltage determined by the test stage. Additionally, the initial voltage levels in phase5 and phase6 are medium and high, respectively. The voltage in the run-time error detection stage cannot be lower than the voltage level determined by the crosstalk-aware test error detection stage. Therefore, in phase 6, the voltage level is always high in the run-time stage. Based on the error rate, the voltage control unit can further increase or decrease the signal voltage swing during run-time. The timing overhead of voltage switching is 1 cycle over (256+2) cycles.

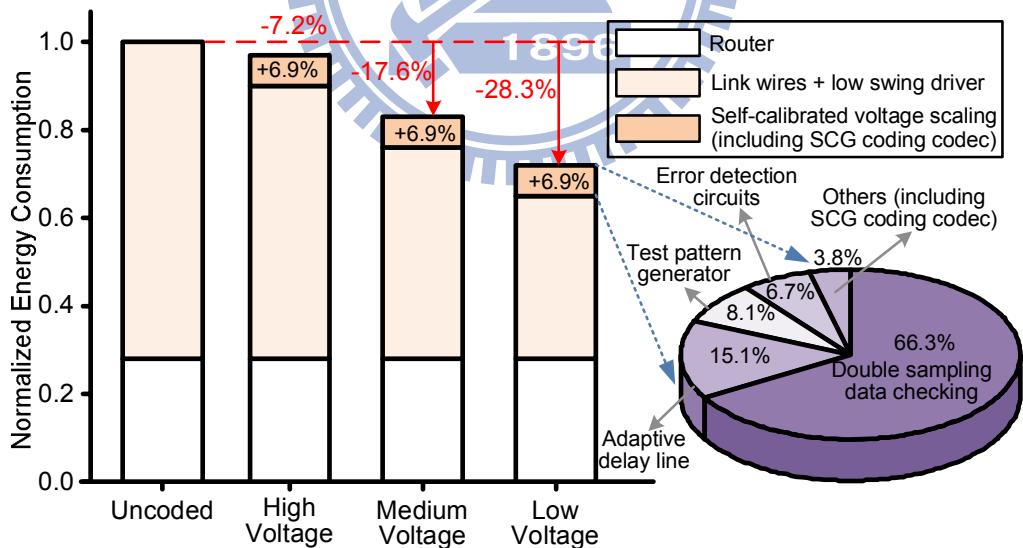


Fig. 3.21 Energy analysis of the self-calibrated energy-efficient and reliable interconnection architecture.

In OCINs, link wires in channels dominate the overall power consumption in advanced technologies. The proposed SCG coding scheme eliminates most crosstalk effects and achieves energy reduction. From Fig. 3.15(b), the EDP reduction of low

swing link wires can reach above 60% compared with that of an un-coded bus when low swing drivers are operating at 0.7V. The proposed self-calibrated voltage scaling technique finds the optimal operating voltage, and the trade-off between energy consumption and reliability is determined by the self-calibrated circuitry. However, the power overhead of the self-calibrated voltage scaling technique reduces the energy efficiency of the channels. Fig. 3.21 shows the energy analysis of the proposed self-calibrated energy-efficient and reliable channels at different voltages. The wire length is set as 1800 $\mu$ m. The SCG coding stage reduces the energy consumption about 14.1% by decreasing the coupling effect and self-switching activities. From Fig. 3.21, the total overhead of the SCG coding scheme and self-calibrated voltage scaling technique is roughly 6.9%. To elucidate the energy overhead, the right side in Fig. 3.21 shows the energy breakdown of the SCG coding and self-calibrated voltage scaling. The double sampling data checking mechanism with the adaptive delay line accounts for almost 80% of energy overhead as a large number of flip-flops is needed. If error correction decoders are moved to before the run-time error detection stage, energy overhead can be reduced by decreasing the number of flip-flops to one-third. However, not only reliability will deteriorate, but the range of adaptive timing borrowing will degrade. Therefore, this is again a trade-off between reliability and energy consumption.

Table 3.4 Summaries of SCG coding and self-calibrated voltage scaling lists the summaries of the SCG coding scheme and self-calibrated voltage scaling technique, including area overhead in a router, energy overhead and energy reduction in channels. The wire length is also set as 1800 $\mu$ m. The energy reduction of the self-calibrated voltage scaling technique is due to the low swing of link wires. The total area overhead is about 14.4% related to a router, which is using X-Y routing and

round-robin arbitration. The router architecture is set as 5input/output ports with 4-stage pipeline. And the FIFO depth of each output buffer is 8 flits. The size of each flit size is 32 bits. The area breakdown of adaptive double sampling data checking, MAF-based test generator and voltage control unit in the self-calibrated voltage scaling are 71%, 8% and 21%, respectively.

Table 3.4 Summaries of SCG coding and self-calibrated voltage scaling

(length=1800μm, low voltage)	Area overhead in a router	Energy Overhead	Energy Reduction (Channels)
SCG Coding	1.21%	0.26%	14.1%
Self-calibrated voltage scaling	13.23%	6.62%	21.1%

### 3.6 Summary

The physical effects of crosstalk and PVT variations in nano-scale technologies degrade the performance of on-chip interconnection networks (OCINs). This work uses a combination of a *self-calibrated voltage scaling technique* and a *self-corrected green (SCG) coding scheme* to overcome increasing variations and achieve energy-efficient on-chip data communication. The SCG coding scheme is used to construct reliable and energy-efficient channels. The SCG coding scheme has two stages, the triplication error correction coding stage, and the green bus coding stage. Triplication error correction coding is a reliable mechanism that achieves rapid correction ability to reduce the physical transfer unit (phit) size in routers via self-correction at the bit level. Green bus coding reduces energy reduction significantly via a joint triplication bus power model that eliminates crosstalk effects. The self-calibrated voltage scaling technique is designed with the SCG coding scheme. The self-calibrated voltage scaling technique adjusts the voltage swing of link wires via two error detection stages, the crosstalk-aware test error detection stage and

run-time error detection stage. Furthermore, the self-calibrated voltage scaling technique is tolerant to timing variations of channels. Based on UMC 65nm CMOS technology, the proposed self-calibrated energy-efficient and reliable channels reduce energy consumption by nearly 28.3% compared with that of un-coded channels at the lowest voltage.



# ***Chapter 4:***

## ***Two-Level FIFO Buffer Design for Routers***

The on-chip interconnection network (OCIN) is an integrated solution for system-on-chip (SoC) designs. The buffer architecture and size, however, dominate the performance of OCINs and affect the design of routers. This work analyzes different buffer architectures and uses a data-link two-level FIFO (first-in first-out) buffer architecture to implement high-performance routers. The concepts of shared buffers and multiple accesses for buffers are developed using the two-level FIFO buffer architecture. The proposed two-level FIFO buffer architecture increases the utilities of the storage elements via the centralized buffer organization and reduces the area and power consumption of routers to achieve the same performance achieved by other buffer architectures. Depending on a cycle-accurate simulator, the proposed data-link two-level FIFO buffer can realize performance similar to that of the conventional virtual channels, while using 25% buffers. Consequently, the two-level FIFO buffer can achieve about 22% power reduction compared with the similar performance of the conventional virtual channels using UMC 65nm CMOS technology.

### **4.1 Background**

The generic OCIN is based on a scalable network, which considers all requirements associated with on-chip data communication and traffic. OCINs have a few beneficial characteristics, namely, low communication latency, low energy consumption constraints, and design-time specialization. The motivation in establishing OCINs is to achieve performance using a system communication

perspective. OCINs provide the micro-architecture and the building blocks for network-on-chips (NoCs), including network interfaces, routers and link wires [4.1], [4.2].

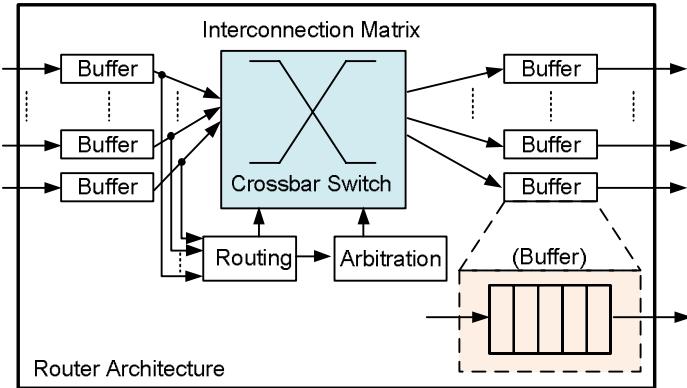


Fig. 4.1 A generic router architecture.

Routers are the essential components of OCINs. Fig. 4.1 shows a generic router architecture, consisting of a set of input buffers, an interconnect matrix, a set of output buffers and control circuitries, including a routing controller, an arbiter and an error detector. The control circuitries serve ancillary tasks and implement some functions of the control flow protocol. Additionally, the interconnection matrix can be implemented using a single crossbar or by cascading various stages. The control circuitry and interconnection matrix are key components of routers [4.3]. Furthermore, the buffers significantly increase the overall performance and decrease the complexity of control policies [4.4], [4.5]. The buffers allow for local storage of data that cannot be routed immediately. Unfortunately, queuing buffers have high costs in terms of area and power consumption; thus, implementations of OCIN design strive against limited buffer size. In the realm of on-chip buffer design, both size and organization are directly related to performance and power consumption of the OCIN [4.6]. Buffer size in particular has been thoroughly investigated in [4.7]-[4.10]. If a design lacks sufficient buffer space, buffers may fill up too fast to decrease the overall performance;

conversely, over-provisioning buffers clearly wastes scarce area resources. Thus, in this work, buffer utilization is optimized via a centralized buffer in a router as shown in Fig. 4.2(a). Compared with a generic router as shown in Fig. 4.2(b), a centralized buffer has a shared buffer mechanism allowing channels to share the centralized buffer with sufficient buffer space.

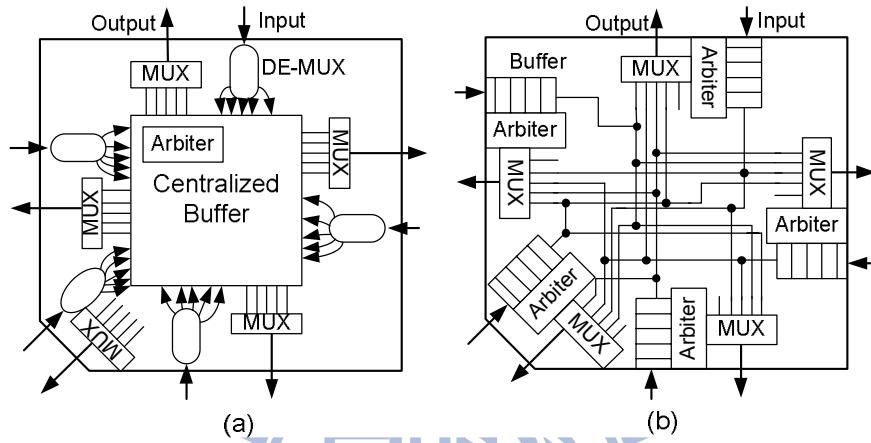


Fig. 4.2 (a) A router with the centralized buffer (b) A generic router.

A data-link two-level FIFO (first-in first-out) buffer architecture with the centralized shared buffer is proposed in this paper. The proposed two-level FIFO buffer architecture has a shared buffer mechanism allowing the output channels to share the centralized FIFO with sufficient buffer space. Additionally, the proposed architecture reduces the area and power consumption to achieve the same performance. The reminder of this chapter is organized as follows. Section 4.2 compares and analyzes different buffer architectures and different circuit implementations. The concept of the proposed two-level FIFO buffer architecture is presented in Section 4.3. Section 4.4 and 4.5 describe the behavior and circuit implementation of two-level FIFO buffer for the synchronous router and asynchronous router, respectively. The associated two-level FIFO buffer architectures are presented in Section 4.5. Section 4.7 shows simulation results. Finally, summaries are given in Section 4.8.

## 4.2 Buffer Implementations and Architectures

The queuing buffer is adopted for routers or network interfaces to store un-routed data. Buffer size and management are directly linked to the flow control policy which affects OCIN performance and resource utilization [4.3]. Buffer architectures can be classified by their location and circuit implementation of buffers. Queuing buffers consume the most area and power among composing blocks in OCINs [4.5], [4.11]. However, insufficient buffer size induces head-of-line blocking problems. Fig. 4.3 shows an example of the head-of-line blocking problem. When head data of a virtual channel cannot be routed and data behind the head data are occupying queuing buffers, network performance is decreased. Nevertheless, head-of-line blocking problems reduce the network performance and increase power consumed during on-chip data communication. Therefore, head-of-line blocking is a key factor when evaluating different buffer architectures.

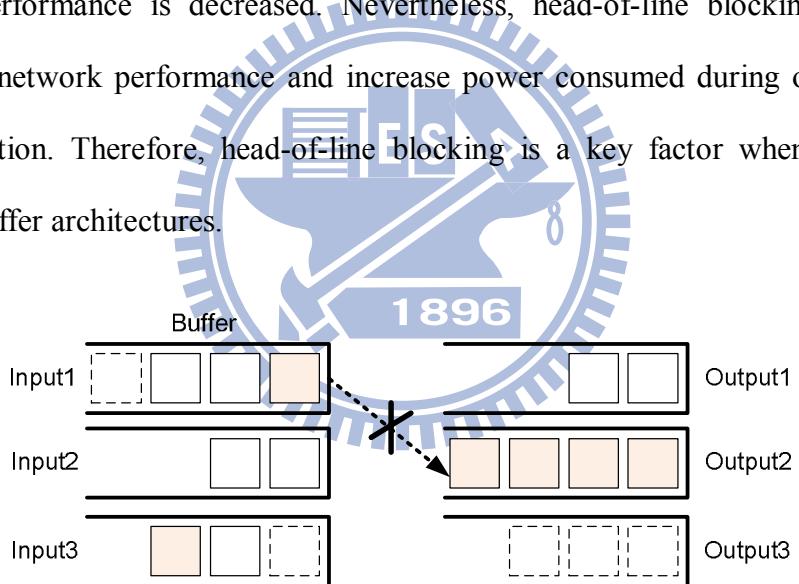


Fig. 4.3 Head-of-line blocking problem induced by insufficient buffer.

The buffer circuits can be implemented using registers (flip-flops) or SRAM according to the buffer sizes. For large capacity queuing, the SRAM-based queuing buffer with separated read/write ports is preferred over a register-based buffer [4.12], [4.13]. However, SRAM incurs large latency overhead [4.5]. For achieving high-performance OCINs, register-based buffers are usually realized in the routers with small buffer sizes. Since register-based implementations have a limited capacity

due to rapid increasing power consumption and circuit area [4.6], [4.10]. In most OCINs, register-based buffers are adopted to provide high bandwidth of on-chip data communication. Consequently, register-based buffers can be classified into four different implementations — (a) Shift Register, (b) Bus-In Shift-Out Register, (c) Bus-In Bus-Out Register, and (d) Bus-In MUX-ut Register [4.1].

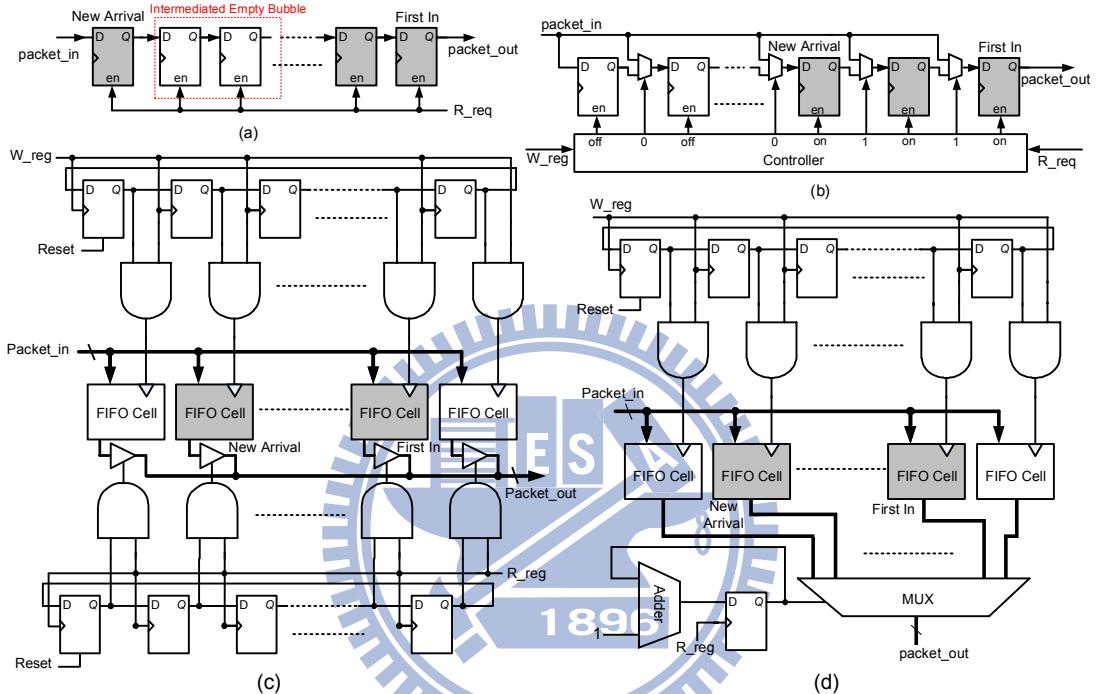


Fig. 4.4 Different buffer implementation (a) Shift Register (b) Bus-In Shift-Out Register (c) Bus-In Bus-Out Register (d) Bus-In MUX-Out Register.

Fig. 4.4(a) shows a conventional shift register. When a consumer sends a request to a buffer, a shift register will enable all registers and shift the data to the output port. Indeed, implementing a shift register is less complicated than implementing others. However, intermediate empty cells induced by different packet in/out rates temporally influence the network performance by adding unnecessary latency. Nevertheless, shifting all registers in a buffer consumes a huge amount of power. Implementing a shift register on a chip is not desirable due to unnecessary latency and massive power consumption. Fig. 4.4(b) shows the Bus-In Shift-Out Register, which only shifts full cells to remove intermediate empty bubbles. An arrival packet can be stored in the

empty cell behind the full cells. Hence, this register can remove unnecessary latency and power consumption caused by empty bubbles. However, as queuing capacity increases, the driving ability of the sender should be increased for large fan-outs. Furthermore, a bus-in shift-out still consumes large amounts power by shifting all occupied cells. To reduce power consumption during shifting operation, Fig. 4.4(c) shows a Bus-In Bus-Out Register, and all register outputs are connected to a shared output bus via tri-state buffers. The writing and reading tokens constructed in rings are the head and tail of full cells, respectively. The tri-state buffers are controlled by the reading token for reading the first-in packet, while the writing token activates the register, which is behind the full cells, to store the input packet. As queuing capacity increases, the capacitance of shared input/output buses also increase, especially the output bus. The parasitic capacitance of tri-state buffers will increase both delay and power consumption. Therefore, the Bus-In MUX-Out Register with output multiplexers can be utilized to eliminate the parasitic capacitance of tri-state buffers.

Fig. 4.4(d) illustrates the Bus-In MUX-Out Register. Additionally, a bus-in MUX-out register needs an extra adder as a pointer and to calculate output packet address.

Depending on the location of queuing buffers, buffers can be placed before or after the interconnection matrix in a router; these buffers are input buffer and output buffer, respectively. To be sure, input buffers and output buffers differ. If a data word is delayed in a router with input buffers, it will stall all data words arriving at the same input. None can be processed until the first data word has been forwarded successfully. With output buffers, this situation differs because switching is performed prior to buffering. If a router cannot send data through one of its outputs, the buffers at that output will fill up. However, congestion on outputs has no immediate influence on inputs; that is, successive data words can still be received. An architectural

disadvantage of output buffering is that in one cycle, data from multiple input ports may be written to the same output port. Nevertheless, a multiple-access buffer can be implemented in parallel at the output to deal with this shortcoming. Both output buffers and input buffers can cause the head-of-line blocking problem and stall input data. Fig. 4.5 shows the input buffers, middle buffers and output buffers in routers. During middle buffering, the buffer placement moves to the middle of switching circuits. Middle buffer architectures have  $O(N^2)$  buffer blocks for an  $N$ -port router, while input and output buffering architecture only have  $O(N)$  buffer blocks. The middle buffer architecture, however, can reduce the effects of head-of-line blocking via multiple virtual channels during switching. This is a trade-off between traffic problems and buffer sizes.

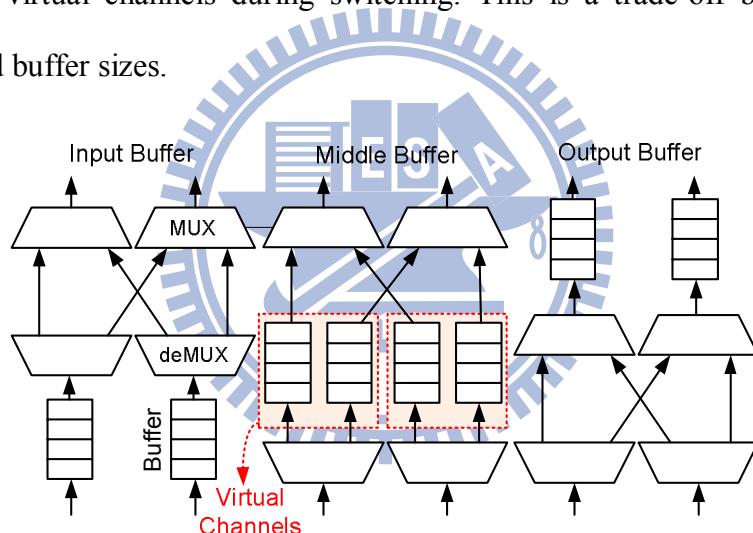


Fig. 4.5 Diagram of input buffer, middle buffer and output buffer.

Since buffer resources are costly in resource-constrained OCIN environments, minimizing buffer size without adversely affecting performance is essential. However, based on observed traffic patterns, buffer size and architecture cannot be changed dynamically during operation. Therefore, some approaches [4.6], [4.7] optimize pre-determined buffer size during the design stage via a detailed analysis of application-specific traffic patterns. Additionally, static virtual channel allocation techniques were proposed to optimize the performance, area and power for target

applications based on the traffic characteristics [4.9], [4.14].

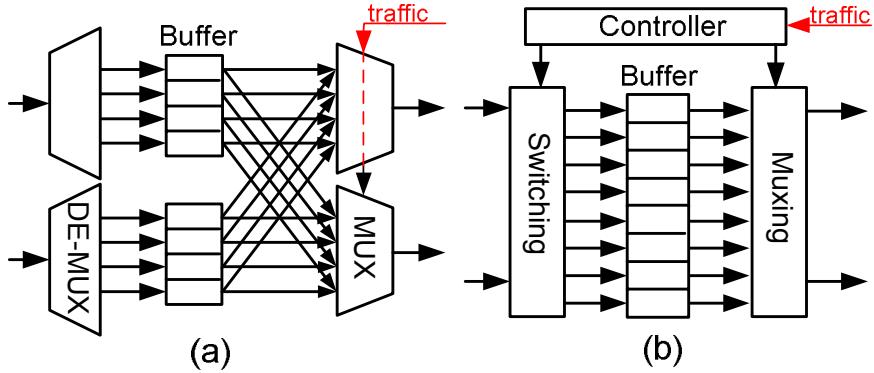


Fig. 4.6 Concepts of (a) dynamic virtual channel allocation (b) centralized shared buffer.

For general-purpose and reconfigurable SoC executing different applications, advanced buffer architectures maximize the utilization of buffers under different traffic patterns in NoC applications. As virtual channels are not equally used in different applications, dynamically allocated multi-queue (DAMQ) buffer schemes were proposed to share a common buffer [4.15]-[4.18]. However, these approaches are not suited to OCIN implementation, which is typically resource-constrained [4.19]. Moreover, NoC applications are intolerant of large latency against the quality of service constraint. Hence, in view of resource and latency overhead, dynamic virtual channel allocation schemes were proposed to maximize throughput for resource-constrained OCIN [4.19]-[4.23]. Fig. 4.6(a) shows the concept of dynamic virtual channel allocation techniques to share the virtual channels and arbitrate output packets based on the traffic conditions. The dynamic virtual channel regulator (ViChaR) proposed in [4.19] introduced a unified buffer structure that dynamically allocated virtual channels and buffer resources based on network traffic patterns. The ViChaR has the unified buffer structure and unified control logic. The unified buffer structure shares buffers in virtual channels for each input port. Additionally, the unified control logic controls the arriving/departing pointers and virtual channel

allocation of each virtual channel via virtual channel control tables and dispensers. However, the hardware overhead would increase non-linearly. In view of this, other dynamically-allocated virtual channel architectures were proposed by inspecting the physical link state and speculating the packet transferring [4.20]-[4.23]. However, when the shared buffers of an input port are full, these approaches do not provide a mechanism for accessing the buffers of other virtual channels at other input ports. Furthermore, the performance of these dynamical virtual channel allocation schemes is also limited since the resource-constraints of the pointers and virtual channel control tables.

Fig. 4.6(b) shows the centralized shared buffer architecture that maximizes buffer utilization [4.24]-[4.26]. Shared buffer architectures are implemented by centralized buffer organizations, which dynamically alter buffer size for different channels. The input packets from different ports can access all buffers without any head-of-line blocking. This architecture enhances OCIN performance regardless of traffic type. Shared buffering, in addition, achieves the best buffer utility with the fewest memory elements. The centralized shared buffer architectures enhance the buffer utilization via allocation tables [4.25], [4.26]. Nevertheless, the control mechanisms of these shared buffer architectures are more complex than those of other buffer architectures and increase the pipeline stages. Hence, the new proposed data-link two-level FIFO buffer architecture is utilized as the shared buffer architecture to simplify the shared buffer architecture and achieve better performance than other buffer architectures while not increasing buffer size.

### 4.3 Concept of Two-Level FIFO Buffer Scheme

The proposed two-level FIFO buffer is constructed by a centralized level-2 FIFO

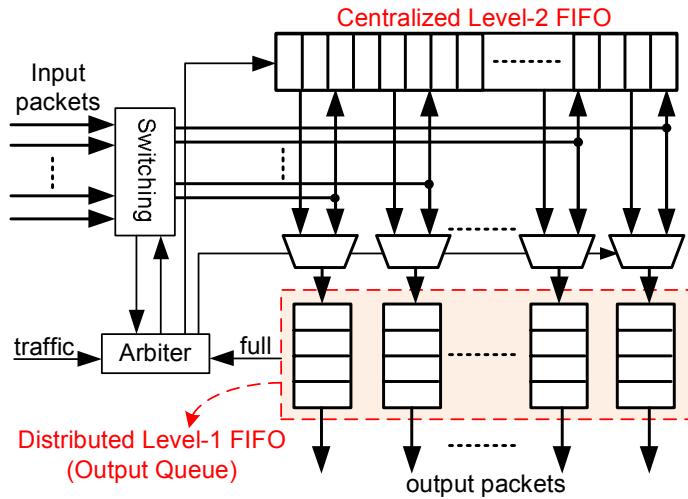


Fig. 4.7 Data flow of two-level FIFO buffer scheme.

and distributed level-1 FIFOs at output channels. Fig. 4.7 illustrates the data flow of the two-level FIFO buffer scheme. The distributed level-1 FIFOs performs output queues for output channels, and the centralized level-2 FIFO is a unified shared buffer for all output channels. The purposes of distributed level-1 FIFOs is to provide a linear increasing of the FIFO sizes to retrieve the fixed sizes of the centralized level-2 FIFO. The operation of the two-level FIFO buffer is described as follows. After switching packets, the packets are dispensed to the distributed level-1 FIFOs of output channels. If the distributed level-1 FIFO is full or congestions exist in an output channel, packets are dispensed to the centralized level-2 FIFO to prevent head-of-line blocking problems. The centralized level-2 FIFO reduces head-of-line blocking problems via a unified shared buffer to increase the OCIN performance. This unified shared buffer is utilized for all input/output channels that can access all memory elements in the shared buffer. Moreover, the multiple-access mechanism of the shared buffer is also provided for all input/output channels to keep the data flows in OCINs. Therefore, the input/output channels can send/get data to/from the shared buffer at the same time slot via multiple accesses of the shared buffer. Additionally, the centralized level-2 FIFO maximizes buffer utilization. In view of the operation of the two-level

FIFO buffer, the arbiter only manages the order of switching packets in output channels.

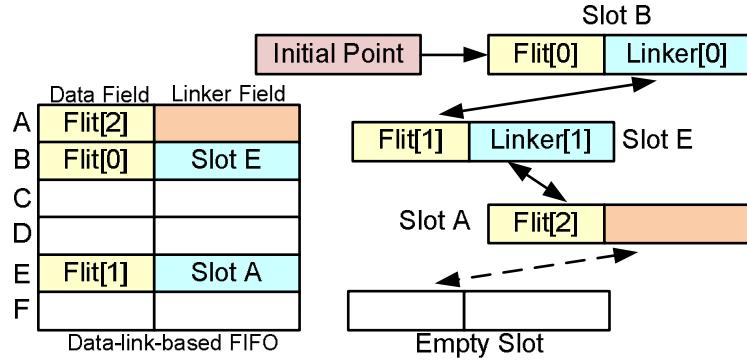


Fig. 4.8 Concept of the data-link-based FIFO.

The centralized level-2 FIFO achieves shared buffering using data-link-based FIFO. Fig. 4.8 presents the concept of the data-link-based FIFO, which takes advantage of data continuity in an FIFO queue. Each slot in the data-link-based FIFO has two stored fields, the data field and linker field. In a slot, the data field stores a flit and linker field stores the address of the next slot, which may not be the adjacent slot in the data-link-based FIFO. In the other words, the linker[i] will store the address of the flit[i+1] in the same FIFO queue. Therefore, the read controller reads the next datum depending on the address stored in the linker field.

The two-level FIFO buffer scheme can be employed at the flit level or packet level depending on flow control techniques, store-and-forward switching, virtual cut-through switching or Wormhole switching [4.27]. Wormhole flow control was proposed to improve performance at the flit level and relaxes the constraints on buffer sizes. Therefore, the wormhole switching technique is the most popular switching technique in packet-switching-based OCINs [4.28]-[4.30]. At the flit level, when more than one packet are sent to the same output, the links between these packets cannot be constructed. Therefore, the two-level FIFO buffer needs an extra linker table to record

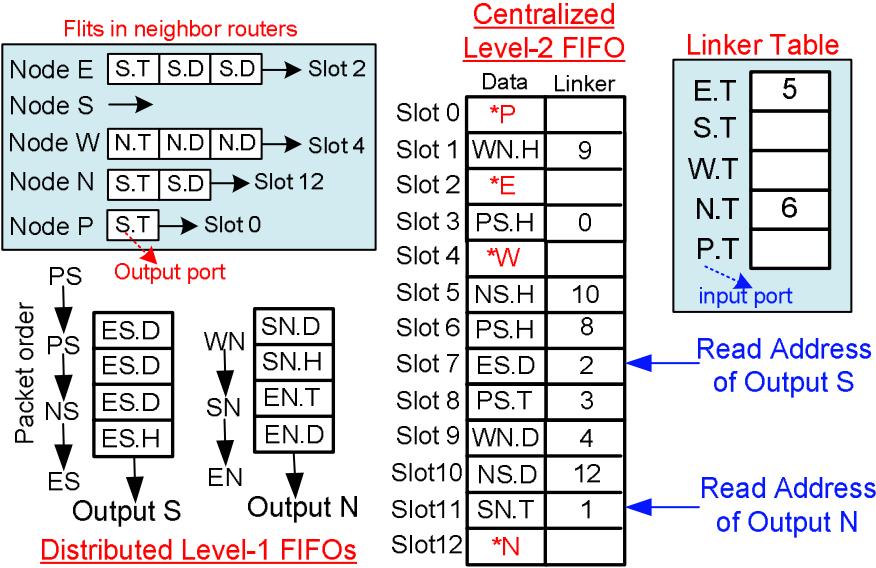


Fig. 4.9 An example of two-level FIFO buffer scheme.

the linked addresses if the tail flit of the front packet is not arrived. Fig. 4.9 gives an example of the two-level FIFO buffer scheme based on a 5input/5output router in a mesh OCIN at the flit level. Therefore, this router is connected to the east router (E), south router (S), west router (W), north router (N), and processor element (P). The flits in the neighbor routers will be dispensed to this router. The first capital letter of a flit indicates the output port of the flit, and the second capital letters (.H, .D and .T) represent the header flit, data flit and tail flit in a packet, respectively. In the two-level FIFO buffer, the first two capital letters indicate the input port and output port of this packet. For example, ES means a packet has an EN turn in this router. In other word, this packet is from the east router, and will be dispensed to the north router. For the output channel S, the packet order is ES–NS–PS–PS, and the centralized level-2 FIFO will construct the links in the linker fields based on the packet order. The linker field will store the address of the linked slot. The read address of each output channel denotes the first flit in the centralized level-2 FIFO. When the router N send a request for this router, the distributed level-1 FIFO will dispense the EN.D flit to the router N, and the flit in the slot 11 will be transferred to the distributed level-1 FIFO. At the

same time, the read address of output N will be changed to slot 1. Additionally, the data flit from the router W will be stored into slot 4 that is linked to slot 9. In this example, the packets form E, N and P are routed to the output S, and the order of these packets is E–N–P. The header flit of packet N should be linked to the tail flit of packet E. However, the tail flit of packet E is not dispensed to this router yet. In view of this, the two-level FIFO buffer scheme needs an extra linker table to reconstruct the link by recording the linker for the tail flit.

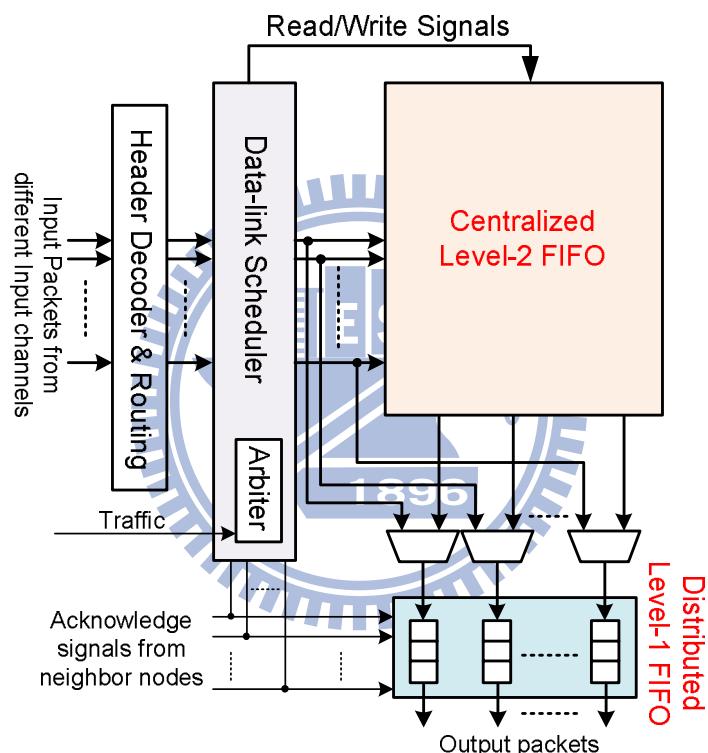


Fig. 4.10 Two-level FIFO buffer architecture in routers.

#### 4.4 Synchronous Two-Level FIFO Buffer Architecture

The two-level FIFO buffer architecture is implemented using register-based buffer and consists of a data-link scheduler, distributed level-1 FIFOs, and a data-link-based centralized level-2 FIFO. Fig. 4.10 shows the architecture of the data-link two-level FIFO buffer. The operation of the two-level FIFO buffer router is briefly described as follows. When input packets arrive to the two-level FIFO buffer architecture, the

header decoder first de-multiplexes input data from header information. The data-link scheduler then schedules empty buffers and sends de-multiplexed data to the centralized level-2 FIFO. The link scheduler records the address of the output buffer in the linker fields. When acknowledge signals are asserted from the next stage, the distributed level-1 FIFO will transfer output data. Moreover, the data-link scheduler transfers the address, which indicates the bottom of the output buffer, to the centralized level-2 FIFO. The centralized level-2 FIFO delivers accuracy data to the level-1 FIFO. The details of the functional blocks in the two-level FIFO buffer architecture are described as follows.

#### 4.4.1 Header Decoder and Routing

The packets delivered from processor elements contain headers and payloads. The headers describe the paths the packets will go through. Header information depends on the routing algorithm and OCIN architecture. The two-level FIFO buffer scheme can be employed for both deterministic routing and adaptive routing algorithms.

#### 4.4.2 Data-Link Scheduler and Centralized Level-2 FIFO

The data-link scheduler and centralized level-2 FIFO are kernel blocks of the two-level FIFO buffer architecture. Fig. 4.11 shows block diagrams of the data-link scheduler and data-link-based centralized level-2 FIFO. The data-link scheduler consists of a write generator, a wordline encoder, a linker table and linker fields that records the addresses of linked data. The centralized level-2 FIFO is constructed using a read controller and data fields. For  $k$  flits in the two-level FIFO buffer, the data fields and linker fields are implemented by  $k$  slots (words), which can be accessed via write control signals (wordlines). Each slot contains  $m$ -bits in the data field and  $\log_2(k)$ -bits in the linker field. Restated, the width of the linker fields is  $\log_2(k)$ -bits to

record the linked addresses. The width of the data fields is  $m$ -bits, and depends on the physical size of a flit.

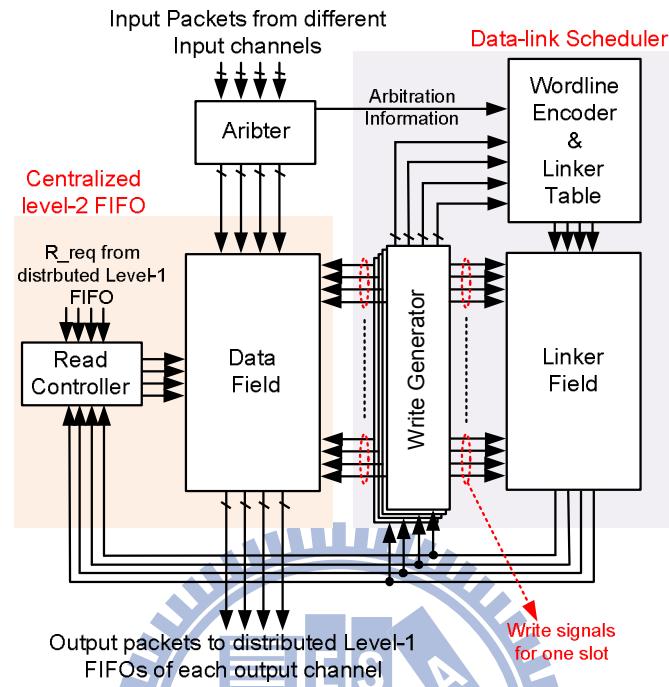


Fig. 4.11 Data-linked based centralized level-2 FIFO and data-link scheduler.

The data-link scheduler creates links among output channels using the write generator and linker fields. The write generator generates the writing wordlines for the data fields to write input flits. While asserting the writing wordlines for the data fields, the linked addresses are produced using the wordline encoder. The wordline encoder encodes these writing wordlines and feeds the encoded addresses (linked addresses) into the linker fields to create links. Therefore, the write generator also latches writing wordlines of the data fields for the linker fields to record the addresses of the next arrival flits. Thus, the switching circuits of the router are utilized in the write generator and data fields based on the link information. Clearly, the read controller obtains addresses from the linker fields to read the next flits of the output channels in the data fields. Hence, the read controller reads the output flits and linked addresses at the same time, and latches the linked addresses for the next transaction. Restated,

when the data have been read from the data fields, the read controller records the reading addresses of the data fields to read the next address of the first-in datum in output queues from the linker fields.

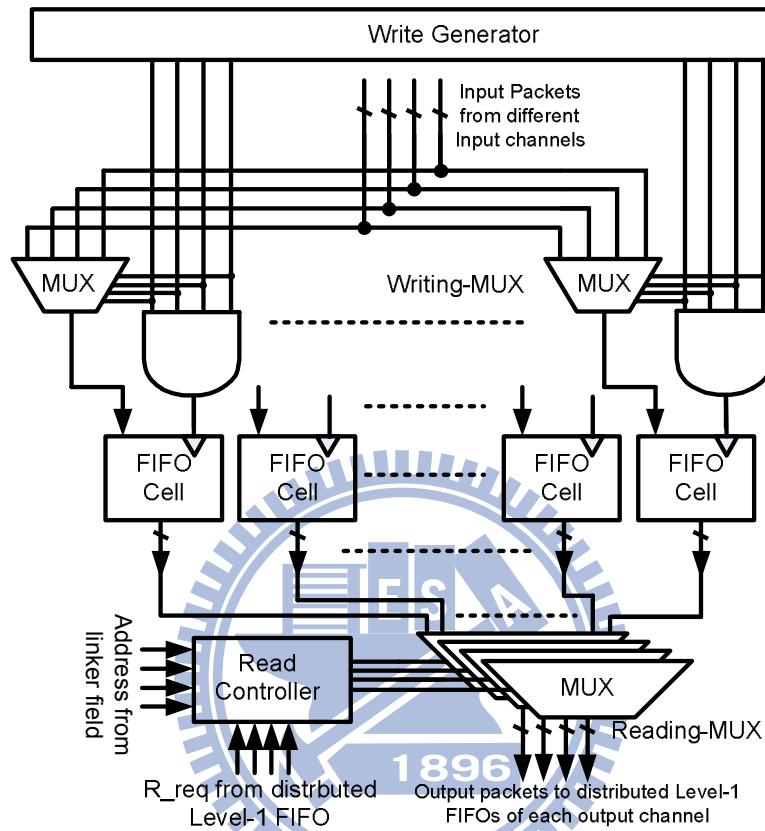


Fig. 4.12 Implementation of the centralized level-2 FIFO.

The centralized level-2 FIFO provides unified shared buffer and a multiple-access mechanism. Fig. 4.12 shows the schematic of the centralized level-2 FIFO. The data fields and linker fields are both implemented by the Bus-MUX-In MUX-Out registers. Each slot (word) of the data fields contains  $m$  bits to store input data. The linker field is constructed using  $\log_2(k)$  bits for storing the addresses of the next datum in queues. The Bus-MUX-In structure provides multiple accesses for the unified shared buffer. Additionally, the Bus-MUX-In structure also performs the switching circuits depending on information from the arbiter and write generator. The arbiter determines the order of packets in an output channel and transfers the routing and arbitration

information to the write generator first. Thus, the write generator switches the Bus-In data into appropriate words by writing wordlines and writing MUXs. Further, depending on switching conditions, the write generator transfers writing wordlines to the wordline encoder and creates links. The read controller and reading MUXs decode link addresses and send output data to the distributed level-1 FIFO.

#### 4.4.3 Distributed level-1 FIFO

The distributed level-1 FIFOs are designed as output queues located in output channels. Hence, the distributed level-1 FIFOs are implemented using Bus-In Mux-Out registers for shallow output queues. The purpose of distributed level-1 FIFOs are to provide a linear increasing of the FIFO sizes to retrieve the fixed sizes of the centralized level-2 FIFO. Therefore, the size of the distributed level-1 FIFO is usually small, and the Bus-In MUX-Out register is preferred. Moreover, the distributed level-1 FIFOs pre-fetch flits from the centralized level-2 FIFO and to keep the data flow when other output channels are congested.

#### 4.4.4 Arbiter

The arbiter determines the order of multiple accesses in the same cycle. When more than one packet at different input ports require the same output port, the arbiter prioritizes the packets. The arbitration algorithm, however, relies on buffer sizes. When buffer size is insufficient, the complexity of the arbiter algorithm increases to eliminate traffic problems. The two-level FIFO buffer architecture provides sufficient buffer sizes using the shared buffer mechanism and multiple accesses for output buffers. That is, the arbiter only decides the order of packets from different input channels when the header flits of these packets are arrival at the same time.

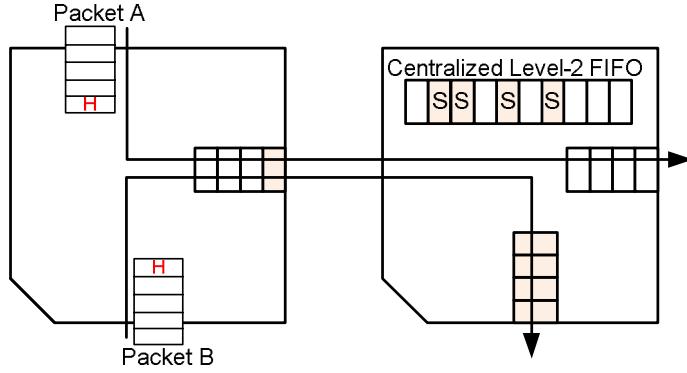


Fig. 4.13 Example of the arbitration policy in deterministic routing algorithms.

The design of the arbiter in the two-level FIFO buffer depends on the characteristic of the routing algorithm. For deterministic routing algorithms, the arbiter can decide the packet order based on the traffic information in the next router. Fig. 4.13 give an example of the arbitration policy, both packet A and packet B are routed from the left router to the right router. However, the output channel of packet B is congested. If the priorities of packet A and B are at the same level, the order of packet A is in front of packet B. For adaptive routing algorithms, the output of next router cannot be determined in this router. To avoid starvation with low-priority packets and ensure transmission speed of high-priority packets, the two-level FIFO buffer architecture uses the time division multiple access (TDMA) arbitration algorithm, which can be implemented by a counter to transfer priorities for successive input ports. Packet priority determines the position of a packet in the output channel.

## 4.5 Asynchronous Two-Level FIFO Buffer Architecture

There are major problems in having various synchronous on-chip communications, including modularity and design reuse, electromagnetic interference (EMI), worst case performance, clock power consumption and Clock skew [4.31]. Asynchronous NoC platforms are proposed for globally-asynchronous locally-synchronous (GALS) systems to achieve energy efficiency, quality-of-service and clock de-skewing

[4.32]-[4.34]. In GALS systems, data synchronization and communication across different clock domains are extremely challenging. An alternative solution is provided by GALS interface and an asynchronous network [4.35]-[4.38].

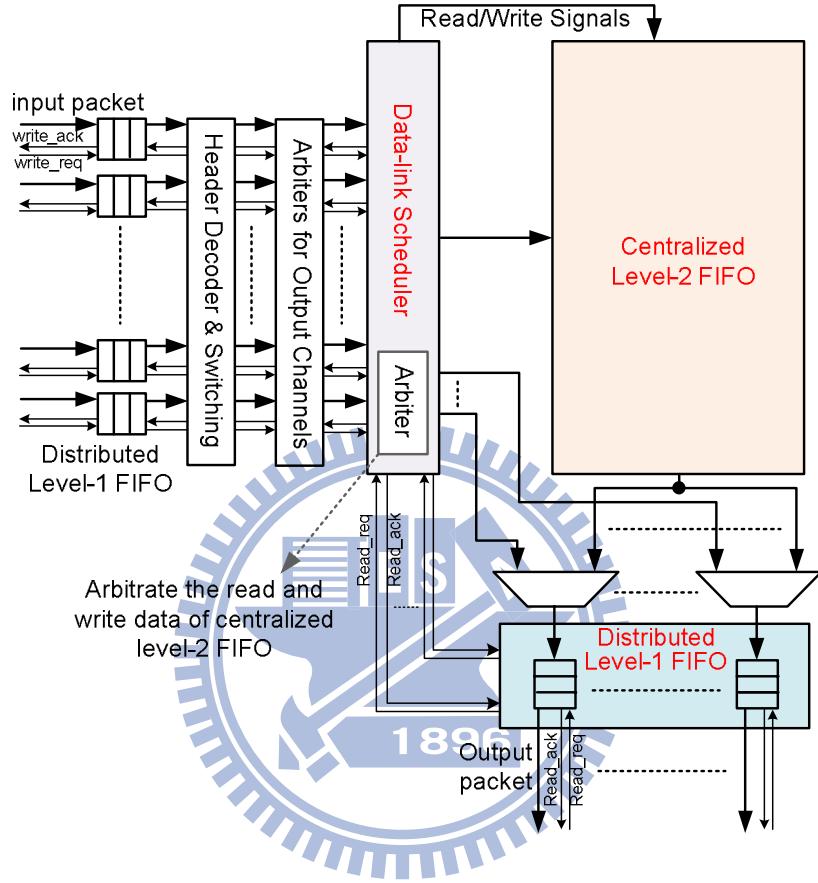
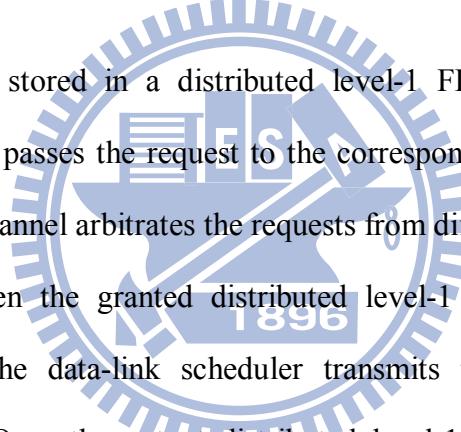


Fig. 4.14 Asynchronous two-level FIFO buffer architecture.

The two-level FIFO buffer is also employed for an asynchronous router, and the behavior of the asynchronous two-level FIFO buffer is similar to that of the synchronous two-level FIFO buffer. Fig. 4.14 shows the architecture of the asynchronous two-level FIFO buffer. However, the main difference between the synchronous and asynchronous two-level FIFO buffers is that the synchronous two-level FIFO has a multiple access mechanism. In terms of the different frequencies of each local clock domain, the arrival times of data from different input channels cannot be predicted. Thus, the asynchronous two-level FIFO buffer must be time-insensitive. Therefore, implementing the level-2 FIFO with multiple access

mechanism is unnecessary. Additionally, the asynchronous two-level FIFO buffer requires a read arbiter and write arbiter to control read and write access, respectively. Communications in the two-level FIFO architecture are asynchronous between the distributed level-1FIFOs and centralized level-2 FIFO via a four-phase handshaking protocol. Since the depth of the distributed level-1 FIFOs are shallow, these FIFOs are implemented as asynchronous FIFOs for input and output channels. These FIFOs are implemented by the Bus-In Bus-Out structure to provide low latency and high throughput FIFO cells. The read/write order depends on the token in circular FIFO cells. Further, the distributed level-1 FIFOs connect GALS wrappers and the asynchronous two-level FIFO buffer.



When input data is stored in a distributed level-1 FIFO, the header decoder decodes the header and passes the request to the corresponding output channel. The arbiter of each output channel arbitrates the requests from different input channels and creates the link between the granted distributed level-1 FIFO and the data-link scheduler. Therefore, the data-link scheduler transmits the granted data to the centralized level-2 FIFO or the output distributed level-1 FIFO depending on the buffering condition in the output channel. If no data exist in the centralized level-2 FIFO for this output channel and the distributed level-1 FIFO is not full, data are transmitted to the distributed level-1 FIFO. Restated, if the distributed level-1 FIFO is full or the centralized level-2FIFO for this output channel is not empty, the data are delivered to the centralized level-2 FIFO. For single access of the centralized level-2 FIFO, the data-link scheduler also requires a read arbiter and write arbiter to arbitrate congestion on the read and write ports.

The arbitration circuitry must deal with concurrent and even simultaneous requests, and should not grant access to more than one sender at any time. Furthermore, to

to avoid confusion, the grant signal must be free of hazards and not be in other meta-stable states. The arbitration for the four-request asynchronous architecture can be performed using a MUTEX-based scheme, which is called MUTEX-NET [4.33]. When four requests arrive from different senders, the MUTEX-NET guarantees that the first request and last request are granted in their original order, but the other two requests may not retain their order.

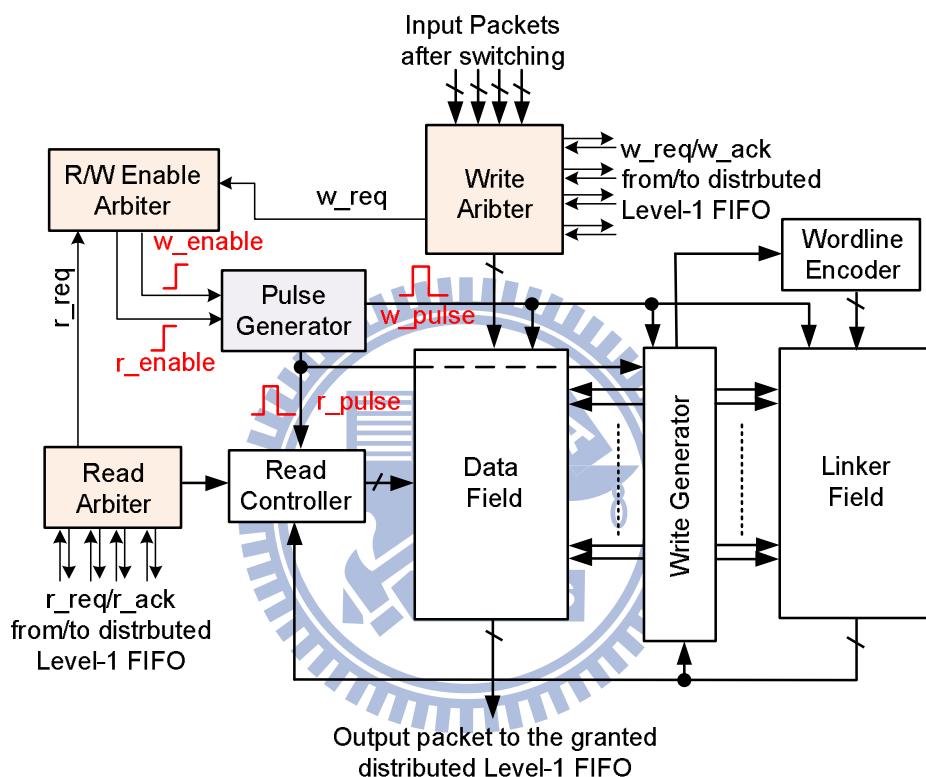


Fig. 4.15 Data-link scheduler and centralized level-2 FIFO for asynchronous two-level FIFO buffer.

Fig. 4.15 shows the details of the data-link scheduler and centralized level-2 FIFO for the asynchronous two-level FIFO buffer. The write and read arbiters determine access permission of the centralized level-2 FIFO for different output channels when more than two output channels issue requests. The R/W-enable controller produces R/W-enabled signals for the centralized level-2 FIFO based on read\_req and write\_req, which are requests from the read port and write port, respectively. Because the write generator must identify the written word for the next transaction according to the

occupied condition of the centralized level-2 FIFO, the read pulse and write pulse must be separated. That is, the centralized level-2 FIFO cannot be read and written simultaneously. Therefore, the R/W-enabled arbiter permits the reading operation or writing operation by a MUTEX-based arbiter. Furthermore, in an asynchronous two-level FIFO buffer, implementation of the write generator is similar to that of a synchronous two-level FIFO buffer without multiple slides. Moreover, the pulse generator generates the read pulse and write pulse for the read and write operations.

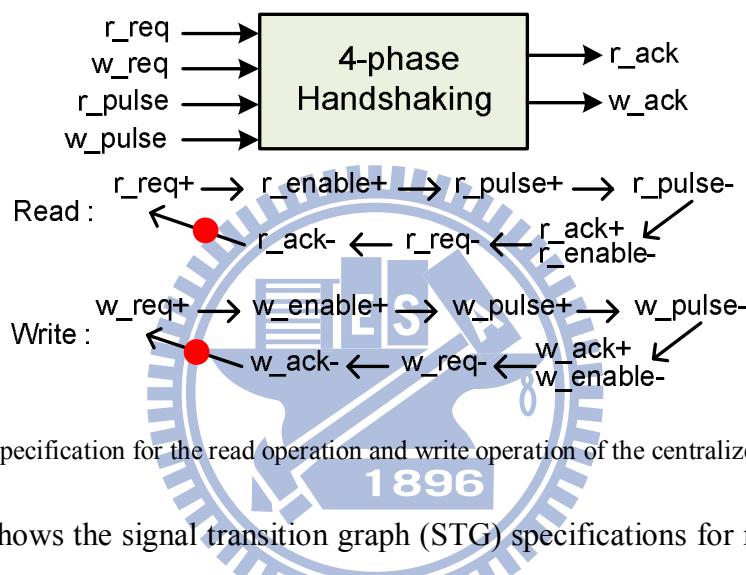


Fig. 4.16 STG specification for the read operation and write operation of the centralized level-2 FIFO.

Fig. 4.16 shows the signal transition graph (STG) specifications for read and write operations. The STG specifications show the four-phase handshaking protocol for the centralized level-2 FIFO buffer. For a read operation, while  $r\_req$  is asserted, the R/W-enables arbiter will assert  $r\_enable$  when  $w\_req$  is not asserted. The pulse generator then generates a pulse ( $r\_pulse$ ) to read data and reading control signals for the next reading transaction. Depending on the falling edge of a pulse, the 4\_handshaking circuitry will activate  $r\_ack$  to acknowledge the signal for the distributed level-1 FIFO to de-assert the request. Additionally, the acknowledge signal is pulled down by the falling request. The handshaking of the write operation is the same as that of the read operation.

## 4.6 Associated Two-Level FIFO Buffer Architecture

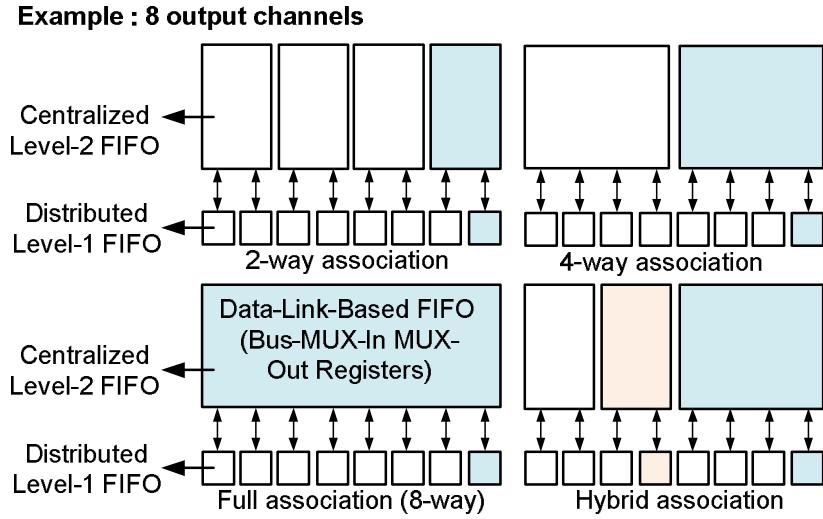


Fig. 4.17 Different associations between the distributed level-1 FIFO and the centralized level-2 FIFO for 8 output channels.

The two-level FIFO buffer architecture has a unified shared buffer to eliminate head-of-line problems by the data-link-based FIFO and multiple-access mechanism. However, based on the register-based buffer, the power and area overhead of multiple accesses for the centralized level-2 FIFO increases rapidly as the number of output channels increases. Therefore, a trade-off exists between buffer utilities and the power overhead of multiple accesses. That is, the centralized level-2 FIFO can be divided into subgroups for specific output channels. Fig. 4.17 shows the different associations between the distributed level-1 FIFO and centralized level-2 FIFO for 8 output channels. The centralized level-2 FIFO can be deconstructed into different subgroups—two-way association, four-way association, full association or hybrid association. The higher association between the level-2 FIFO and level-1 FIFO will increase buffer utilities of the two-level FIFO buffer. That is, each output channel can access an increased number of buffers in the higher association. Moreover, the physical size of the linker field decreases with the increasing association. Assume the number of available buffer in the centralized level-2 FIFO is  $k$  slots. For the  $m$ -way

association two-level FIFO buffer in an n-port router, the total size of the linker fields is as Eq. (4.1).

$$\text{Total Linker Size} = \frac{n}{m} \log_2 \left( \frac{k}{n/m} \right) \quad (4.1)$$

## 4.7 Simulation Results

### 4.7.1 Synchronous Two-Level FIFO Buffer

In this section, a cycle-driven simulator is used to evaluate different buffer architectures in SystemC, including output buffer, middle buffer, ViChaR [4.19], distributed shared buffer (DSB) [4.26] and the proposed two-level FIFO buffer. The middle buffer architecture establishes multiple virtual channels during switching to reduce head-of-line problems via static virtual channel allocation. The ViChaR architecture provides unified buffer structures at input ports as dynamical virtual channels. Additionally, different buffer architectures are also evaluated with different routing algorithms, including XY routing, DyXY [4.39] and an adaptive routing [4.40].

The number of pipeline stages in a router depends on the buffer architecture. Fig. 4.18 presents the pipeline stages of different buffer architectures, and the link traversal (LT) stage indicates flits traverse the link wires to arrive the downstream router. The middle buffer and ViChaR are realized in 4-stage pipeline routers comprising router computing (RC), virtual channel allocation (VC), switching allocation (SA) and switch traversal (ST). The difference between the middle buffer and ViChaR is in the VC stage. The output buffer router also realizes a 4-stage pipeline consisting of RC, arbitration, SA and ST. The DSB provides a centralized

shared buffer to increase performance. However, an extra pipeline stage is added into the DSB router that has a five-stage pipeline comprising RC, timestamping (TS), conflict resolution (CR) and VA, first switching traversal (ST1) and middle memory writing (MM\_WR), and middle memory reading (MM\_RD) and second switching traversal (ST2). The proposed data-link two-level FIFO buffer also provides a centralized shared buffer without inserting extra pipeline stage. Therefore, the 4 pipelining stages include RC, arbitration and write wordlines generating & encoding (W\_Gen), data buffer writing (Data\_W) and data link creating (Link\_W), and data buffer reading (Data\_R) and linker reading (Link\_R). Moreover, the switching circuit is concealed in write wordlines generating and data link creating as described in Section 4.4.

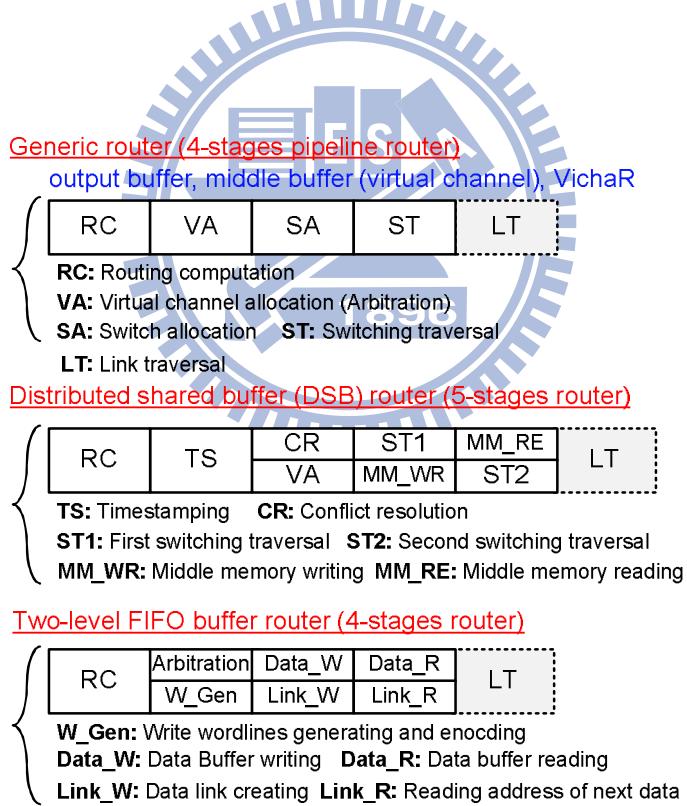


Fig. 4.18 Pipeline stages of the generic router, DSB router and two-level FIFO buffer router.

According to the cycle-driven simulation in SystemC, Fig. 4.19 shows the performance of output buffer, middle buffer, ViChaR, DSB and two-level FIFO buffers (including 2-3 hybrid association and full association) with different buffer

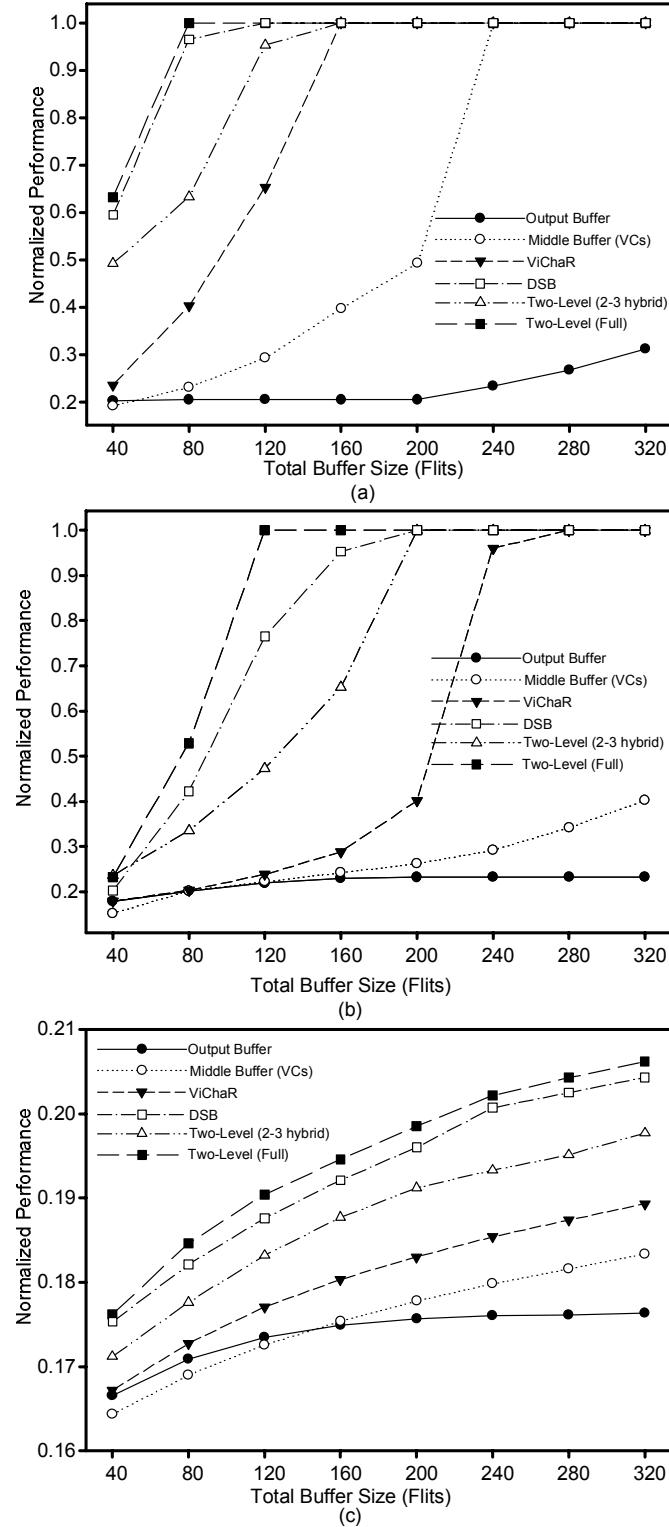


Fig. 4.19 Normalized performance versus FIFO sizes with different buffer organizations in (a) low injection load (b) medium injection load (c) high injection load.

sizes. The simulation environment is an  $8 \times 8$  mesh network with an X-Y routing algorithm and uniform traffic patterns. Each packet contains 2, 4 or 8 flits randomly.

In a mesh network, each router has 5 inputs and 5 outputs. In a ViChaR router, each input channel has a unified buffer structure with the same buffer size of the input/output buffer. For the middle buffer, each input port has 4 virtual channels. Therefore, the depth of each virtual channel is from 2-flit to 16-flit in this simulation. For the two-level FIFO buffers, the distributed level-1FIFO is set as 2-flit for each output channel. The 2-3 hybrid associated two-level FIFO buffer divides the centralized level-2 FIFO into two subgroups. One is shared by the east and west ports, and the other is shared by the north port, south port and processor element. Figs. 15(a)–(c) show the simulation results under different injection loads of low (0.15), medium (0.25) and high (0.35), respectively. Performance is normalized to the throughput of infinite buffers within constant cycles. The two-level FIFO buffer architecture performs best with the same size of other buffer architectures regardless of injection loads. For the ViChaR, the unified buffer structure shares buffers in virtual channels for each input port. Thus, the unified control logic controls arriving/departing pointers and virtual channel allocation of each virtual channel through virtual channel control tables and dispensers. However, when the shared buffer of an input port is full, the ViChaR does not provide a mechanism for accessing buffers of other virtual channels in other input ports. For the centralized shared buffer, the performance of the DSB buffer is similar to that of the fully associated two-level FIFO buffer. Additionally, when the total buffer size is small, the performance of the middle buffer is worse than that of the output buffer. The reason of this phenomenon is due to shallow virtual channels. In high injection load, the throughput of different buffer architectures is quite smaller than that of infinite buffers because the performance is dominated the heavy congestion of the network. Compared to the traditional router with middle buffers, the total buffer size of the two-level FIFO buffer can be reduced to 20%–25% for achieving the same performance.

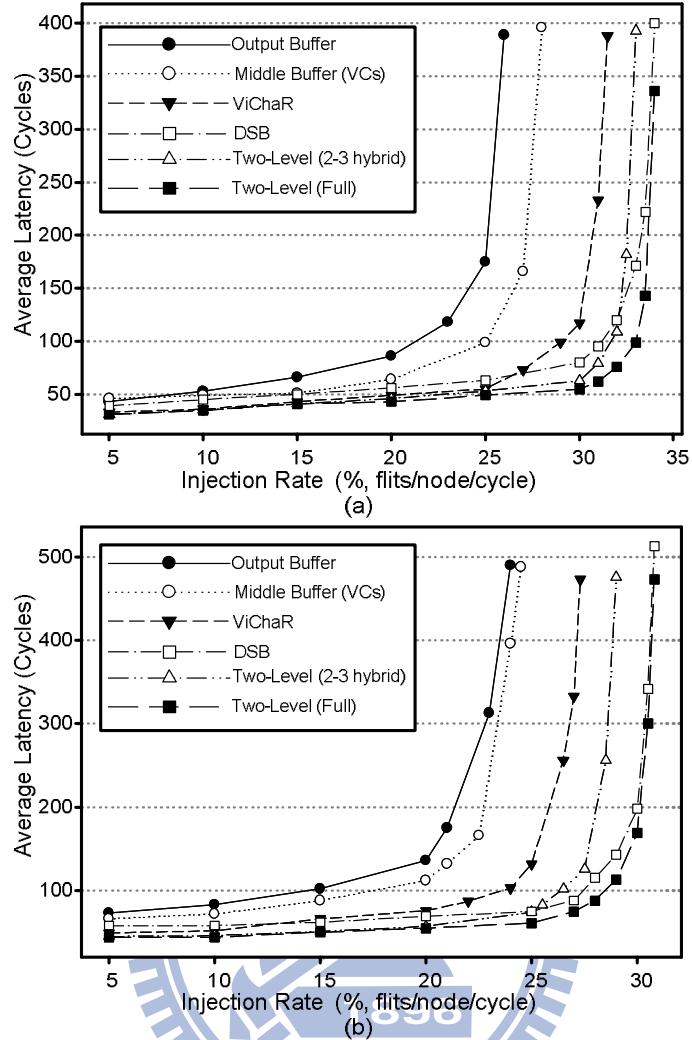


Fig. 4.20 Average latencies of different buffer architectures in (a) uniform patterns (b) hotspot patterns.

Different buffer architectures are evaluated by another metric of network performance, namely average latency. Fig. 4.20(a) and Fig. 4.20(b) present the average latencies of different buffer architectures with uniform patterns and hotspot patterns, respectively. The simulation environment is an  $8 \times 8$  mesh network with an X-Y routing algorithm, and each packet contains 4 or 8 flits randomly. In Additional, the total buffer size is set as 160 flits. In hotspot traffic, uniform traffic is applied, but then 30% of the packets change their destination to one of six nodes (2,3),(2,4),(3,3),(3,4),(6,5),(6,6) with equal. Compared with conventional output buffer and middle buffer, the ViChaR, DSB and two-level FIFO buffer architectures can achieve the lower average latencies no matter what injection load is. In lower

injection load, the average latencies of the DSB buffer are larger than those of ViCharR and two-level FIFO buffer because inserting one extra pipelining stage. With the increasing injection load, the average latencies of DSB buffer are reaching to those of two-level FIFO buffer because the latencies are dominated the heavy congestion of the network. Moreover, the fully associated two-level FIFO buffer can realize the lowest average latencies compared to other buffers. Nevertheless, the boundaries of shared buffers, including ViChar and two-level FIFO buffer, decrease significantly in hotspot patterns. Restated, the shared mechanism cannot lighten the traffic efficiently with hotspots.

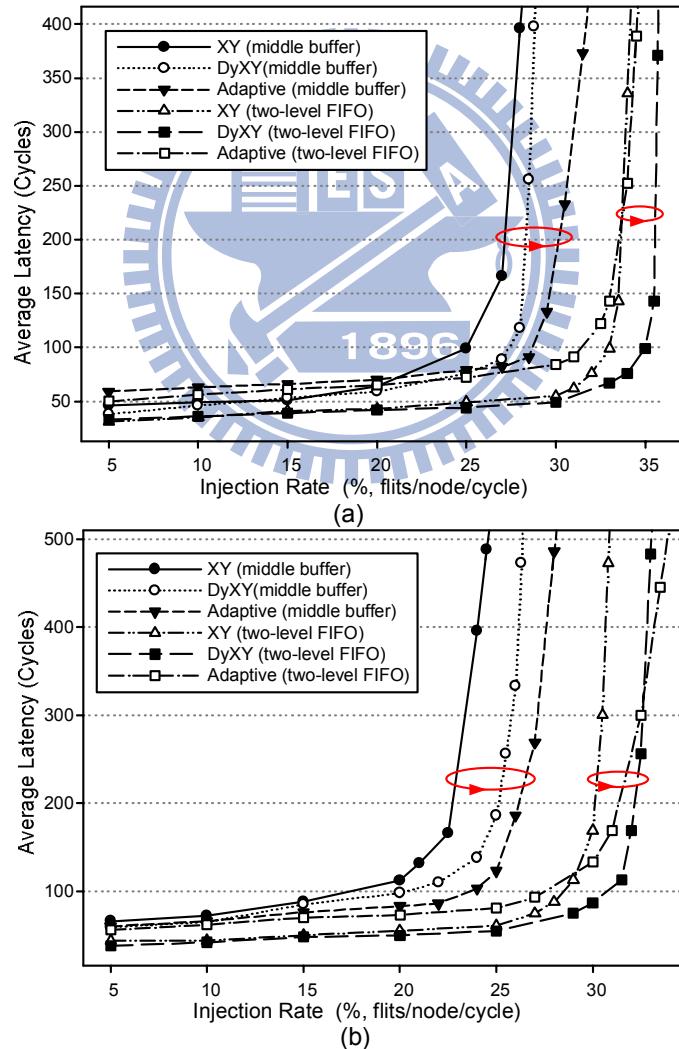


Fig. 4.21 Average latencies of XY, DyXY and adaptive routing algorithms in (a) uniform patterns (b) hotspot patterns.

After comparing the buffer architectures of different routing algorithms, Fig. 4.21 presents the average latency under different routing algorithms with the middle buffer and fully associated two-level FIFO buffer architectures. The routing algorithms are XY routing and DyXY [4.39] routing algorithms and an adaptive congestion-aware routing algorithm [4.40]. In DyXY and adaptive routing algorithms, the two-level FIFO buffer uses the TDMA arbiter described in Section 4. These graphs follow the same trend as the latency simulations. Fig. 4.21(a) and Fig. 4.21(b) show the average latencies with uniform random patterns and 6 hotspots in the center region, respectively. The two-level FIFO buffer reduces the influence of performance on average latencies induced by the routing algorithms. In addition, the DyXY routing algorithm with a two-level FIFO buffer performs better than the adaptive algorithm with a two-level FIFO buffer. Moreover, the adaptive routing algorithm increases the average latencies when the injected load is low. However, the adaptive algorithm can achieve the lowest average latencies with high injected load in hotspot patterns.

Table 4.1 Area and power comparisons between different buffer architectures in the same buffer size.

Buffer architecture	Area ( $\mu\text{m}^2$ )	Power (mw)
Middle buffer (Bus-in MUX-out, 160 flits)	60524.3(0%)	16.94(0%)
Middle buffer (Bus-in Bus-out, 160 flits)	56832.3(-6.1%)	18.61(+9.9%)
ViChaR (160 flits)	79322.6(31.1%)	23.37(+30.0%)
DSB (160 flits)	88235.3(+45.7%)	25.31(+49.4%)
Fully associated two-level FIFO buffer (158 flits)	83654.8(+38.2%)	23.91(+41.1%)
2-3 hybrid associated two-level FIFO buffer (64 flits for each subgroup, 158 flits)	79053.8(+30.6%)	21.54(+27.2%)

The two-level FIFO buffer architecture is implemented via Synopsys Design Compiler and PrimePower to estimate the area and power consumption based on

UMC 65nm CMOS technology at 1.0V and 1GHz. Table 4.1 lists the area and power consumption of routers with different buffer architectures for similar buffer sizes. These buffer architectures include the middle buffer using the static virtual channel allocation, a dynamic virtual channel regulator (ViChaR), DSB and two-level FIFO buffer architectures. The middle buffer architectures are implemented as the Bus-In MUX-Out and Bus-In Bus-Out registers, respectively. The middle buffer is implemented as 5 input ports with 4 virtual channels for input port; each virtual channel has 8 flits and each flit size is 64 bits. Therefore, the total number of flits for each router is 160 flits ( $5 \times 4 \times 8$ ). The ViChaR has a unified buffer structure that dynamically allocates virtual channels and buffer resources according to network traffic patterns. The ViChaR is composed of a unified buffer structure and unified control logics, which are the arriving/departing pointers and the VC control table. In each input port, the size of the unified buffer structure is 32 flits. For fully associated two-level FIFO buffer architecture, the centralized level-2 FIFO is implemented as Bus-MUX-in MUX-out registers; the centralized level-2 FIFO is 128 flits (words) deep and 64 bits wide. For the 2-3 hybrid associated centralized level-2 FIFO, the depth is 64 flits (words) and width is 64 bits for each subgroup. In order to obtain the same size of two-level FIFO buffers for different buffer architectures, each distributed level-1 FIFO has 6 flits with 64 bits that linearly increase FIFO sizes to determine the fixed sizes of the centralized level-2 FIFO. Therefore, the total number of flits in the fully associated and 2-3 hybrid association is 158 flits ( $128 + 6 \times 5$ ,  $64 \times 2 + 6 \times 5$ ). For a similar number of flits, the DSB occupies the largest area compared with those of other buffer architectures because of two switch circuits, complex arbitration and great amount of lookup tables. The proposed two-level FIFO buffer architecture induces 38.2% area overhead since multiple accesses of the centralized level-2 FIFO. Nevertheless, the two-level FIFO buffer architecture dissipates the smaller power than

the DSB and ViChaR because the VC control table dissipates more power than the data-link scheduler. Although the size of the linked field in the hybrid associated two-level FIFO buffer is larger than that in the fully associated two-level FIFO buffer, the hybrid association uses less power and area overhead by reducing the number of multiple accesses. This is a trade-off between performance and power consumption. Therefore, Fig. 4.22 presents the power and area analysis of different associated two-level FIFO buffers corresponding to an 8input/8output router. Both the power and area are reduced when the association decreases. With the decreasing association, the number of multiple accesses in each sub-group also decreases but the size of linker fields increases. Therefore, the power and area overheads of the linker fields are both smaller than those of the multiple-access mechanism.

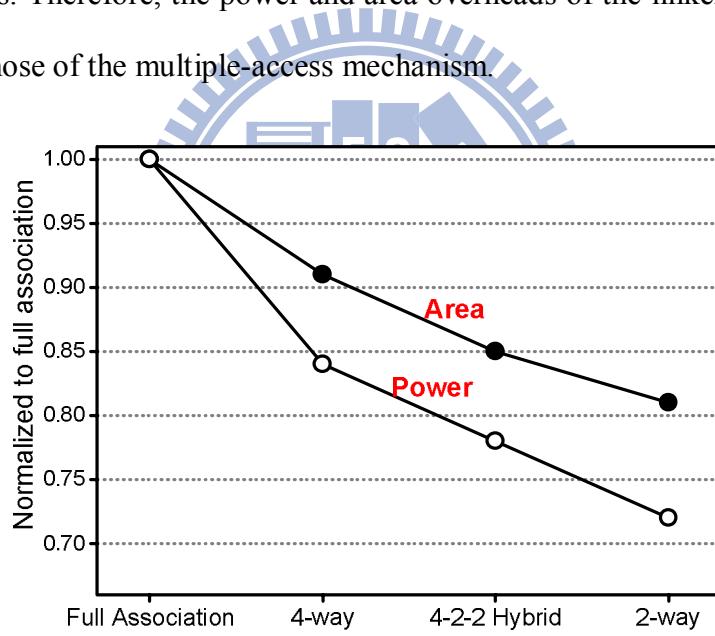


Fig. 4.22 Power and area analysis of the different associated two-level FIFO buffers in an 8input/8output router.

The proposed fully associated two-level FIFO buffer can achieve performance similar to that of the conventional virtual channels, while using 20%-25% buffers. Therefore, the area and power consumption of the middle buffer, ViChaR, DSB and two-level FIFO buffer are also analyzed under similar performance as list in Table 4.2. The ViChaR uses half buffers (80 flits) to realize the similar performance.

Consequently, DSB and two-level FIFO buffer achieve the similar performance using one-fourth buffers (40 flits), and each flit size is 64 bits. Based on UMC 65nm CMOS technology at 1.0V and 1GHz, the ViChaR, DSB and proposed two-level FIFO buffer can achieve 9.3%, 19.2% and 22.3% power reduction, respectively.

Table 4.2 Area and power comparisons between different buffer architectures with similar performance.

Buffer architecture	Area ( $\mu\text{m}^2$ )	Power (mw)
Middle buffer (Bus-in MUX-out, 160 flits)	60524.3(0%)	16.94(0%)
ViChaR (80 flits)	52213.6(-13.7%)	15.37(-9.3%)
DSB (40 flits)	47771.6(-21.1%)	13.69(-19.2%)
Fully associated two-level FIFO buffer (40 flits)	44251.3(-26.9%)	13.17(-22.3%)

#### 4.7.2 Asynchronous Two-Level FIFO Buffer

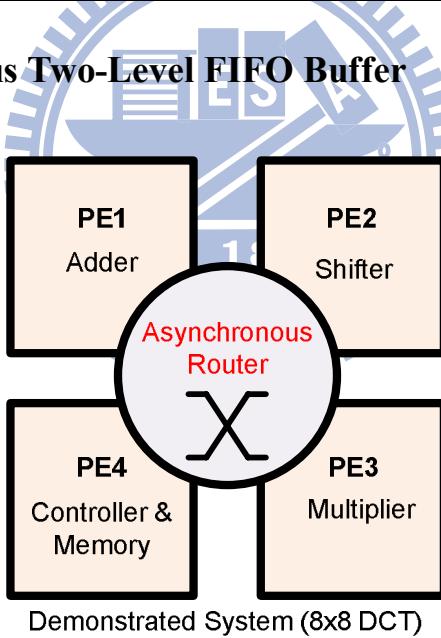


Fig. 4.23 The demonstrated 8x8 DCT system for an asynchronous router.

To estimate performance of the proposed asynchronous two-level FIFO buffer, a fixed-point 8×8 discrete cosine transform (DCT) is demonstrated. This demonstrated 8x8 DCT is performed with an 8-point 1-dimensional DCT along x-axis and y-axis individually. Fig. 4.23 shows the setup of the fixed-point 8x8 DCT with an

asynchronous router using UMC 90 nm CMOS technology. Additionally, the  $8 \times 8$  DCT system is divided into four parts (PE1–PE4), and the data communication between PEs depends on the asynchronous router. PE1, PE2 and PE3 are arithmetic units, an adder, a shifter and a multiplier, respectively. PE4 is a DCT controller and a 256x16 asynchronous SRAM. The asynchronous router is implemented by the output buffer, middle buffer and two-level FIFO buffers (two-way association and full association). Each packet contains 2 or 4 flits (including the header flit), and the width of each flit is 16 bits for this demonstrated system. The total size of the output buffer and middle buffer is 24 flits. The size of the centralized level-2 FIFO buffer is 16 flits. Thus, the total size of the distributed level-1 FIFO is 8 flits (2+2+2+2). For the two-way association, PE1 and PE2 are grouped into a subgroup; the other subgroup contains PE3 and PE4. Therefore, the size of the centralized level-2 FIFO buffer in each subgroup is 8 flits.

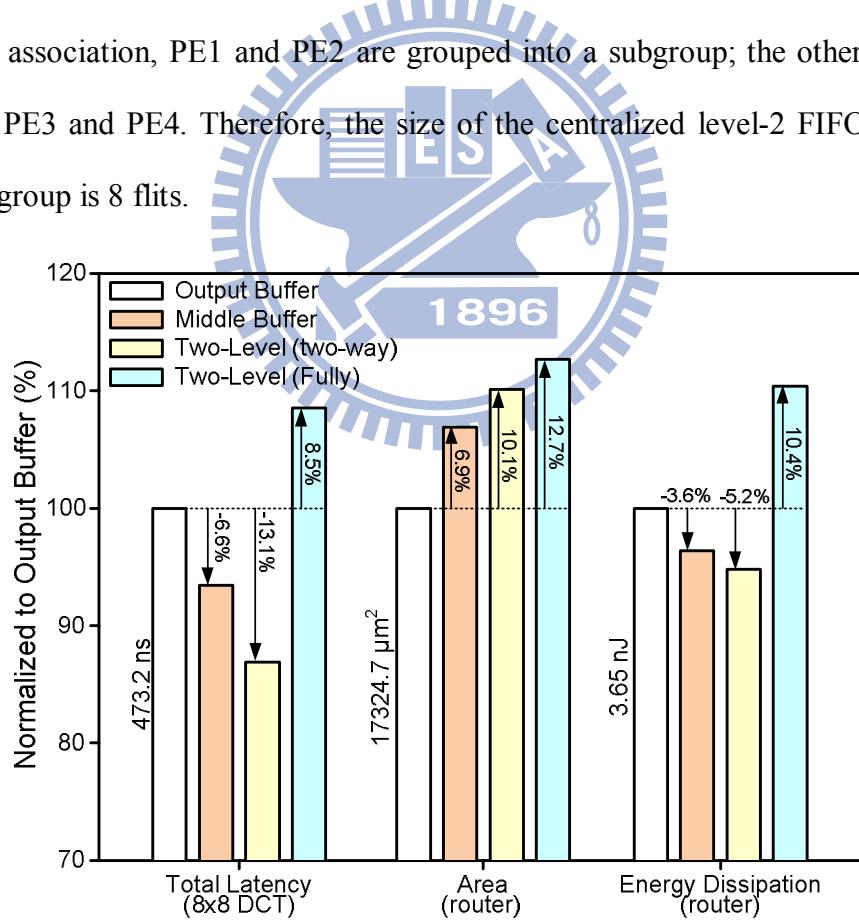
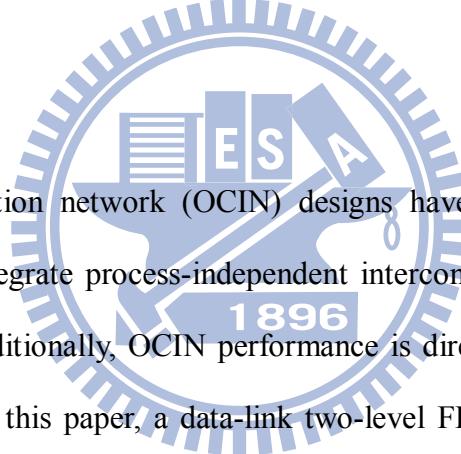


Fig. 4.24 Latencies, area and energy dissipations for an  $8 \times 8$  DCT with different buffers.

Fig. 4.24 illustrates total latencies of the  $8 \times 8$  DCT, area and energy dissipations of the asynchronous router with different buffer architectures. Additionally, the latency,

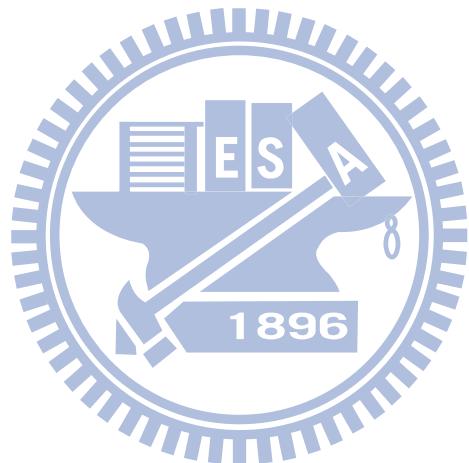
area and energy are normalized to the output buffer. And thus, the latency, area and energy dissipation of the output buffer are 473.2ns,  $17624.7\mu\text{m}^2$  and 3.65nJ, respectively. According to the increasing latency, area and energy of the fully associated two-level FIFO buffer, this buffer architecture is not adapted to the asynchronous router due to shallow level-1 FIFOs and the limited bandwidth of the centralized level-2 FIFO. The tow-way associated two-level FIFO buffer reduces total latency and energy consumption by 13.1% and 5.2% compared with those of traditional output buffers due to the decrease in head-of-line problems. Nevertheless, the area overhead of the two-way association is 10.1% occupied by extra arbiters and the data-link scheduler.

## 4.8 Summary



On-chip interconnection network (OCIN) designs have been considered as an effective solution to integrate process-independent interconnection architectures and multi-core systems. Additionally, OCIN performance is directly related to the buffer sizes and utilization. In this paper, a data-link two-level FIFO buffer architecture is presented as a good solution for routers in OCINs based on a shared buffer mechanism and multiple accesses. Additionally, the centralized level-2 FIFO is realized via a data-link scheduler. This buffer architecture with a small buffer size reduces the magnitude of head-of-line blocking problems and performs well. According to the cycle-accurate simulator, the two-level FIFO buffer can realize performance similar to that of the conventional virtual channels, while using 20%-25% buffers. Based on UMC 65nm CMOS technology, the proposed data-link two-level FIFO buffer can achieve about 22% power reduction compared with the similar performance of the conventional virtual channels. In addition to the asynchronous router using UMC 90nm CMOS technology, the two level FIFO buffer

can achieve 13.1% latency reduction compared to the output buffer in a demonstrated 8×8 DCT system. The asynchronous two-level FIFO buffer also realizes 5.2% energy saving of the asynchronous router. The two-level FIFO buffer architecture is very useful as alternative design that increases the performance of routers in OCINs.



# ***Chapter 5: Adaptive Congestion-Aware Routing Algorithm for Mesh On-Chip Interconnection Networks***

In this chapter, an adaptive congestion-aware routing algorithm is proposed for mesh on-chip interconnection networks (OCINs). In mesh OCINs, data of each processor elements (PEs) are communicated by packet switch techniques. The destinations of the packets are determined by the routing algorithm, which plays a key role to determine the performance and power consumption. Depending on the traffic around the routed node, the proposed routing algorithm provides not only minimum paths but also non-minimum paths for routing packets. Both minimum and non-minimum paths are based on the odd-even turn model to avoid deadlock and livelock problems. The decision of the minimum paths or non-minimum paths depends on the utilities of buffers in neighbor nodes and the specific switching value. In this adaptive algorithm, the congestion conditions and distributed hotspots can be avoided. It has the advantages of higher performance and lower latency. From the simulation results, it clearly shows that the adaptive congestion-aware routing algorithm is superior to other algorithms for the mesh OCINs.

## **5.1 Background**

Multi-core system-on-chip (SoC) designs provide the integrated solutions for the applications in communications, multimedia and consumer electronics. Due to the requirements of high speed and low power on-chip communication are growing continuously, OCINs are introduced to migrate challenges caused by the bus interconnect technology and the complexity of next generation SoC designs [5.1],

[5.2]. OCINs have a few distinctive characteristics, namely low communication latency, energy consumption constraints and design-time specialization [5.3]. The motivation of establishing OCIN is to achieve high performance by using a system perspective of communication. Systems based on the generic OCIN architecture are organized as unities of nodes, where each node has its own PE, local buffers and network interface (NI). The nodes communicate with others by sending messages through routers. To deal with the routing messages in the network, each node is connected to a router to transfer the messages to the destinations.

Routers are the essential elements of OCIN components which are key design components for the integrated implementation. A router consists of a set of input buffers, an interconnect matrix, a set of output buffers and control circuitries, which are a routing controller, an arbiter, an error detector and so on. The router controller controls the links between input and output channels, which connect this router to the neighboring routers. For routing the messages efficiently, the algorithms of the routing controller determine the shortest paths to optimize the network performance according to the information of the system. In addition, the arbiter is also an important component in the router. It is used for arbitrating more than two packets which want to traverse the same path. The objective of arbiter is to determine the priority of packets for the output channels to increase the bandwidths of the network. Therefore, the routing algorithm and the arbitration mechanism dominate the overall performance of the OCINs.

The routing algorithm is a key factor to achieve high efficiency of on-chip communication. The routing algorithm determines the paths where the delivered packets are transferred to the destinations. In order to evaluate the routing algorithms, the designers consider several potentially conflicting metrics which need to be

balanced, including power consumption, area overhead, performance and robustness. For the performance issue, the routing algorithms have to reduce the latency and to maximize the traffic utilization. The latency will be affected by the congestion and contention problems. Hence, handling resources spatially and temporally can prevent the contention, which overloads the resources individually, and the congestion, which is caused by packets collection. For the robustness to traffic changes, the routing algorithms should behave balanced to a large spectrum of traffic conditions.

Another major constraint for the routing algorithms is assuring the freedom from deadlock and livelock [5.4]. The definition of deadlock is that a packet is blocked at some intermediate resource and it does not reach its destination. Deadlock occurs when one or more packets in the network are blocked and keep blocked for an indefinite time, waiting for an event that can't happen. The prominent strategy for dealing with deadlock is avoidance, and most deadlock-free routing algorithms are deduced by the strategy. The designers should check whether this algorithm is deadlock free or not. If not, the designers have to add a hardware resource or to restrict routing rules for deadlock free. Deadlock free is analyzed by building a dependency graph. And thus the dependency graph can't be cyclic. There are lots of algorithms to solve deadlock by prohibiting some specific turns to prevent the deadlock, such as West-First Routing Algorithm, North-Last Routing Algorithm and Negative-First Routing Algorithm [5.5]. The definition of livelock is a packet enters a cyclic path dose not reach its destination. The livelock will be occurred in dynamic routing algorithms.

The routing algorithms can be classified to several categories [5.6]. The routing decision at each router can be static or dynamic. In the static routing, the path is completely determined the source and destination address. The routing does not

consider the current load and the traffic in the network. XY- routing algorithm is a typical static routing algorithm. On the contrary, the dynamic routing scheme decides the path by not only source and destination address but also the dynamic network condition. The advantage of static routing scheme is its simplicity of design and hardware overhead, but it may perform poorly if traffic pattern is not uniform. The dynamic routing can use alternative paths which consider the network traffic. West-First Routing Algorithm, North-Last Routing Algorithm, Negative-First Routing Algorithm [5.7], Dynamic Routing Algorithm for Avoiding Hot Spots [5.8], Neighbors-on-Path [5.9], Antnet [5.10], and DyAD [5.11] and DyXY[5.12] are the approaches of the dynamic routing algorithm. Other approaches were proposed for specific applications [5.13] or for reliable issues [5.14]-[5.16].

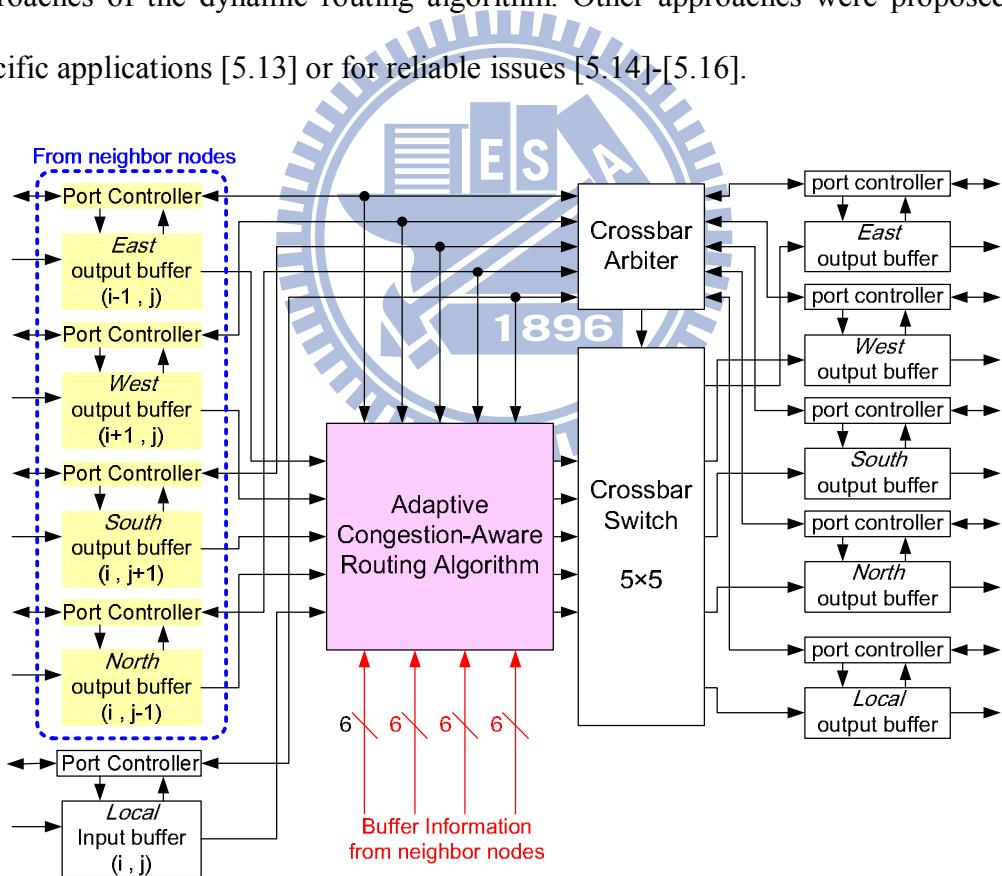


Fig. 5.1 The architecture of the congestion-aware router.

## 5.2 Congestion-Aware Routing Concept

An adaptive congestion-aware routing algorithm is proposed for the mesh topology

to avoid hotspots and to shorten the average latency of transmitted packets through adopting minimal and non-minimal path alternately without virtual channels. When a hotspot is close to a router, the routing algorithm will choose the non-minimal paths. Otherwise, the minimal path will be selected. In addition, a guaranteeing QoS arbitration mechanism is also implemented in the router depending on the priority of the packets. Fig. 5.1 illustrates the architecture of the router design. We implemented an NxN network of interconnected tiles with the mesh topology. Each tile is composed of a router and a processing element. Therefore, a router is connected with four adjacent routers and its processing element.

The flow diagram of the routing procedure in a router is shown as Fig. 5.2. And the detail of each step will be described as follow.

- Step 1. To begin with the packet injection, the routing controller will calculate the scores of each direction by summing the number of available buffer unit of adjacent router and the buffer of its neighbors.
- Step 2. Determine the next nodes whether are on minimal paths or not.
- Step 3. If the scores of next nodes which are on minimal paths are higher than a specific value we can adjust, select one of the next nodes on minimal paths with the highest score. Otherwise, select the score is the highest one from the next nodes which are on the minimal and the non-minimal paths.
- Step 4. If there are more than one packet will be routed to the same output channel, the arbiter will arbitrate the packets according to the priorities which will guarantee the quality-of-service. However, in order to avoid deadlock caused from guaranteeing QoS, once any packet is halted for more than 10 cycles, the arbitration policy is switched to choose the packet with the longest waiting time.

Step 5. Switch the packets to the corresponding output channels.

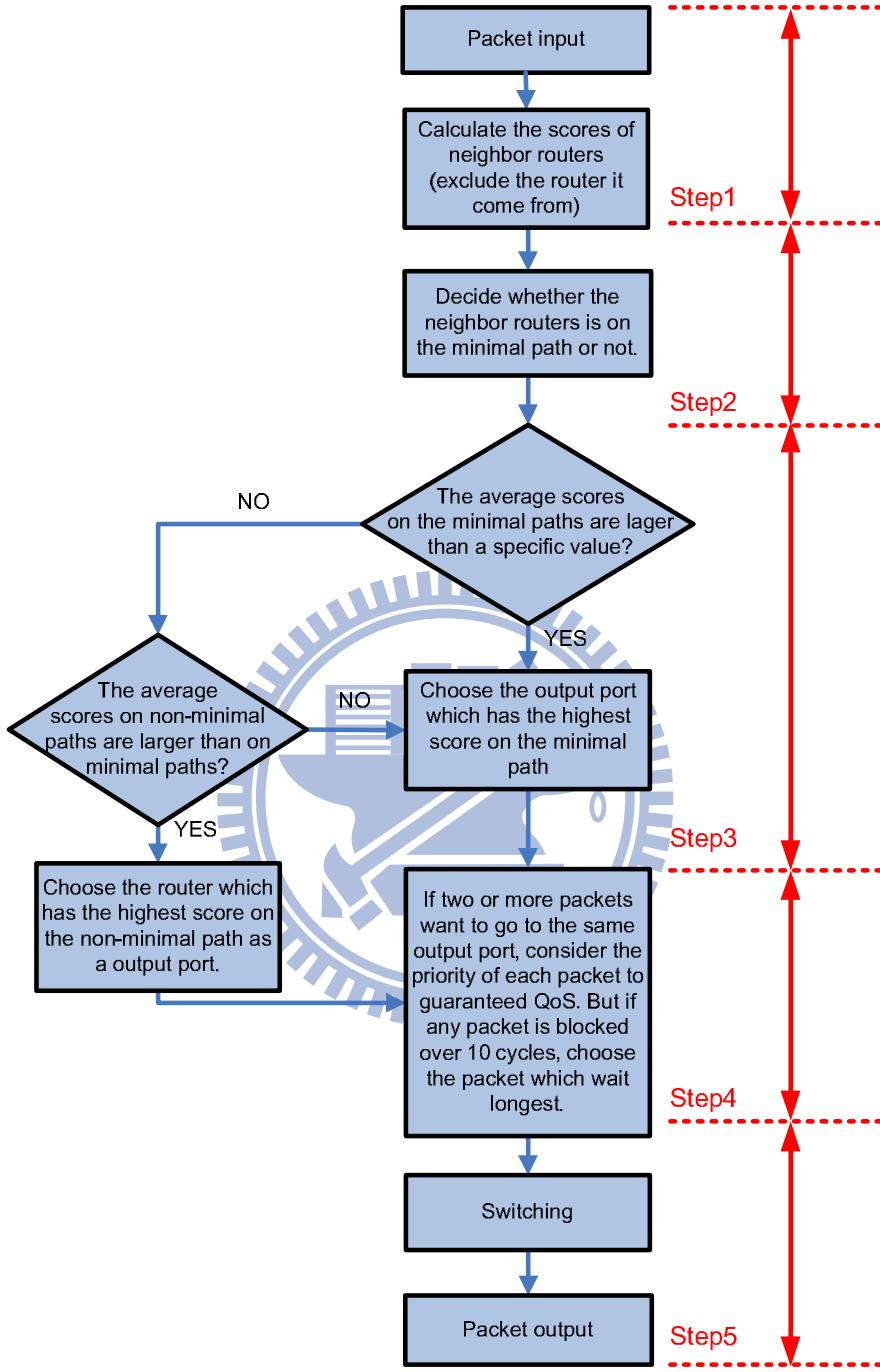


Fig. 5.2 The flow diagram of the routing procedure.

The five steps construct the packet flow in a router. Step1 to step3 are realized by the congestion-aware adaptive routing algorithm. And thus step4 and step5 are accomplished by the crossbar arbiter and crossbar switch, respectively.

## 5.3 Congestion-Aware Routing Algorithm

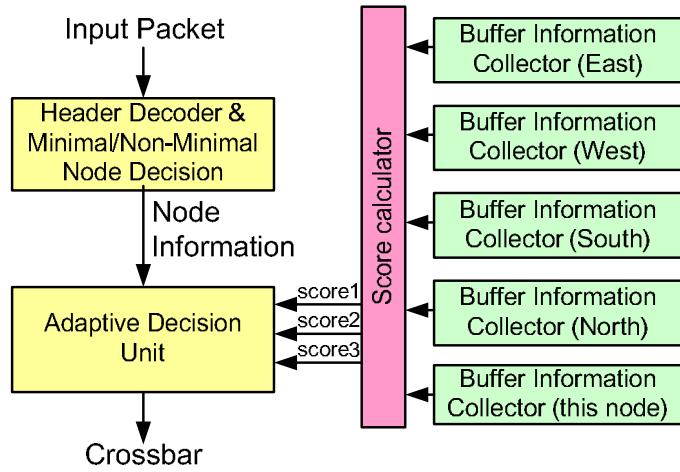


Fig. 5.3 The architecture of adaptive congestion-aware routing algorithm.

In this section, the proposed adaptive congestion-aware routing algorithm will be described. It is proposed for the mesh topology to avoid hotspots and to shorten the average latency of transmitted packets through adopting minimal and non-minimal path alternately. Fig. 5.3 shows the architecture of the routing algorithm. Once a packet injects a node, the adaptive congestion-aware routing algorithm will decide the output of the injected packet depending on the adaptive decision. The adaptive decision of the output channels depends on the scores of the probable neighboring nodes which indicate the traffic around this router. The detail of each block will be described as follow.

### 5.3.1 Deadlock Avoidance by the Odd-Even Turn Model

The proposed congestion-aware adaptive routing algorithm adopts Odd-Even turn model to solve the deadlock problem [5.17]. The Odd-Even (OE) turn model prohibit some turns for eliminating deadlock. The rules of OE turns are as following.

Rule 1. Any packet is not allowed to take an EN turn at any nodes located in an even column, and it is not allowed to take an NW turn at any nodes located in an odd column.

Rule 2. Any packet is not allowed to take an ES turn at any nodes located in an even column, and it is not allowed to take an SW turn at any nodes located in an odd column.

To compute the adaptiveness of OE Turn model, one metric for measuring the adaptiveness of a partially adaptive routing algorithm is the degree of adaptiveness, which is essentially the number of shortest paths the algorithm allows from the source to the destination. From the analysis of the metric in [5.17], [5.18], it is apparent that adaptiveness of OE turn model is larger than others.

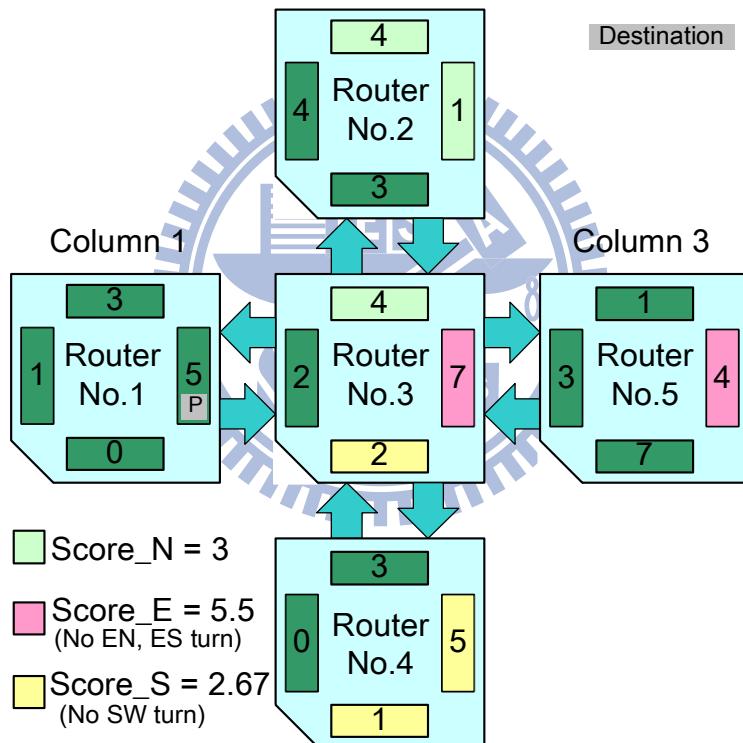


Fig. 5.4 An example of the score calculation

### 5.3.2 Score Calculator

The score of the next node is computed by referring the available buffer space of the next node and the neighboring nodes of the next node with Odd-Even turn model. Fig. 5.4 is an example of the score calculation. The light blue square represents the router, and the green square represents the buffer. 'P' represents the position of the

packet. In addition, the number in the green square represents the number of available buffer unit. Now, the packet is in the East output buffer of No.1 router, and it indicates the packet will go east in this cycle. Since the packet does not arrive at the destination and cannot go back to west, its next node will only be North, East and South of No.3 router.

In order to calculate the north score, the score calculator should get the buffer information of the north node from the buffer information collector with the Odd-Even turn model. The north score includes the number of available buffer unit of the north output buffer and the next probable output buffers, namely the average of the number of the light green parts. Beware that the included output buffer should be based on Odd-Even turn model. Because the NW turn is prohibited, the west output buffer of No.2 router is not included in the north score. The calculation of the east score is the same as that of the north score. Since the east score should be based on Odd-Even turn model (the EN, ES turn are prohibited.), the score is the average of the numbers which are shown as the light green parts. And the South score is calculated in the same manner as shown in Fig. 5.4.

Table 5.1 The modified score calculator.

The valid buffer of the next node (Depending on the Odd-Even Turn Model)	Score Calculation
0	out
1	$(out + next\_1) / 2$
2	$(2 \times out + next\_1 + next\_2) / 4$
3	$(out + next\_1 + next\_2 + next\_3) / 4$
The output buffer of the routed node is denoted as “out”. And the output buffers of the next node are denoted as “next_1”, “next_2” and “next_3”.	

For the hardware implementation, the function of the average will be implemented as a divider, as the average number is from 2 to 4 by the Odd-Even turn model. The

cost of the divider, however, will occupy large area and dissipate a lot of power consumption. Therefore, in order to implement the score calculator efficiently, the function of score calculator will be modified depending on the average number as shown as Table 5.1. If the average number is 3, the score of the output buffer in the routed node will be weight as twice and the average number will be increased to 4. Therefore, the average number will be modified as (1, 2, 4), and is easy to implemented as shifter.

### 5.3.3 Adaptive Decision Unit

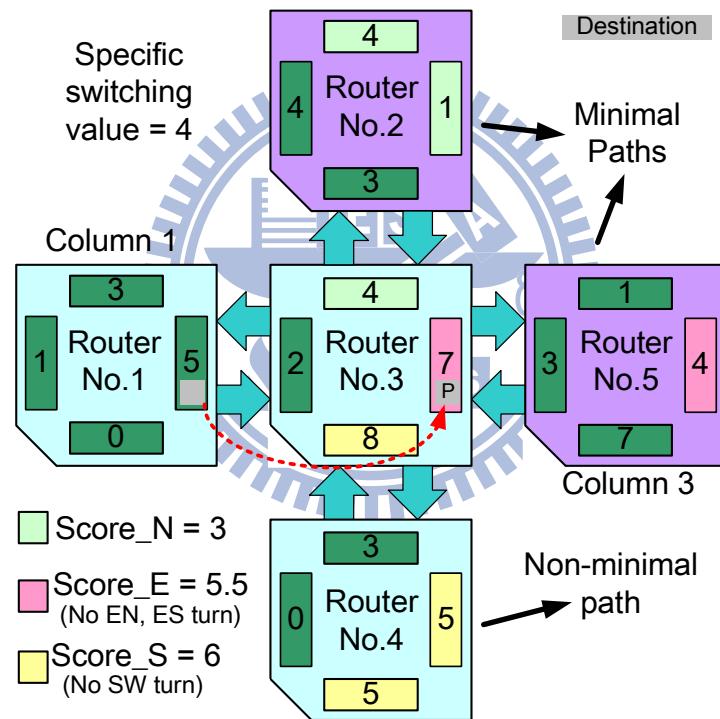


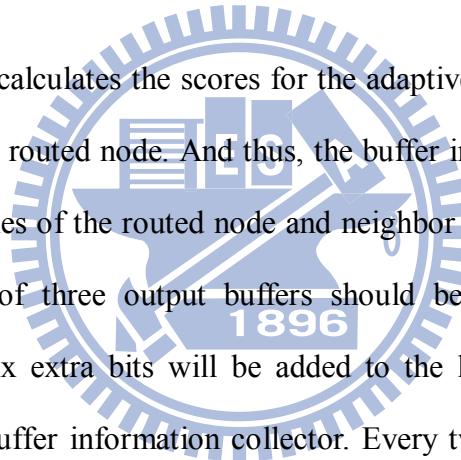
Fig. 5.5 An example of the adaptive decision.

The adaptive decision unit decides the output channel of the routed packet by the scores from the score calculator and the information of minimal/non-minimal paths. If the scores of next nodes which are on minimal paths are higher than a specific switching value, the adaptive decision unit will select the highest score on the minimal paths. Otherwise, the adaptive decision unit will select the highest scores from the next nodes which are on the minimal and the non-minimal paths. If the score

of the minimal path is higher than the specific switching value, it indicates that the next node is not a hot spot and far from hotspots also. Therefore, the non-minimal path routing is unnecessary to be considered, even if the score of the non-minimal path is much higher than that of the minimal path.

Fig. 5.5 shows an example while the score of the nominal path is higher than the specific switching value. (Assume the specific switching value equals to 4.) Although the score of the non-minimal path is higher than the minimal path, the adaptive decision unit still selects the minimal path because there are no hot spots on this path.

#### 5.3.4 Buffer Information Collector



The score calculator calculates the scores for the adaptive decision unit depending on the traffic around the routed node. And thus, the buffer information collectors will indicate the buffer utilities of the routed node and neighbor nodes. For each neighbor node, the information of three output buffers should be collected for the score calculator. Therefore, six extra bits will be added to the link wires to transfer the buffer utilities for the buffer information collector. Every two bit indicates the three states of each output buffer, which are “increase”, “decrease” and “silence”. The buffer information collector will record the buffer utilities by the three states.

### 5.4 QoS Guarantee Arbitration Mechanism

The arbitration mechanism is applied after the routing algorithm. If there are two or more packets routed to the same next node, the arbiter must decide which packet can pass. A Quality-of-Service (QoS) guarantee arbiter is proposed as a dynamic arbitration mechanism. In the proposed arbitration architecture, it considers not only the number of waited cycles of packets, but also the QoS of the system.

In the fields of packet-switched networks, the Quality-of-Service refers to resource reservation control mechanisms. Quality-of-Service can provide different priority to different users or data flows, or guarantee a certain standard of performance to a data flow according to requests from the application program or the internet service. QoS guarantees are important if the network bandwidth is limited. A best-effort network or service does not support Quality of Service. QoS is also a set of capabilities that allow you to create differentiated services for network traffic, thereby providing better service for selected network traffic. For example, with QoS, you can increase bandwidth for some critical traffic, limit bandwidth for the other non-critical traffic, and provide consistent network response, among other things. This allows you to use network connections more efficiently, and to establish service level agreements with customers of the network.

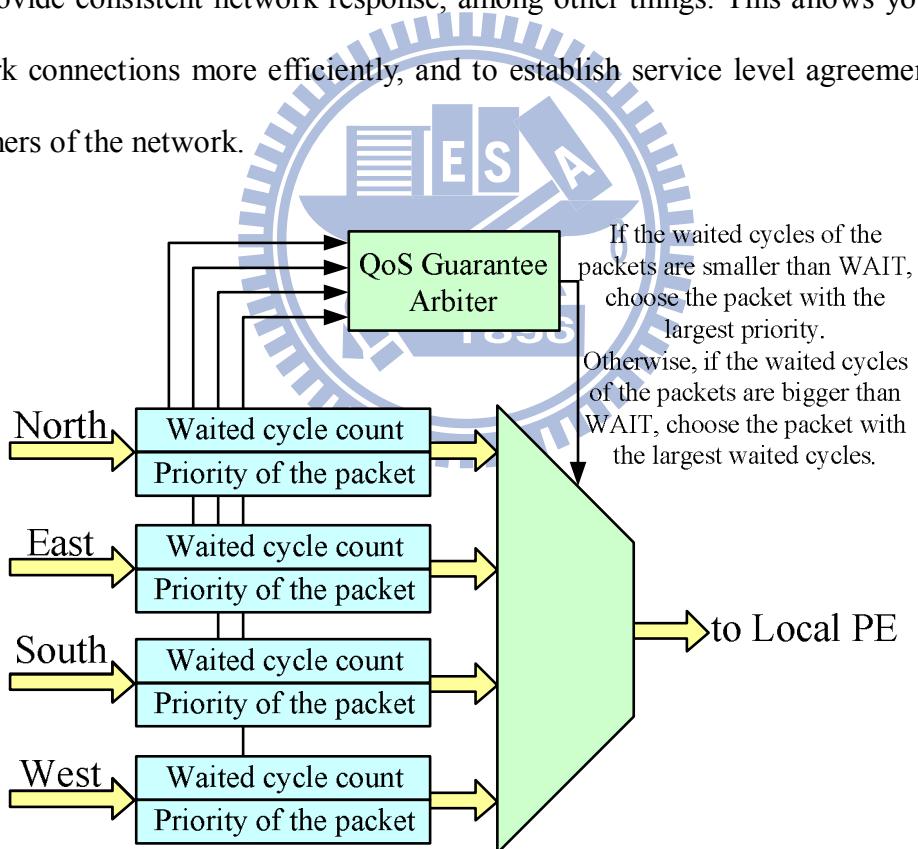


Fig. 5.6 QoS guarantee arbitrator.

According to the reason of above, we decided to use QoS into the arbitration mechanism. The QoS guarantee arbiter is shown as Fig. 8. The packet header contains the information of the QoS by setting the two priority bits. Therefore, each input

packet has the four levels of priority. This arbitration chooses the highest priority one if the waited cycle of each input packets does not exceed the value of WAIT (this variable is set the same value for every algorithms). If one of input packets exceed, the controller will choose the maximum waited cycle of input packets to prevent the deadlock problems.

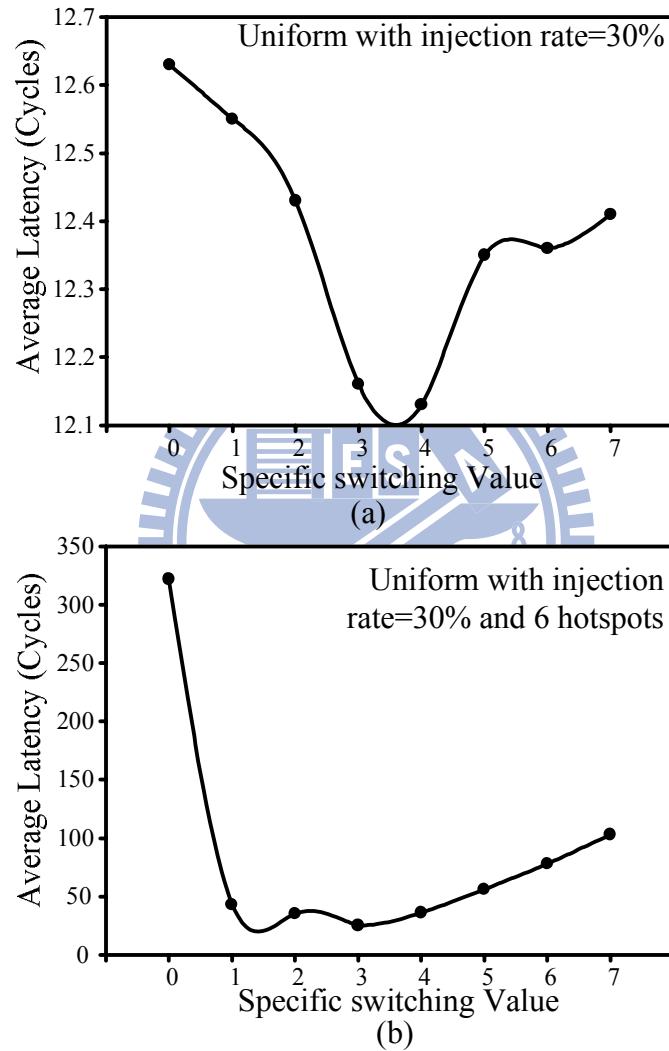


Fig. 5.7 The average latencies versus the specific switching values under the uniform patterns (a) without hotspots (b) with 6 hotspots.

## 5.5 QoS Guarantee Arbitration Mechanism

Different traffic patterns are applied for the simulation of the routing algorithms, such as uniform, uniform with hotspots, transpose and transpose with hotspots. In

each cycle, the packets are sent into the local buffer occasionally, it is decided by injection rate. The 8x8 mesh topology is implemented for the simulation in C++ program, and each channel in the router has 8 available buffer units.

Fig. 5.7(a) shows the average latencies of different “specific switching values” with fixed injection rate in the uniform traffic patterns. The specific switching value is the switching point of considering the non-minimal paths. Fig. 5.7(b) shows the average latencies with the uniform traffic patterns and 6 hotspots. The 6 hotspots are set as Fig. 10, and the 6 hotspots are placed on (2,3) , (2,4) , (3,3) , (3,4) , (7,3) , and (7,4). The 6 hotspots have 50% ~80% injection rates.

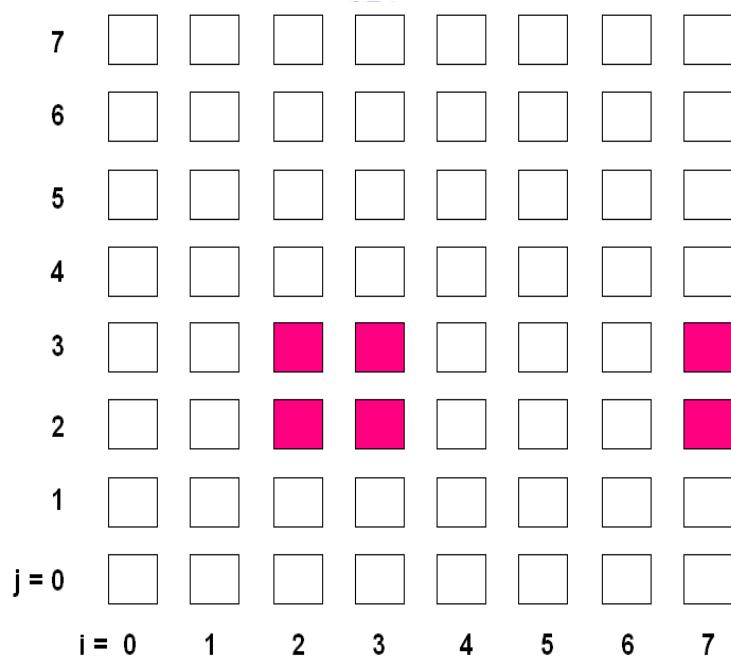


Fig. 5.8 Hotspots setting for the 8x8 mesh network.

The average latencies versus the specific switching value with the transpose patterns and the hotspots are shown as in Fig. 5.9(a) and Fig. 5.9(b), respectively. The transpose traffic pattern transfer a packet from  $(i,j)$  to  $(7-i, 7-j)$ . With the increasing of the specific switching value, it indicates that the probability of selecting non-minimal paths is increasing. If the network has hotspots, the benefit of the non-minimal path selection will be revealed. For the low injection rate of uniform patterns without any

hotspots, it is obviously that the differences between the different specific switching values are not significant. According the analysis of the specific values as shown in Fig. 5.7 and Fig. 5.9, the performance will be better when the “specific switching value” is from 3 to 5.

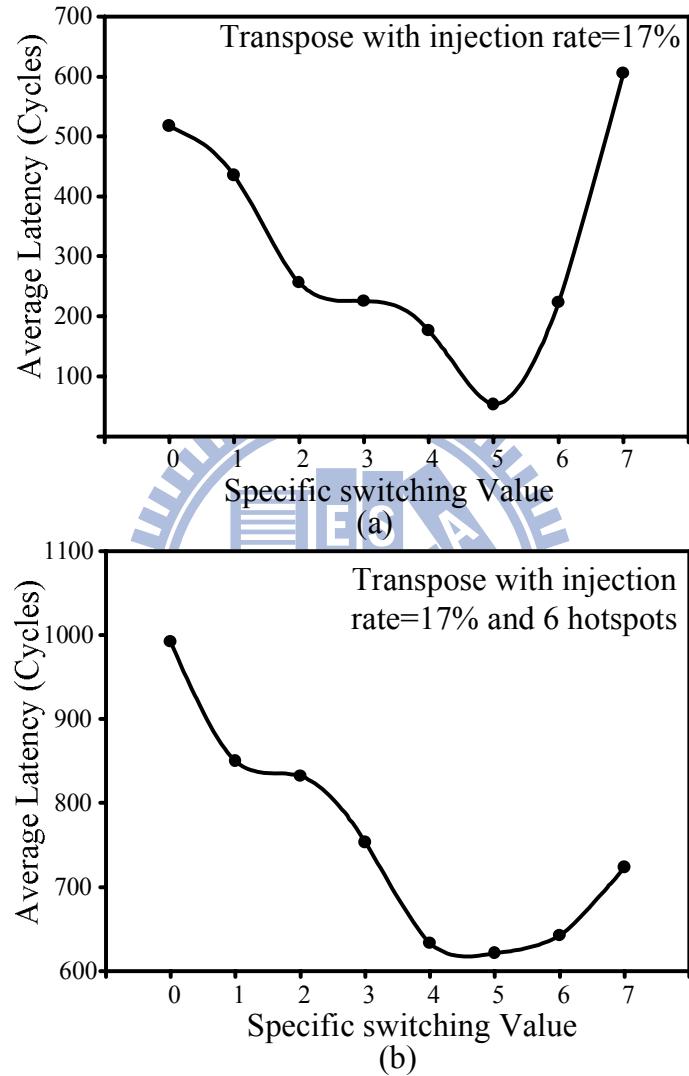


Fig. 5.9 The average latencies versus the specific switching values under the transpose patterns (a) without hotspots (b) with 6 hotspots.

Before analysis the performance of the proposed routing algorithm, some compared routing algorithms will be introduced as follow first.

- The *XY-routing algorithm* is a minimal algorithm. In XY-routing, every node chooses the next node concern the direction X of destination first. The packet

would go along the x direction of destination, and then follow the Y direction until it arrives at the destination. The XY-routing algorithm could be realized simply. For some specific traffic pattern, the XY-routing algorithm may perform very well.

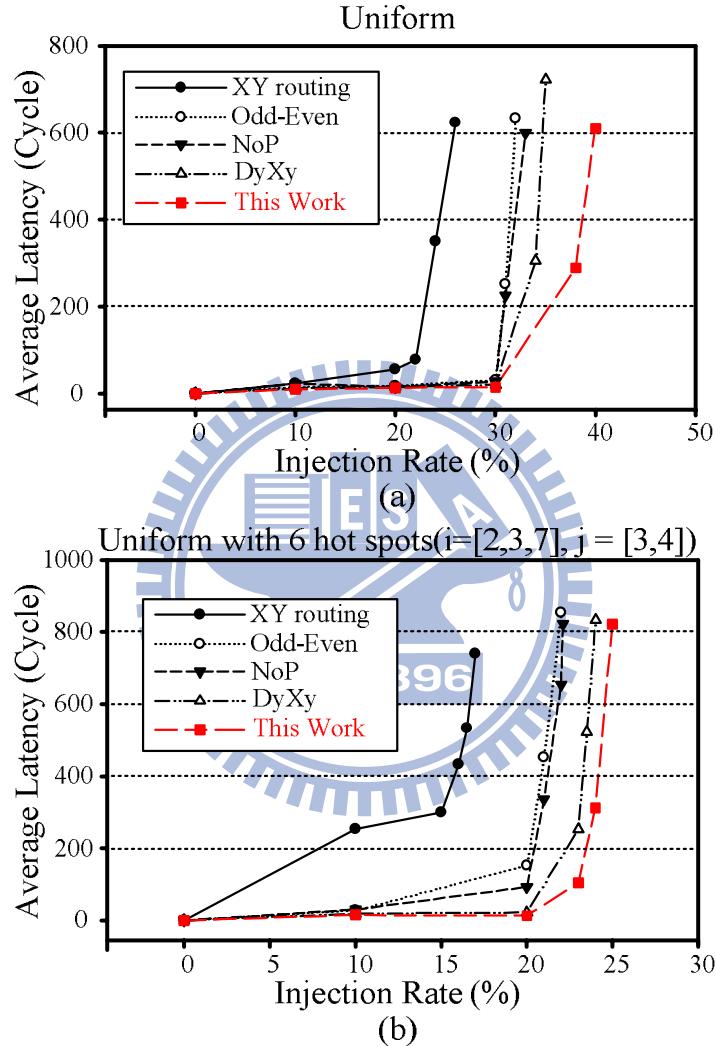


Fig. 5.10 The comparisons under the uniform patterns (a) without hotspots (b) with 6 hot spots.

- The OE algorithm is used to prevent the deadlock. The minimal version with odd-even turn model is used to realize.
- The *NoP algorithm* is a new method in the field of dynamic routing algorithm. NoP concerns the available of neighbor buffers and the neighborhood of neighbor buffers to decide the next node.

- DyXY, dynamic XY routing algorithm, is proposed to provide adaptive routing by the stress value which is a parameter representing the congestion condition. Each packet only travels along the minimal paths.

The comparisons between the adaptive congestion-aware routing algorithm and other approaches are shown in Fig. 5.10(a) and Fig. 5.10(b) under the uniform traffic patterns without/with 6 hotspots, respectively. The 6 hotspots have 50% ~80% injection rates. No whether the network has hotspots or not, the proposed routing algorithm can achieve average latency reduction. The transpose traffic patterns are utilized to compare the latency of the given algorithms. In addition, the comparisons between the routing algorithms are also simulated under the transpose traffic patterns as shown in Fig. 5.11.

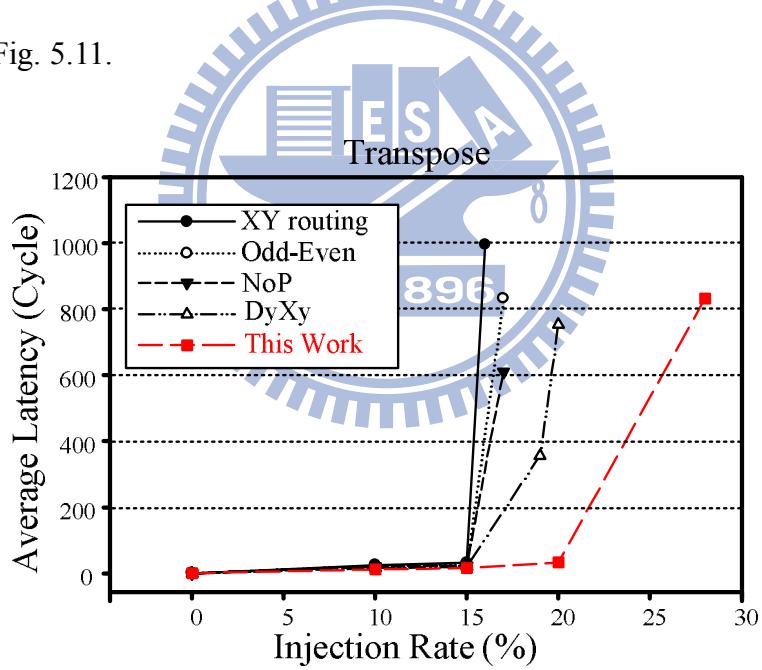


Fig. 5.11 The comparisons under the transpose patterns.

The OE algorithm and NoP algorithm are not outstanding no whether in the uniform traffic patterns or the transpose traffic patterns. The DyXY performs better in the traffic pattern with hotspots, but not as well as the proposed algorithm, an adaptive congestion-aware routing algorithm. The proposed algorithm is effective in the pattern with hotspots at the proper injection rates. It is very practical in the real application

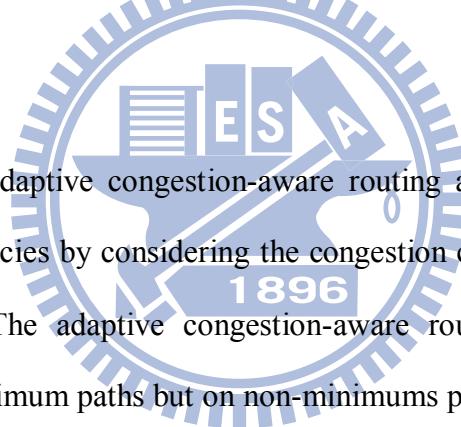
when there are some nodes which have high injection rate (like CPU, DSP...etc.).

The proposed adaptive congestion-aware routing algorithm is also implemented based on UMC 65nm CMOS technology. The implementation includes the header decoder, the adaptive congestion-aware routing algorithm, the MUX-based crossbar, the QoS guarantee arbiter and the circular output buffers. Table 5.2 shows the area overhead of the proposed routing algorithm Compared to the XY-routing algorithm.

Table 5.2 Area overhead of the proposed routing

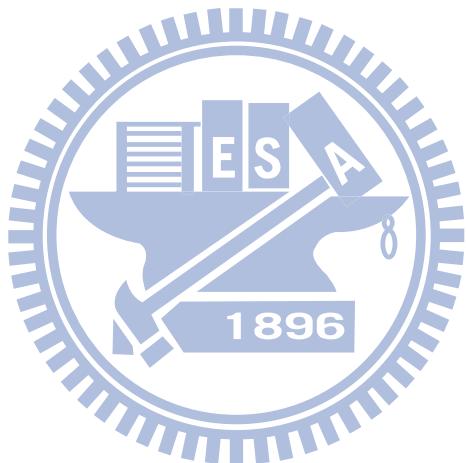
	Area overhead (%)
XY-routing algorithm	0 %
Adaptive congestion-aware routing algorithm	22.47 %

## 5.6 Summary



In this chapter, an adaptive congestion-aware routing algorithm is presented to reduce the average latencies by considering the congestion conditions and distributed hotspots of network. The adaptive congestion-aware routing technique switches packets not only on minimum paths but on non-minimum paths by the buffer utilities of the network. It adopts the minimal paths and non-minimal paths depending on a specific switching value, which will achieve the optimal performance. Both minimum and non-minimum paths are based on the odd-even turn model for avoiding deadlock and livelock problem. The decision of minimum path and non-minimum path depends on the utilities of buffers in neighbor nodes and the specific switching value. According to the simulation results, the proposed adaptive routing architecture is smarter than other architectures no whether which traffic pattern is. The proposed routing technique can achieve 54.1% latency reduction compared to DyXy routing with 23% injection rate and 6 hotspots. It has been observed that the proposed

adaptive routing architecture is superior to other routing algorithm no whether which traffic pattern is. If there are hotspots distributed in the mesh, the proposed architecture would become more useful. Therefore, the proposed algorithm is effective in the pattern with hotspots at the proper injection rates. It is very practical in the real applications of mesh OCINs.



# ***Chapter 6: Energy-efficient Routing Tables for OCINs and IPv6 Applications***

Content addressable memory (CAM) is widely utilized to execute lookup-table functions for routing packets in on-chip interconnection networks (OCINs). Furthermore, ternary content addressable memory (TCAM) is extensively adopted in network systems. As routing tables become larger, energy consumption and leakage current become increasingly important issues in the design of TCAM in nano-scale technologies. This chapter presents a novel 65nm energy-efficient TCAM macro design for IPv6 applications. The proposed TCAM employs the concept of architecture and circuit co-design. To achieve an energy-efficient TCAM architecture, a butterfly match-line scheme and a hierarchy search-line scheme are developed to reduce significantly both the search time and power consumption. The match-lines are also implemented using noise-tolerant XOR-based conditional keepers to reduce not only the search time but also the power consumption. To reduce the increasing leakage power in advanced technologies, the proposed TCAM design utilizes two power gating techniques, namely super cut-off power gating and multi-mode data-retention power gating. An energy-efficient 256x144 TCAM macro is implemented using UMC 65nm CMOS technology, and the experimental results demonstrate a leakage power reduction of 19.3% and an energy metric of the TCAM macro of 0.165 fJ/bit/search.

## **6.1 Background**

CAM, also called associative memory, executes the lookup-table function in a

single clock cycle using dedicated comparison circuitry. CAM compares input search data against a table of stored information, and returns the matching data. Accordingly, CAM cells contain storage memories and comparison circuits. CAM cells are of two types - binary content addressable memory (BCAM) and ternary content addressable memory (TCAM) - depending on their comparison function as presented in Fig. 6.1. A BCAM cell has two states – the “one” state and the “zero” state. A BCAM cell contains 1-bit storage memory and a 1-bit comparison circuit. TCAM has three states - logic 0, logic 1, and don’t-care X. The third state, don’t-care X, which is used in masking, makes TCAM suitable for network router applications. Hence, the difference between BCAM and TCAM is that TCAM contains an extra SRAM to store the don’t-care state. If the datum in don’t-care cell is 1, then the match-line (ML) will bypass the don’t-care cells and be discharged to ground. It will not perform any comparison operation. If the datum in don’t-care cell is 0, then the function of TCAM is the same as that of BCAM.

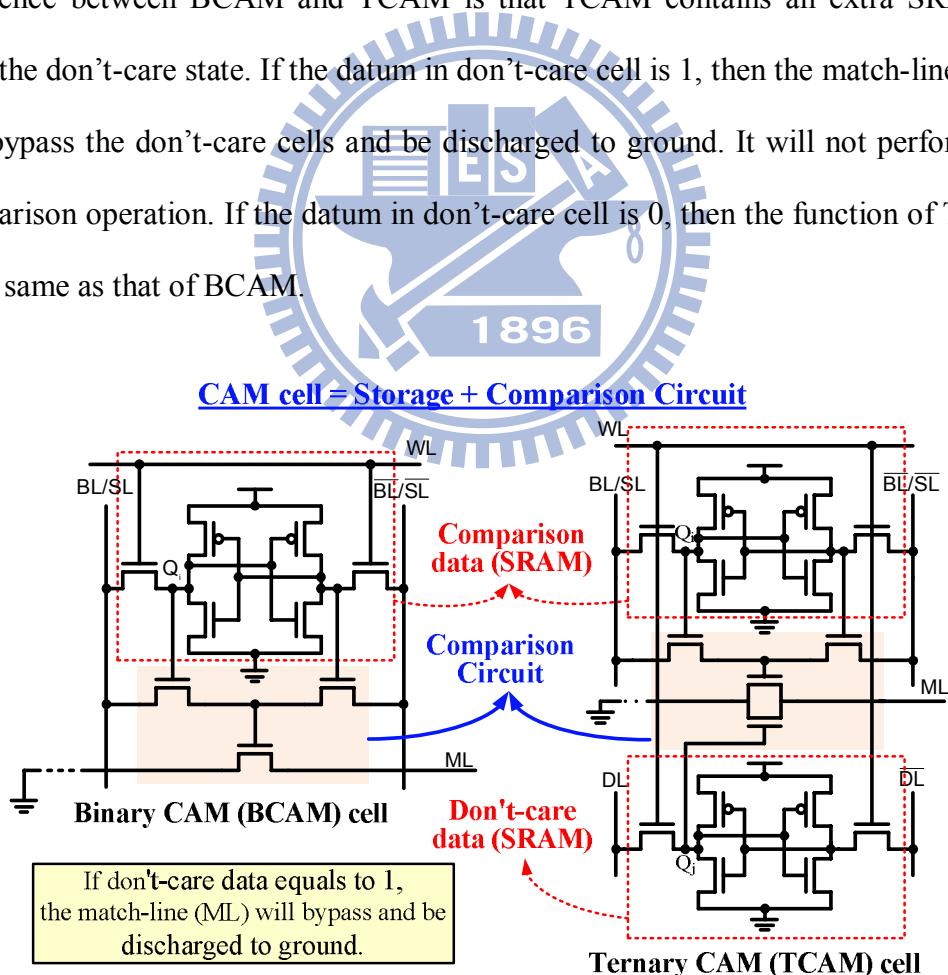


Fig. 6.1 Binary CAM (BCAM) cell and ternary CAM (TCAM) cell.

In past decades, CAM has been employed in numerous applications that depend on

fast search. These applications are parametric curve extraction [6.1], Hough transformation [6.2], Lempel-Ziv compression [6.3], image coding [6.4], the human body communication controller [6.5], the periodic event generator [6.6], and the virus-detection processor [6.7]. At present, CAM is popular for use in network routers for packet forwarding, packet classification, asynchronous transfer mode (ATM) switching, and other functions. As the range of CAM applications grows, power consumption becomes one of the critical challenges. The trade-off among power, speed, and area is the most important issue in recent researches on large-capacity CAMs. The primary commercial application of CAMs today is the classification and forwarding of Internet protocol (IP) packets in network routers. Internet Protocol Version 6 (IPv6) 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 or 144-bit identifiers for interfaces and sets of interfaces [6.8]-[6.12]. Each address must be stored in 128 or 144 TCAM cells, resulting in a long search delay path in network routers in packet forwarding applications. Accordingly, high speed and low power are the two major goals of TCAM design for IP-address forwarding applications, especially in nano-scale technologies.

## 6.2 Routing Tables in OCINs

The lookup-table functions are also adopted for routing packets in OCINs via CAM and TCAM [6.13]-[6.16] that is called table-based routing. Many routers use routing tables either at the source or at each hop along the route to implement the routing algorithms, including deterministic, oblivious and adaptive routing. With a single entry per destination, a table is restricted to deterministic routing, but oblivious and adaptive routing can be implemented by providing multiple table entries for each destination. The value of the relation for each pair of inputs is stored in the table and

the table is indexed by the inputs. The major advantage of table-based routing is generality. Subject to capacity constraints, a routing table can support any routing relation on any topology. A routing chip that uses table-based routing can be used in different topologies by simply reprogramming the contents of the table.

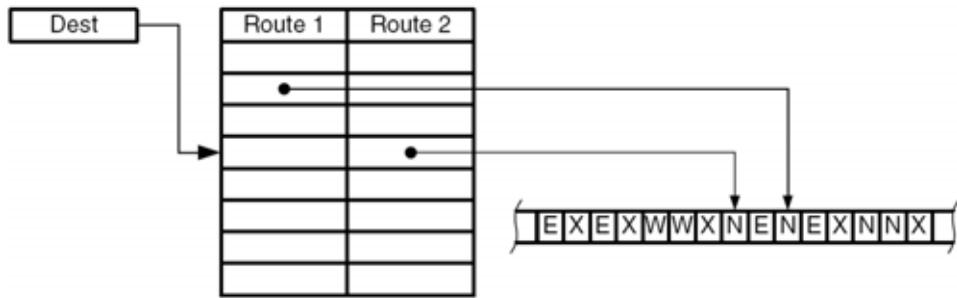


Fig. 6.2 A routing table organized with source routes.

With source routing, all routing decisions for a packet are made entirely in the source terminal by table lookup of a pre-computed route as shown in Fig. 6.2 [6.15]. Each source node contains a table of routes, at least one per destination. To route a packet, the table is indexed using the packet destination to look up the appropriate route or set of routes. Because of its speed, simplicity, and scalability, source routing is one of the most widely used routing methods for deterministic and oblivious routing. Table-based routing can also be performed by storing the routing table in the routing nodes rather than in the terminals. Node-table routing is more appropriate for adaptive routing algorithms because it enables the use of per-hop network state information to select among several possible next-hops at each stage of the route. With the arrangement, when a packet arrives at a router, a table lookup must be performed before the output port for the packet can be selected. Node-tables may be organized in a hierarchical manner to further reduce storage requirements. In such a table, each nearby node gets its own next hop entry, small groups of remote nodes share entries, larger groups of distant nodes share entries. Furthermore, a routing table minimization in OCINs was proposed to reduce the overhead of routing tables [6.17].

### 6.3 Architecture of TCAM Macro in Network Routers

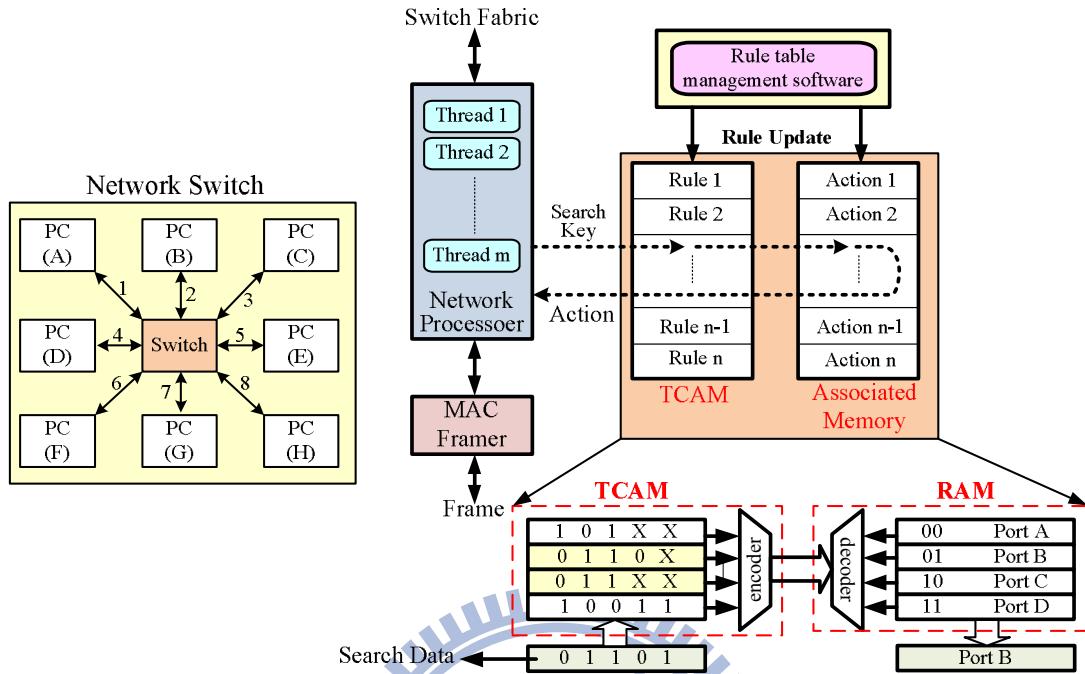


Fig. 6.3 A simplified block diagram of a TCAM macro and packet forwarding by an address-lookup table in network routers.

In recent years, TCAMs have been popularly used in network routers for packet forwarding and packet classification. Network routers forward data packets from an incoming port to an outgoing port, using an address-lookup function [6.18]. Fig. 6.3 schematically depicts a simplified block diagram of a TCAM macro. The search data are broadcast onto the search-lines to the table of stored data. The address-lookup function determines the destination address of the packet and selects the output port that is associated with that address. For example, the packet destination address 01101 is input to the TCAM. As indicated by the table, two entries are matched, and the priority encoder chooses the upper entry and generates the matching location 01. This matching location is the address that is input to a RAM that contains a list of output ports, as shown in Fig. 6.3. A read operation of RAM outputs the port designation, port B, to which the incoming packet is forwarded. This TCAM/RAM system fully implements an address-lookup engine for packet forwarding.

The number of bits in a TCAM word is generally large, and existing implementations range from 36 to 144 bits. A typical TCAM utilizes a table size that has between a few hundred entries and 32K entries, corresponding to an address space that ranges from 7 bits to 15 bits. Each stored word has a match-line that indicates whether the search word is identical to the stored word (matching) or not (mismatching, or “a miss”). The match-lines are fed to an encoder that generates a binary matching location that corresponds to the most-direct routing. An encoder is adopted in systems in which only a single match is expected. In TCAM applications, where more than one word may match, a priority encoder is employed instead of a simple encoder. A priority encoder identifies the location that is matched with the highest priority to map the result of matching, such that words in lower address locations have higher priority. The overall function of TCAM is to take a search word and return the matching memory location.

As nano-scale technologies become more advanced, leakage currents increasingly dominate their overall power consumption. However, previous investigations of low-power TCAM have focused only on dynamic power consumption [6.19]-[6.28]. Two match-line methods are applied with the TCAM architecture. When the stored data are not identical to the search data in every bit, the match-line is discharged to ground. This match-line is called the NOR-type match-line [6.25], [6.26], [6.28]. The other is the NAND-type (AND-type) match-line. The NAND-type match-line is discharged to ground only when all bits of stored data match all bits of search data. Generally, the NOR-type match-line has a shorter search time than the NAND-type match-line but consumes much more power in the search operation. The NAND-type consumes less search power but has a longer search time owing to the deep fan-in circuits. Numerous approaches have been presented to improve the search time of the

NAND-type match-line. They include the pseudo-footless clock-data pre-charge dynamic (PF-CDPD) match-line scheme [6.21] as shown in Fig. 6.4, the range matching scheme [6.22] and the tree-style AND-type match-line scheme [6.23], [6.27]. The evaluation operation of the conventional PF-CDPD match-line is enabled or disabled depending on the output of the preceding stages. The details of PF-CDPD circuitry are described as follows.

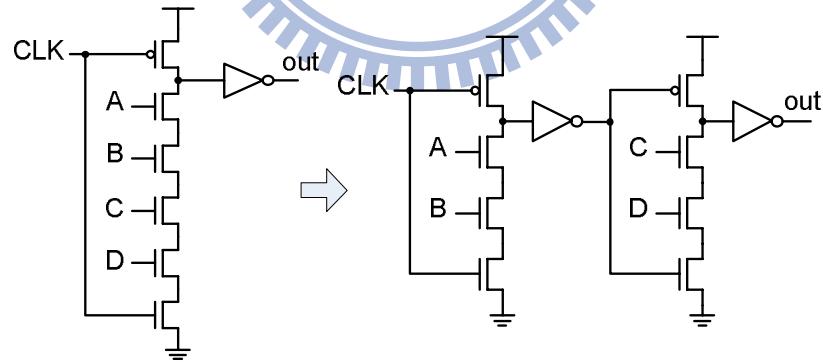
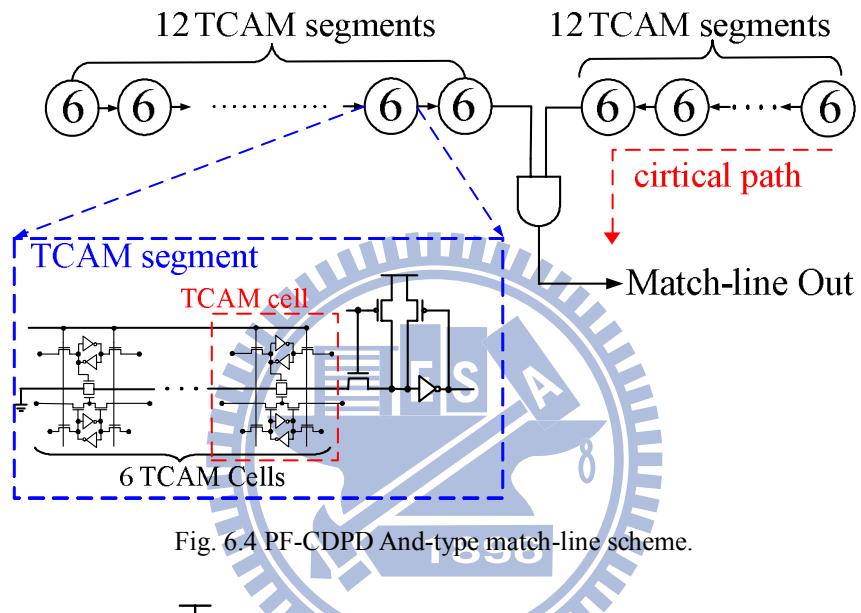
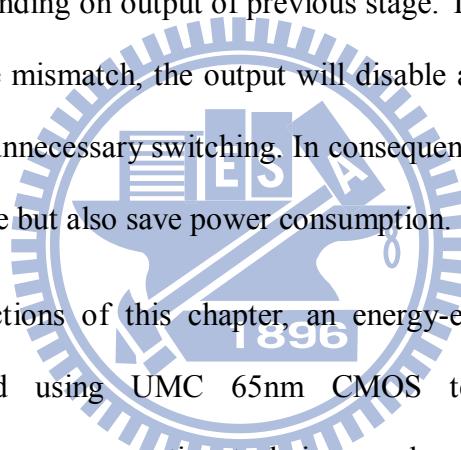


Fig. 6.5 Transfer dynamic logic into clock-and-data pre-charge dynamic (CDPD) circuits.

For the PF-CDPD circuitry as shown in Fig. 6.5, 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 at the beginning of pre-charge phase. 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. 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.



In the following sections of this chapter, an energy-efficient 256x144 TCAM macro is implemented using UMC 65nm CMOS technology. Additionally, multi-mode data-retention power gating technique and super cut-off power gating technique are proposed to reduce leakage currents. Moreover, the TCAM design employs the concept of architecture and circuit co-design using noise-tolerant XOR-based conditional keepers, butterfly match-line scheme and hierarchy search-line scheme to reduce not only critical paths but also power consumption.

## 6.4 Energy-Efficient Match-Line

As technology advances, leakage currents, coupling noise, charge sharing and power/ground fluctuation noises all increase the soft-error rate of dynamic circuits in the match-lines of the TCAM macro. The increasing noise not only worsens

performance; it even destroys the functionality of the TCAM macro. Therefore, the proposed butterfly match-line scheme with the XOR-based conditional keeper supports a low-power, high-speed and noise-tolerant TCAM. The butterfly match-line scheme improves performance by increasing the parallelism of the search operation. It also reduces the power consumption in a manner that depends on the interlaced pipeline since the butterfly connections turn off more TCAM segments than PF-CDPD match-line scheme does. The XOR-based conditional keeper for the match-lines provides noise-tolerant circuitry to reduce both search time and power consumption. The XOR-based conditional keeper eliminates the performance overhead of the butterfly connections. The butterfly match-line scheme and the XOR-based conditional keeper are described in detail below.

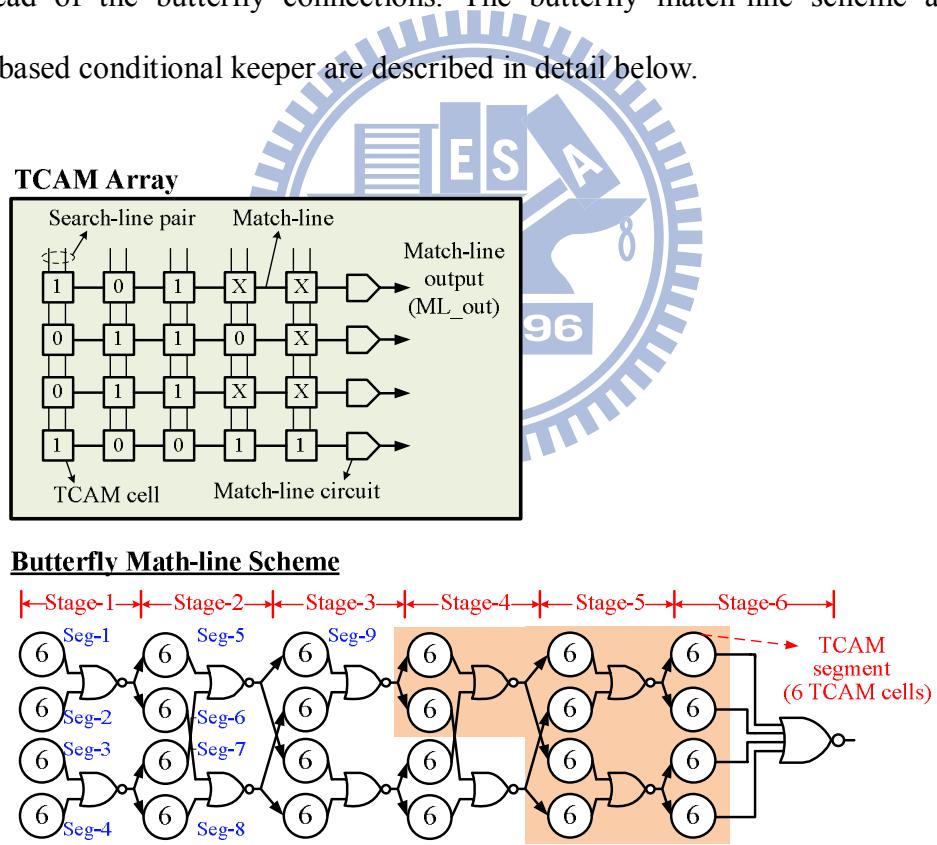


Fig. 6.6 Butterfly match-line scheme.

#### 6.4.1 Butterfly Match-line Scheme

Fig. 6.6 presents the butterfly match-line scheme, which is based on the PF-CDPD

match-line scheme [6.21], for 144 TCAM cells. Each circle represents a TCAM segment, which contains six TCAM cells and a dynamic circuit. The degree of parallelism is double that of the conventional PF-CDPD match-line scheme. In 144-bit TCAM cells, the match-line is folded into four sub match-lines in six stages. Accordingly, the number of segments of the critical path is reduced from 12 to six. Therefore, the critical delay of a match-line is reduced from  $(12T_{seg} + T_{AND2})$  to  $(6T_{seg} + 5T_{NOR2} + T_{NOR4})$  compared to the conventional PF-CDPD match-line scheme [6.21]. The  $T_{seg}$  represents the discharging time of a TCAM segment and is much larger than the delay of NOR gates. To reduce the power consumption, a butterfly connection is made among these four independent sub match-lines by intersecting the interlaced connections, as shown in Fig. 6.6. Therefore, the mismatching signal of one sub match-line can be propagated to other sub match-lines using the butterfly connection. When one of the TCAM segments is mismatched with search data, the proposed butterfly match-line scheme turns off more TCAM segments than does the conventional PF-CDPD match-line scheme. All the search operations behind this mismatched segment are terminated. The butterfly match-line scheme increases the dependence between the four parallel match-lines.

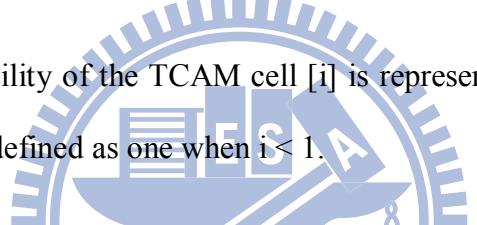
Butterfly match-line scheme not only achieves high performance with a high degree of parallelism but also reduces the power consumption by exploiting butterfly connections. Such a match-line can be implemented using full connections between two stages and thereby propagate all the mismatching information to the subsequent stages. However, it requires a NOR gate with four fan-ins to collect the information about the mismatching associated with the previous stage. Thus, this NOR-gate must provide the driving capacity to trigger the four segments in the subsequent stage. Although it can turn off two more segments than can the butterfly match-line scheme,

the power and performance overheads of the NOR gates with four fan-ins and four fan-outs will dominate the critical path of the match-line. Accordingly, the butterfly connection can turn off the segments behind the mismatching segment most efficiently.

The power analysis of the butterfly match-line scheme is as follows. Before the power formulas of the butterfly match-line schemes can be derived, some assumptions are made for simplicity.

- The power consumption of the search operation is the same in all segments ( $P_{\text{seg}}$ ) when all of the TCAM cells are matched with the search data in one TCAM segment.
- The matching probability of the TCAM cell [i] is represented as  $p_i$  ( $i = 1$  to 144).

The probability  $p_i$  is defined as one when  $i < 1$ .



$$P_{144\text{-bit}} = P_{\text{seg}} \left\{ \sum_{j=0}^2 \left[ \prod_{i=0}^{4+8(j-1)} PS_i \times \left[ PS_{8j-3}PS_{8j-1}(PS_{8j+1} + PS_{8j+2}) + PS_{8j-2}PS_{8j}(PS_{8j+3} + PS_{8j+4}) \right] \right] + \sum_{k=0}^2 \left[ \prod_{i=0}^{8k} PS_i \times \left[ PS_{8k+1}PS_{8k+2}(PS_{8k+5} + PS_{8k+6}) + PS_{8k+3}PS_{8k+4}(PS_{8k+7} + PS_{8k+8}) \right] \right] \right\} \quad (6.1)$$

where  $PS_n = \begin{cases} \prod_{i=6n-5}^{6n} p_i, & n \geq 1 \\ 1, & n \leq 0 \end{cases}$  (probability product of segment n)

Eq. (6.1) is the power formula for the butterfly match-line scheme with 144-bit TCAM cells.  $P_{\text{seg}}$  denotes the power consumption when the TCAM segment is discharged in the evaluation cycle. However, the power consumption of the charge sharing in the dynamic circuit is neglected when the TCAM segment is mismatched. The probability of segment-n,  $PS_n$ , represents the probability that the TCAM segment-n is matched to the search data. Each stage consists of four TCAM segments, and the segments in stage-1 are defined from Seg-1 to Seg-4. The terms,  $j$  and  $k$ , refer

to the odd and even stages of the butterfly match-line scheme, respectively. Therefore, the butterfly match-line scheme can achieve a higher power saving because more TCAM segments are turned off, according to Eq. (1). For example, if Seg-9 as shown in Fig. 6.6 is turned off, all the segments in the gray background will not be activated.

#### 6.4.2 XOR-based Conditional Keeper for Match-Line

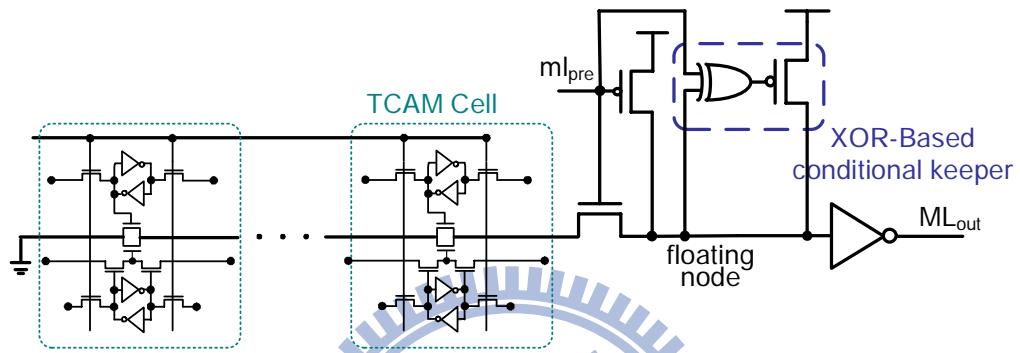


Fig. 6.7 AND-type match-line with XOR-based conditional keeper.

Table 6.1 Control organism of XOR-based conditional keeper.

Clock	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

The AND-type match-line includes high fan-in circuits. However, conventional keepers perform more poorly in terms of propagation delay and power consumption. The main idea of the proposed XOR-based conditional keeper is to ensure that the keeper does not turn on in the dynamic circuit at the beginning of the evaluation phase. Fig. 6.7 and Table 6.1 present the new control signals and their corresponding keeper states. When the match-line pre-charge signal and the floating node are both low, the TCAM circuit is at the beginning of the pre-charge period and the conditional keeper should be turned on to accelerate the pre-charge procedure. When the match-line

pre-charge signal is low and the floating node is high, the pre-charge process is complete and the gate is ready to be evaluated. Therefore, the conditional keeper should be turned off to prevent any impact on the delay and any unnecessary power consumption. When the match-line pre-charge signal and floating node are both high, the match-line is either at the beginning of the evaluation process or stores state HIGH in the floating node at the end of the evaluation process. If it is at the beginning of the evaluation process, the floating node will eventually be at the appropriate voltage as long as the delay of the XOR gate exceeds the propagation delay of the dynamic circuits. Since the delay time of the dynamic circuits is shorter than that of the XOR gate, the conditional keeper is slightly turned on at the beginning of the evaluation process and is fully turned on or off as determined by the final value that is stored in the floating node. When the match-line pre-charge signal is high and the floating node is low, the evaluation mode has been completed and the final value stored in the floating node is low. Consequently, the conditional keeper should be fully turned off.

An XOR gate is required to generate the desired control signals.

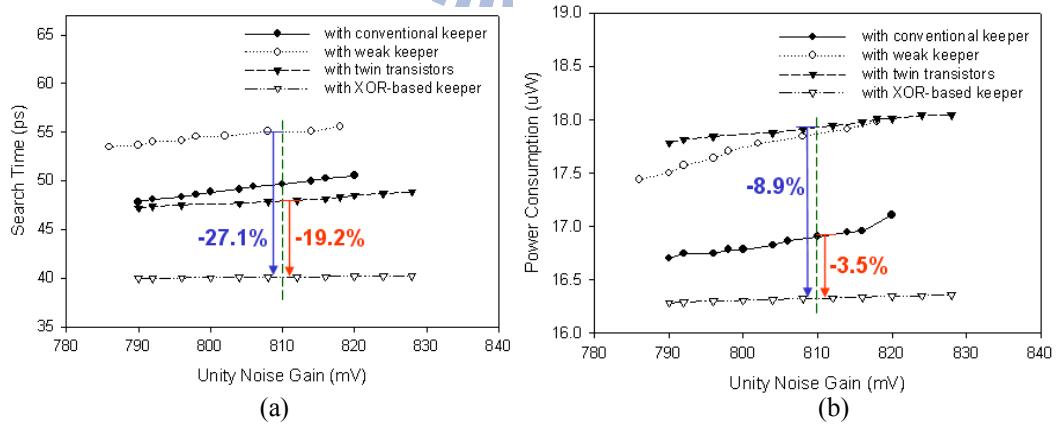


Fig. 6.8 (a) Search time (b) Power consumption versus UNG margin for different keepers.

Four different types of 8-bit AND-type match-line schemes are adopted for the performance comparison. The first one is match-line scheme with conventional keeper, depicted as Fig. 6.4. The second one uses weak keeper to match-line scheme. The third one is

match-line scheme with the twin transistors technique. The fourth one is the proposed AND-type match-lines scheme with XOR-based conditional keeper. During the noise tolerance comparison, what we concern about is not the actual size of the keeper device or actual size of twin transistors but the ability to resist noises. This ability is testified by the widely used Unity Noise Gain (UNG) margin [6.29]. Fig. 6.8(a) and Fig. 6.8(b), present the search time and power consumption versus unity noise gain margin for four types of AND-type match-line, respectively. For example, when UNG is at 810mV, 19.2% search time reduction and 3.5% power saving compared to conventional keeper up-sizing are achieved by using XOR-based conditional keeper. Based on the same condition, compared to weak keeper, 27.1% search time reduction and 8.9% power saving are realized with XOR-based keeper. Even though the twin transistors technique is suitable for deep fan-in dynamic circuits, the performance is worse than XOR-based keeper. According to the simulation results, 16.3% search time growth and 8.9% power consumption increase are achieved compared to XOR-based keeper when UNG is at 810mV. However, the area overhead with XOR-based keeper is 1.8% and 1.0% compared to conventional keeper and weak keeper, respectively.

The noise-tolerant energy-efficient match-line employs the co-design of the architecture and the circuit. Based on the butterfly connection, the inverters behind the XOR-based conditional keepers and the AND gate can together be represented an NOR gate, as presented in Fig. 6.9. Therefore, to reduce the search time overhead caused by the butterfly connection, the XOR-based conditional keeper approach can be used to reduce the delay associated with the critical path of the match-line. Clearly, the increase in the propagation delay of a TCAM segment by the XOR-based conditional keeper is small. Furthermore, the gate delay of a NOR gate is shorter than that of an AND gate. Accordingly, the proposed butterfly match-line scheme with a XOR-based conditional keeper exhibits high performance, because of the high degree

of parallelism. It also saves power by using an XOR-based conditional keeper and turning off the mismatched segments. This match-line also simultaneously reduces the search time and power consumption. Moreover, it is resilient against noise effectively.

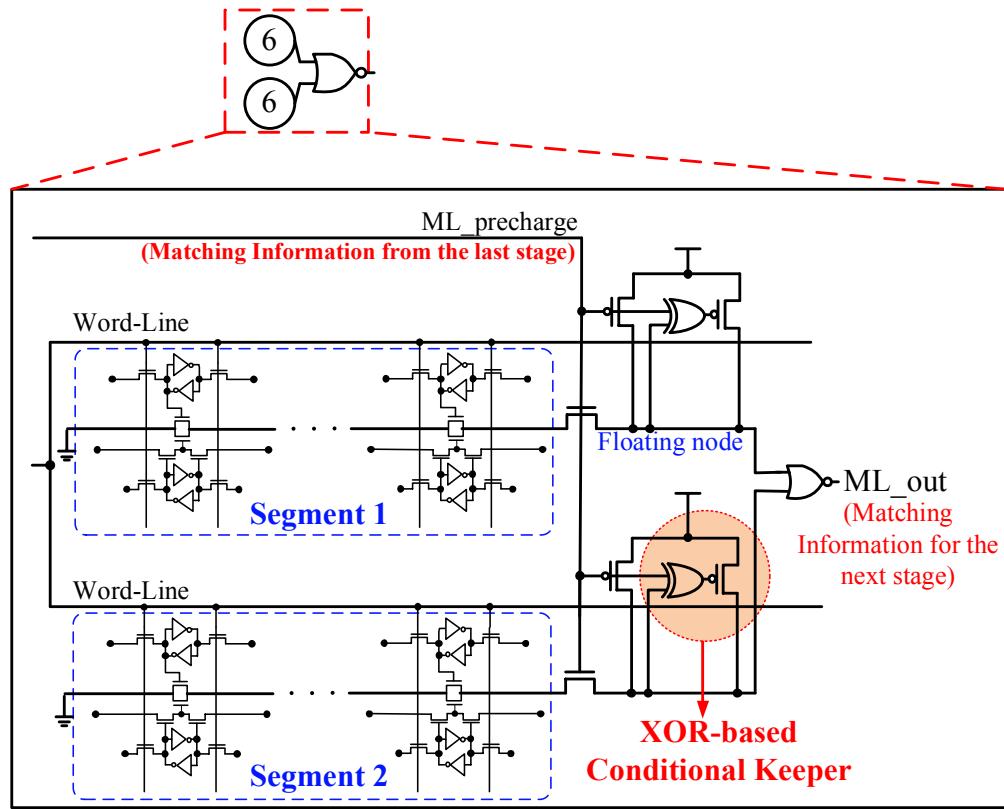


Fig. 6.9 Butterfly connection style with XOR-based conditional keeper and don't-care based power gating scheme.

## 6.5 Don't-Care Based Hierarchy Search-Line Scheme

In a manner determined by an address-lookup function, network routers forward data packets from an incoming port to an outgoing port. Additionally, a TCAM cell stores an “X” value as a mask. The “X” value represents a don’t-care state, referring both “0” and “1”, and allows a wildcard operation. The wildcard operation is a feature of packet forwarding in Internet routers and involves the storing of an “X” value in a cell to yield a match, regardless of the input bit. Furthermore, the list of routing tables can be rearranged and maintained using rule table management with continuous don’t-care X patterns, as presented in Fig. 6.10. Based on the continuous don’t-care X

pattern and pre-fix pattern, a hierarchy search-line scheme, a super cut-off power gating technique and a multi-mode data-retention power gating technique, are proposed. This section presents the don't-care-based hierarchy search-line scheme. The following sections will elucidate the other procedures.

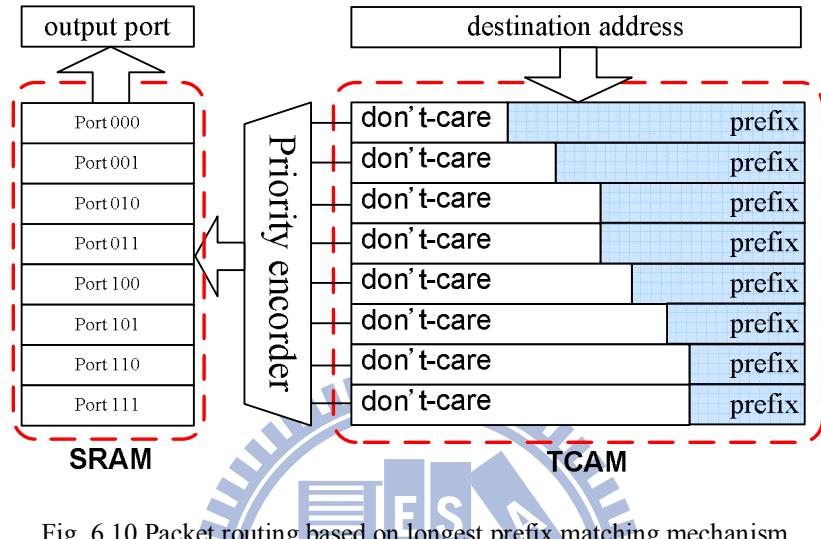


Fig. 6.10 Packet routing based on longest prefix matching mechanism.

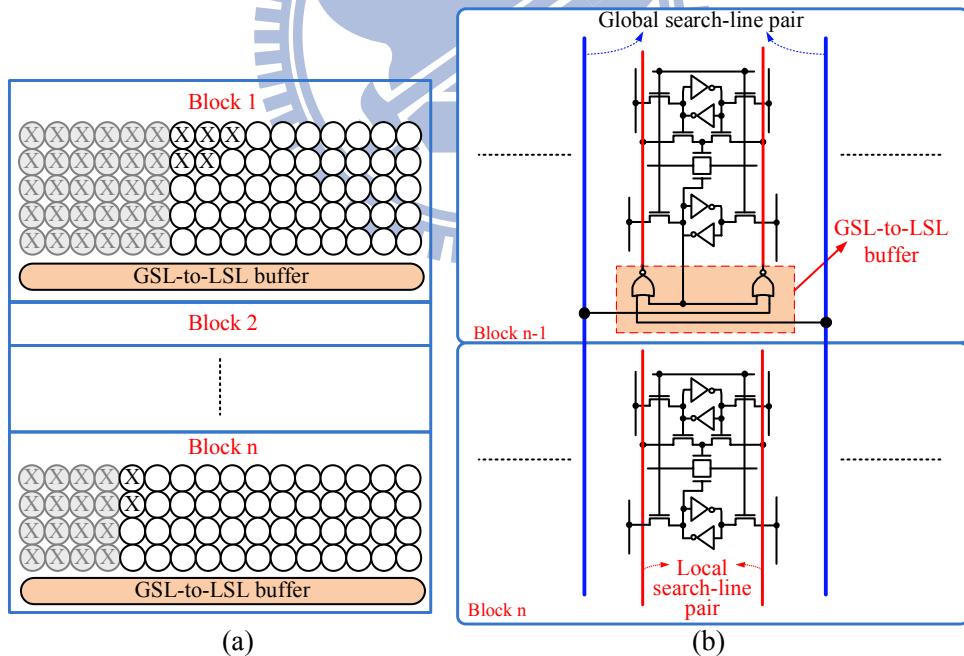


Fig. 6.11 (a) A simplified architecture (b) Circuit implementation of don't-care based hierarchical search-line scheme.

The don't-care-based hierarchical search-line scheme is utilized to decrease the switching capacitances and switching activities without adding any search time

overhead. If the TCAM cell is don't-care, then the matching signal is independent of the searching data and the search-lines can be disabled to reduce consumed power. Therefore, the hierarchy search-line scheme divides the search-lines into a two-level hierarchy of global search-lines (GSLs) and local search-lines (LSLs). The GSLs are active in every cycle, but whether the LSLs are active depends on don't-care cells. Fig. 6.11(a) displays a simplified architecture of the don't-care-based hierarchical search-line scheme. The entire TCAM is divided into  $n$  sub-blocks, each of which is composed of numerous match-lines and one global search-line to local search-line buffer (GSL-to-LSL buffer). In the IPv6 addressing lookup tables associated with TCAM, the prefixes are arranged in order by prefix lengths. The longest prefix is located at the top of the TCAM. Accordingly, the upper TCAM cells must be don't-care terms. The GSL-to-LSL buffers are controlled by the data in the don't-care cells which is stored in the bottom word of each block. Moreover, Fig. 6.11(b) presents the circuit implementation of the don't-care-based hierarchical search-line.

During a search operation, the search data are propagated to GSL first. And then, the search data would be propagated to the LSLs according to the don't-care data. Therefore, the GSL-to-LSL buffer is used to determine whether search data on GSL will be broadcast to LSL. If the don't-care state is true, the local search-line pair will always be discharged to ground without a voltage swing. However, if the don't-care state is false, then the local search line will be active and perform the search operation after the search data are propagated from GSL. Accordingly, the hierarchy search-line scheme can reduce the power consumption by decreasing the switching capacitances and switching activities without adding any searching time overhead.

Unlike the conventional hierarchical search-line [6.23], the don't-care-based hierarchical search-line does not increase the search delay of the search operation. Fig.

6.12 shows the timing diagram of the don't-care-based hierarchical search-line during a search operation. The search-lines are activated when the clock is high, and the delay path is through the D flip-flops, GSL buffers and GSL-to-LSL buffers. For a 50% duty cycle clock, the GSL-to-LSL buffer delay is clearly shorter than half of a clock cycle, and the critical path depends on the match-lines. Before the negative edge of the clock, search data have been transferred to the LSL. Therefore, the search time, which depends on the buffer delay ( $ml\_pre$  buf delay) and the search delay of the match-lines, dominates the clock period. By exploiting the critical delay, the proposed search-line scheme can save power without adding any timing overhead.

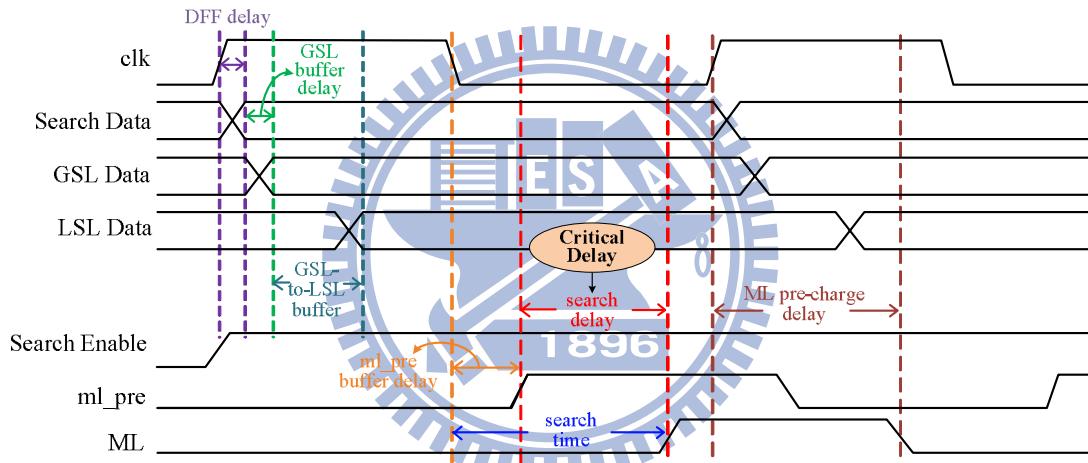


Fig. 6.12 Timing analysis of don't-care based hierarchical search-line scheme.

## 6.6 Don't-Care Based Hierarchy Power Gating Techniques

Two power gating techniques, multi-mode data-retention power gating and super cut-off power gating, are proposed based on the characteristic of continuous don't-care X patterns. The super cut-off power gating and the data-retention power gating are applied in don't-care cells and storage cells, respectively. They reduce the standby power by reducing leakage current. Fig. 6.13 schematically depicts super cut-off power gating and multi-mode data-retention power gating in a TCAM segment. Additionally, the sleep signal indicates that the TCAM macro is in standby mode, and

the most significant bit (MSB) and last significant bit (LSB) of a TCAM segment reveal the don't-care states in this segment. If both MSB and LSB are 1, then all TCAM cells in this segment are in the don't-care states. Therefore, the operations of the proposed power gating techniques depend on the above control signals. Power gating approaches are described and analyzed below.

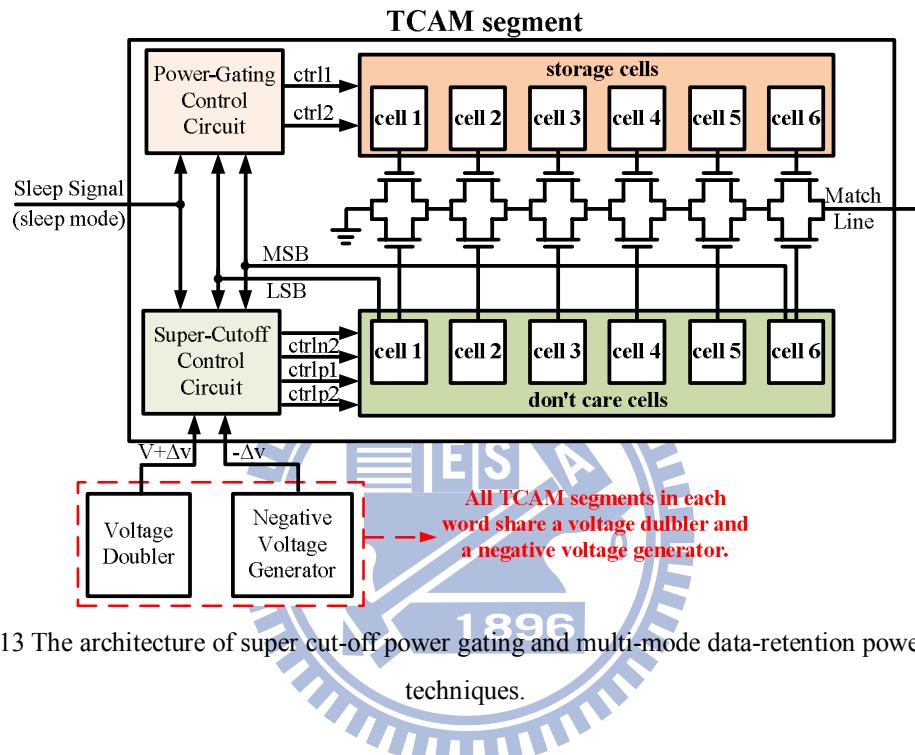


Fig. 6.13 The architecture of super cut-off power gating and multi-mode data-retention power gating techniques.

### 6.6.1 Multi-Mode Data-Retention Power Gating

The multi-mode data-retention power gating is utilized in storage cells, and modified from the power gating devices that we proposed elsewhere [6.30]. Fig. 6.14 presents the circuitry associated with multi-mode data-retention power gating. The regular power gating NMOS (M1) and a diode-connected NMOS (M3), whose functions are similar to those in our earlier investigation [6.30], are adopted herein, but the PMOS diode is replaced by M1 to improve speed in the active mode. Additionally, an additional NMOS (M2) is added to the stack to increase the virtual ground voltage. The diode-connected NMOS causes the virtual ground to saturate to a

limited voltage which depends on the threshold voltage of M3.

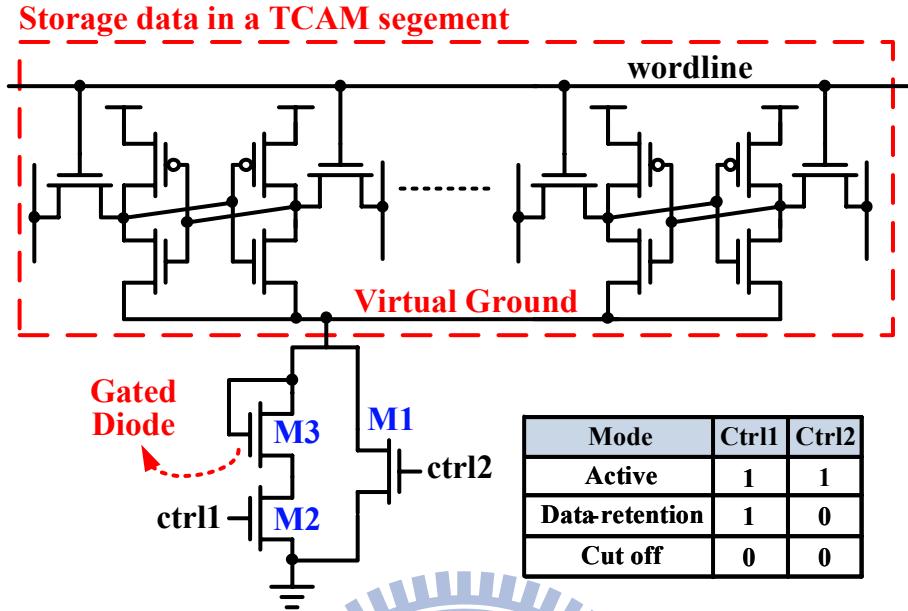


Fig. 6.14 Multi-mode data-retention power gating technique.

The three modes of data-retention power gating control circuit are active, data-retention and cut-off modes, respectively. Fig. 6.14 also presents the truth table of the control signals in these three modes. When the storage cell is in the active mode, both control signals are set to high and the power gating transistors are turned on to support full-speed operation. When the circuit enters the data-retention mode, ctrl1 will be high and ctrl2 will be low. M3 just represents a diode, which causes the voltage of the virtual ground to saturate and provides a sufficient noise margin. If the virtual ground voltage established by the leakage current increases beyond the saturation value, then M3 will be turned on and the virtual ground voltage will discharge through it. Accordingly, the virtual ground declines back to the saturated value, guaranteeing the stability of the data storage cells. When all don't-care cells in a TCAM segment are set to the don't-care state, the data that are stored in storage cells are meaningless and can be eliminated. Therefore, the data-retention power gating control circuit will be switched to the cut off mode while the most significant

bit (MSB) of the don't-care cells is set to high.

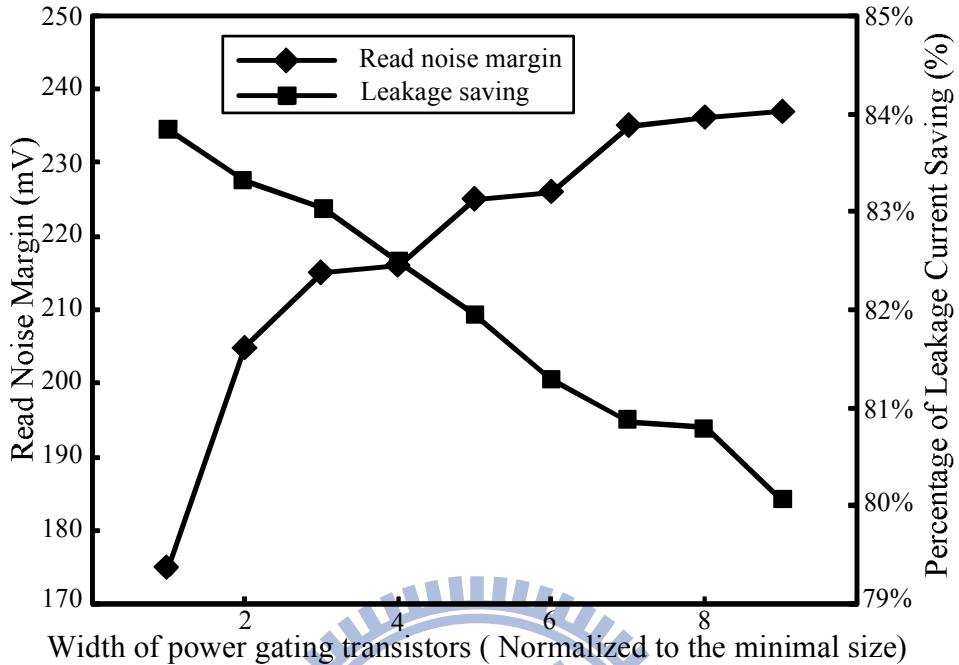


Fig. 6.15 Relation between noise margin, leakage saving and scale factor.

Noise margin analysis is essential for SRAM cells. SRAM cells should secure stored data from destruction by data access operations or noise interference. Decreasing the widths of the gating transistors used in the multi-mode data-retention power gating technique reduces leakage power. However, it also reduces the noise margin. Fig. 6.15 plots the relationship among the sizes of the gating transistors, the reduction in leakage current, and the read noise margin. The left y-axis represents the read noise margin in the active mode, whereas the right y-axis represents the percentage reduction in the leakage current that can be achieved in the cut-off mode. The widths of the power gating transistors (M1 and M2) are normalized to the minimum possible size (0.12 $\mu$ m in UMC 65nm CMOS technology). As the width of the power gating transistors increases, the read noise margin is improved. However, the leakage current in the cut-off mode is also increased. A trade-off exists between the decline in leakage current and the noise margin. In data retention mode, the noise

margin and leakage saving are determined by the virtual ground voltage, which in turn is primarily determined by the width of the diode-connected transistor (M3). M3 also dominates the hold noise margin and the leakage power saving in the data retention mode.

### 6.6.2 Super Cut-Off Power Gating

Super cut-off power gating is a well-known power gating approach, and has been explored elsewhere [6.31]. However, the super cut-off technique is not favorable when applied to SRAMs, because in this approach the signals that are transmitted to the cell array must preserve the data even in stand-by mode. Furthermore, super cut-off power gating suffers from a long wake-up time and a high current peak in the sleep-to-active transition. As determined by the continuous don't-care X patterns, don't-care cells in a segment are all set to 1 or 0, except in the segment that is located at the boundary of the don't-care X patterns. However, the super cut-off power gating technique is effective for don't-care cells without a long wake-up time and a high current peak.

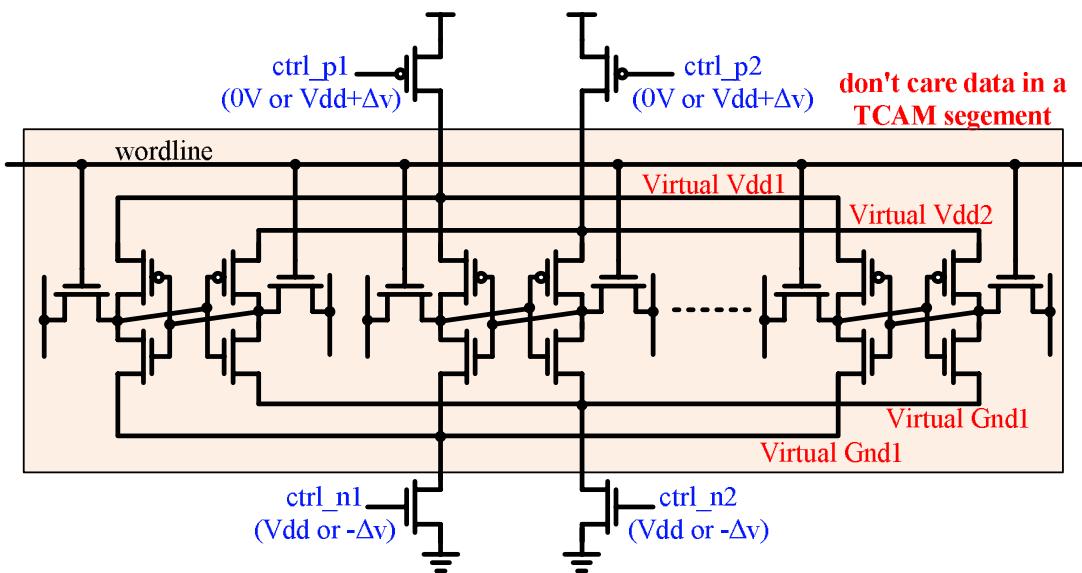


Fig. 6.16 Super cut-off power gating technique.

Table 6.2 The corresponding control signals under different operations.

Operation Mode	(ctrl_p1, ctrl_p2, ctrl_n1, ctrl_n2)								* $\Delta v = V_t$			
	(MSB,LSB) = (0,0)				(MSB,LSB) = (0,1)				(MSB,LSB) = (1,1)			
Write	0v	0v	Vdd	Vdd	0v	0v	Vdd	Vdd	0v	0v	Vdd	Vdd
Search	Vdd+ $\Delta v$	0v	Vdd	- $\Delta v$	0v	0v	Vdd	Vdd	0v	Vdd+ $\Delta v$	- $\Delta v$	Vdd
Standby	Vdd+ $\Delta v$	0v	Vdd	- $\Delta v$	0v	0v	Vdd	Vdd	0v	Vdd+ $\Delta v$	- $\Delta v$	Vdd

Fig. 6.16 presents the circuit implementation of super cut-off don't-care cells.

Table 6.2 also shows the corresponding control signals (P1, P2, N1, N2) under various operations. When all the don't-care cells in the segment are set to 1, P1 and N2 are turned on to preserve the data in the don't-care cells. Additionally, P2 and N1 are turned off to reduce leakage currents. The situation is similar to that in which all the don't-care cells in the segment are set to 0, where P1 and N2 are turned off and P2 and N1 are turned on. Accordingly, each segment contains four cut-off switch transistors to reduce leakage power and preserve data in the don't-care cells. The cut-off voltages are  $Vdd+\Delta v$  for PMOS and  $-\Delta v$  for NMOS, which voltages are generated from a VBB generator and a voltage doubler [6.32], [6.33]. Since the sub-threshold leakage depends exponentially on the gate-source voltage, the super cut-off technique considerably reduces the sub-threshold leakage. Restated, when a larger overdrive voltage is applied to the cut-off transistor, the sub-threshold leakage is lower. However, two design-limiting factors should be considered. First, the increase in the gate-drain voltage caused by the overdrive voltage at the gate may increase the gate leakage, which also depends strongly on the gate-source voltage. The second limiting factor is the over-stress voltage across the gate-source node, which may cause the gate oxide to break down. In UMC 65nm CMOS technology, the oxide stress voltage will not exceed VDD when the gate overdrive voltage is less than 0.5V. Additionally, since the sub-threshold leakage is exponentially related to the gate-source voltage in the weak inversion region, the sub-threshold leakage may be

significantly reduced by increasing the cut-off switch gate voltage. Hence, the overdrive voltage in this approach is set to 0.3V, which is the threshold voltage of a normal  $V_t$  device in UMC 65nm CMOS technology.

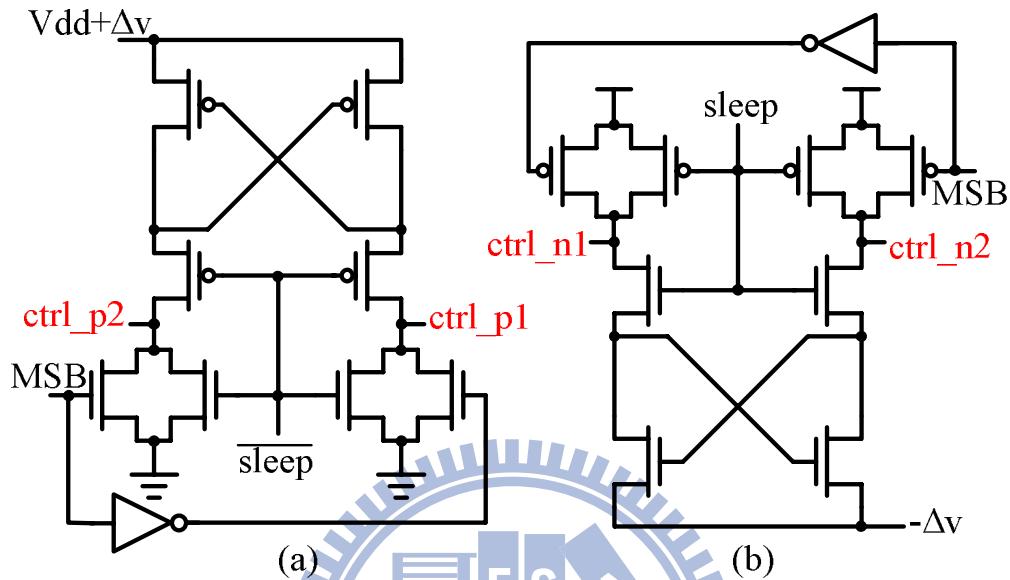


Fig. 6.17 Control circuits for (a) PMOS (b) NMOS cut-off switch.

The super cut-off control circuit between the voltage generators and the don't-care cells must comprise level converters that prevent short currents. Fig. 6.17(a) and Fig. 6.17(b) display the control circuits that are associated with the PMOS super cut-off switch and the NMOS super cut-off switch, respectively. They are constructed from cross coupling level shifters to prevent short circuits. When the sleep signal is 0, the cross coupling circuits are isolated from the output ports; the outputs are discharged to ground for PMOS switching and charged to high for NMOS switching for the write operation. If the sleep signal goes to high, then the cross coupling circuits are connected to the outputs and the outputs are evaluated using MSB. Control signals with steady voltage can be maintained and static short circuits, which otherwise would be induced by  $V_{ds} > V_{gs}$  when the drain voltage is  $V_{dd} + \Delta v$  or  $-\Delta v$ , can be prevented.

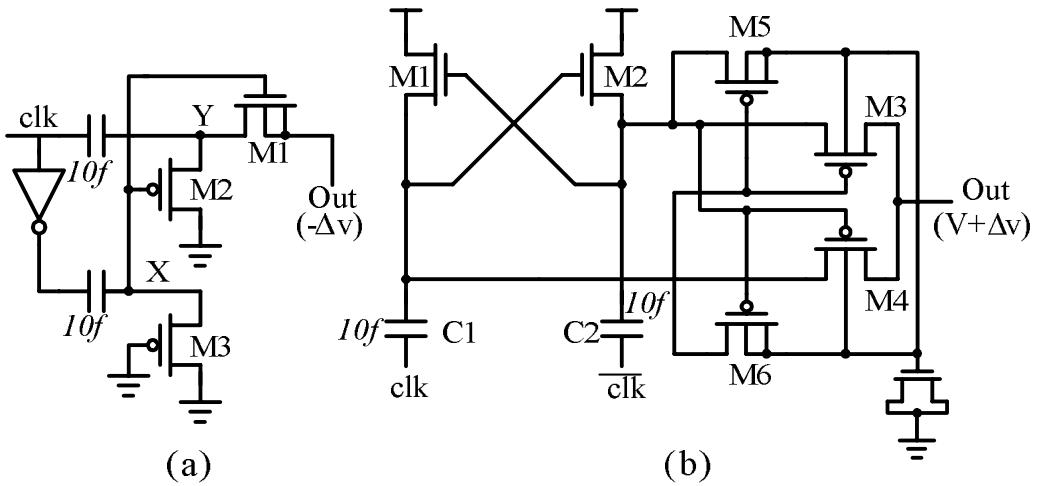


Fig. 6.18 (a) VBB generator (b) Voltage doubler for super cut-off power gating.

Fig. 6.18(a) and Fig. 6.18(b) show the VBB generator and voltage doubler, respectively. The voltage doubler utilizes a dual series switch and applies the principle of bulk switching [6.33]. M3 and M4 are series switches, and M5 and M6 switch to the highest voltage. The bodies of M3 and M4, output node and chip substrate comprise vertical PnP bipolar transistors. Since M5 and M6 switch the bodies of M3 and M4 to the highest voltage, the circuit is latch-up immune. Unlike positive voltage generators, negative-pumping circuits generate voltages that are lower than ground (potential = 0). The VBB generator employs a negative-pumping circuit [6.32]. When clk is high, node X reaches  $(-V_{dd} + |V_{tp}|)$  and node n4 is grounded through M2. When clk goes low, node Y is pulled down to  $-V_{dd}$ . Meanwhile, the high voltage at node X turns on M1, and pulls the output down to  $-V_{dd}$ .

To optimize the area overhead and the power overhead of the VBB generator and the voltage doubler, Fig. 6.19 presents the timing and power analysis of the voltage generators for various numbers of match-lines. For each match-line, the equivalent loads of the PMOS and NMOS gating devices are approximately 10fF. Therefore, the load currents of the VBB generator and voltage doubler are equal, 50nA, for each match-line. As the number of match-lines that share one VBB generator and one

voltage doubler increases, the power overhead declines, but the transient response time of the VBB generators and the voltage doublers increases. If more than 32 match-lines share one voltage generator, the transient response time increases rapidly. However, the power overhead does not significantly decrease. Accordingly, the optimal number of match-lines to share one voltage generator is between 16 and 32. To reduce the area overhead of the voltage generators, 32 match-lines are employed in the energy-efficient TCAM macro. Hence, the 256-word TCAM is divided into eight blocks.

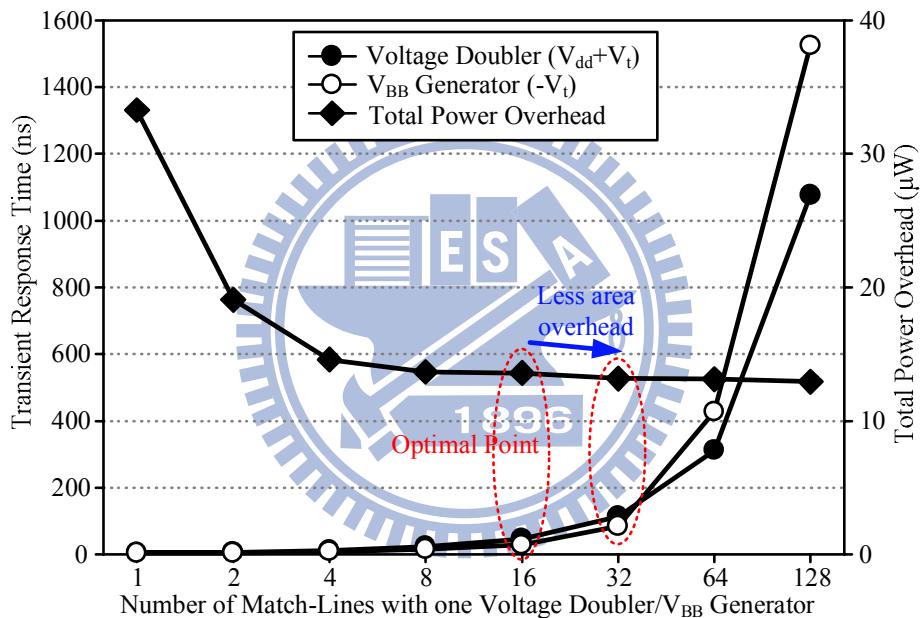


Fig. 6.19 Analysis of voltage generators for different number of match-lines.

The power reduction of the super cut-off power gating technique is independent of the percentage of don't-care bits in the TCAM macro because the super cut-off power gating can be applied to all segments in a match-line, except for the segment at the boundary of continuous don't-care patterns. Moreover, the super cut-off power gating technique does not degrade the cell stability in standby mode. In fact, a charge path and a discharge path exist from the supply and the ground to node-1 and node-0, respectively. Accordingly, the super cut-off power gating technique has two main advantages. First, the cell stability is maintained and the data are not destroyed when

the charge/discharge paths in don't-care cells are preserved. Second, nearly half of the SRAM cells in the TCAM macro, the don't-care cells, operate in super cut-off mode, reducing the leakage current.

## 6.7 TCAM Macro Implementation and Measurements

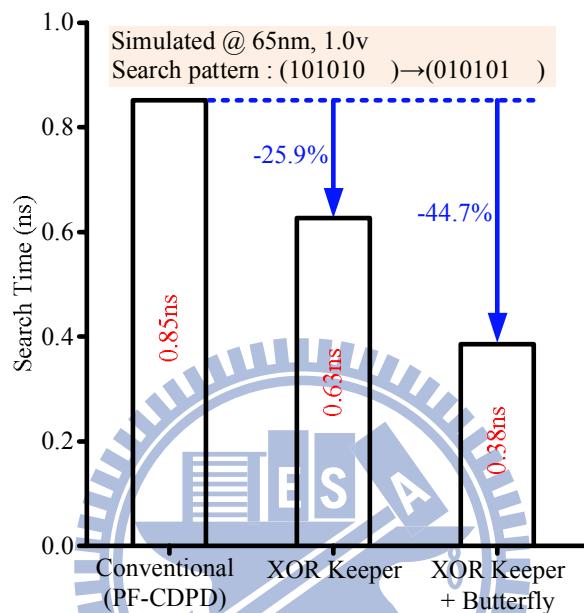


Fig. 6.20 Analysis of the search time under different energy-efficient schemes.

In this section, an energy-efficient 256x144 TCAM macro was implemented using UMC 65nm CMOS technology. The energy-efficient 256x144 TCAM performs a high-speed and low-power search operation using PF-CDPD circuits, the butterfly match-line scheme, the XOR-based conditional keeper and the don't-care based hierarchy search-line scheme. Furthermore, the TCAM macro reduces the leakage current via multi-mode data-retention power gating and super cut-off power gating techniques. The TCAM array is 256-word x 144-bit, and is divided into eight blocks. Each block is composed of 64 match-lines with a VBB generator, a voltage doubler and GSL-to-LSL buffers. In the VBB generator and the voltage doubler, the cut-off voltage is  $1.3v (V_{dd} + V_t)$  for the PMOS cut-off switch and  $-0.3v (-V_t)$  for the NMOS cut-off switch. Each block, in addition, has its own VBB generator and voltage

doubler, which is associated with the 32x24 six-bit TCAM segments. The clock rate of the charge-pump circuits equals the system clock rate.

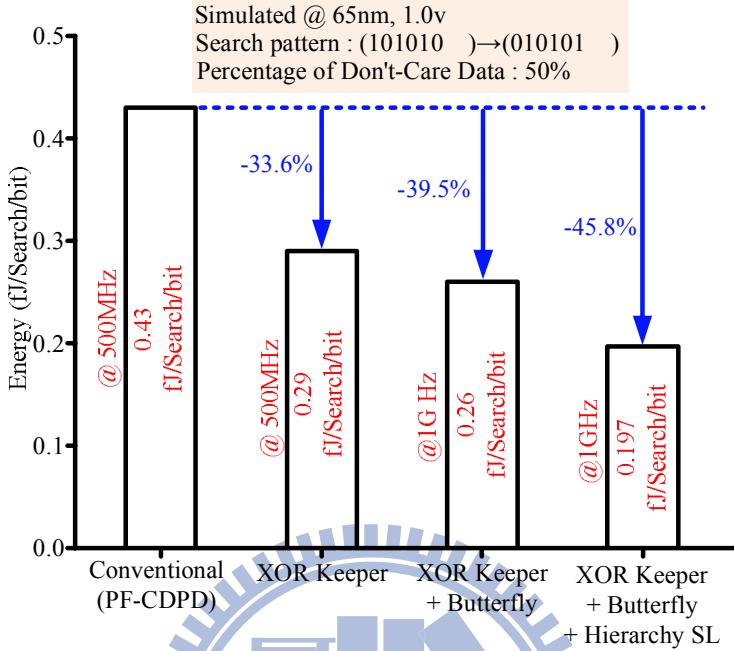


Fig. 6.21 Analysis of the energy consumption under different energy-efficient schemes.

The butterfly match-line with XOR-based conditional keepers can reduce search time based on PF-CDPD circuits. Fig. 6.20 shows the reduction in the search time achieved using the butterfly match-line and the XOR-based conditional keeper. The search time is defined in Fig. 6.12, and the simulation is based on the worst-case patterns from (101010....) to (010101....). The butterfly match-line with XOR-based conditional keepers reduces the search time by 44.7% below that achieved using the conventional PF-CDPD match-line scheme. Fig. 6.21 also presents the analysis of the energy consumption under different energy-efficient schemes. The simulation is conducted using the worst-case patterns, 50% don't care data, 1.0v supply voltage and the highest possible frequencies, which are listed at the side of the bars. The XOR-conditional keeper reduces the energy consumption to 33.6% below that of the PF-CDPD match-line with the conventional keeper, and exhibits enhanced immunity to noise. Additionally, the butterfly match-line scheme with XOR

conditional keepers reduces the energy consumption by 39.5%. The don't-care based-hierarchy search-line schemes further reduce the energy by approximately 45.8%, reducing the power consumption of the search-lines.

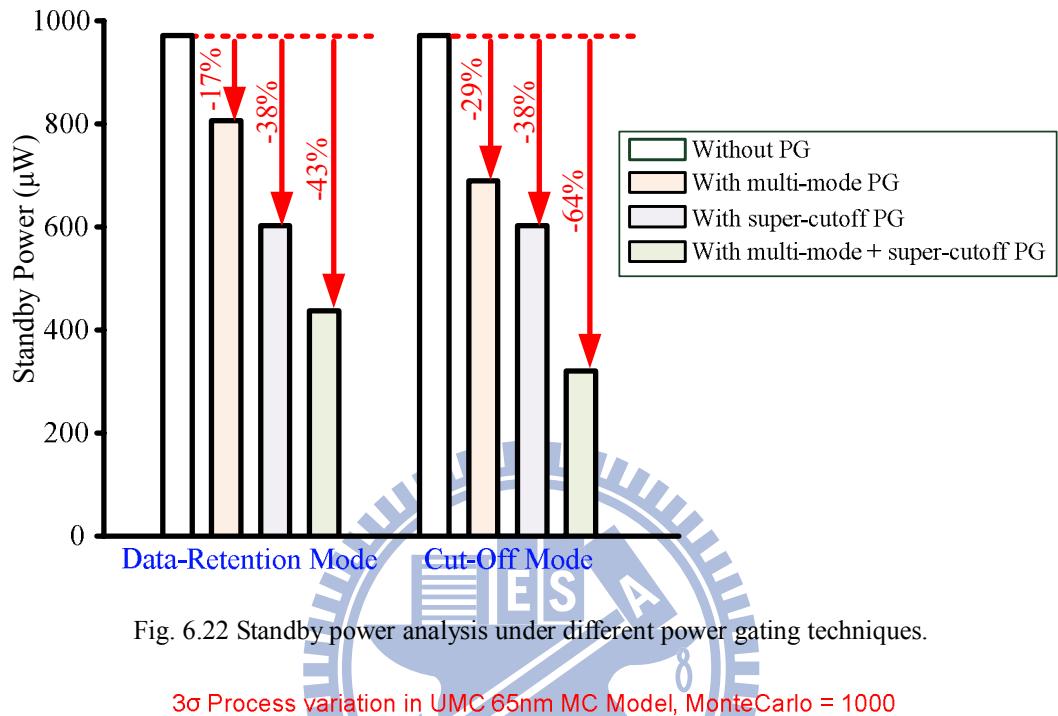


Fig. 6.22 Standby power analysis under different power gating techniques.

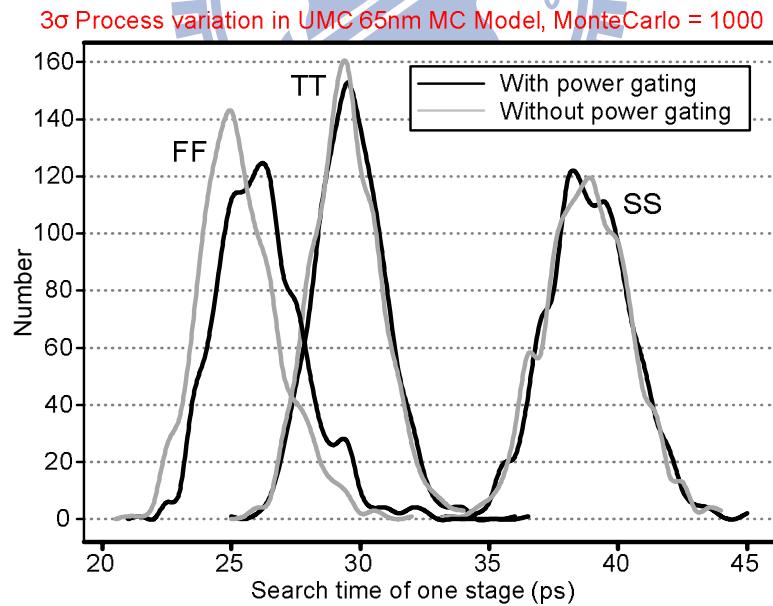


Fig. 6.23 Search time of one stage under  $3\sigma$  process variations.

The standby power with and without power gating is analyzed. Fig. 6.22 presents the standby power under the different power gating approaches. Super cut-off power gating reduces the leakage current by 38% by increasing the gate-drain voltage.

Additionally, the multi-mode data-retention power gating reduces the leakage power by 17% and 29% in the data-retention and cut-off modes, respectively. If both power gating techniques are applied, the leakage power is further reduced. However, the process variation has become one of the critical design challenges in shrinking to nano-scale technologies [6.34]. Fig. 6.23 presents the search delay of one TCAM stage with/without the power gating techniques under  $3\sigma$  process variations in UMC 65nm MC (Monte-Carlo) model with three corners (TT, SS and FF). The search delay with power gating is similar to that without power gating since the discharging paths are determined by NMOSs, which are controlled by the comparisons of stored data and search data. Fortunately, the comparisons were finished before pre-charging the floating node via the PF-CDPD circuitry [6.21]. Therefore, the variation in search time in the TCAM stage, ranging from 20ps–45ps, is tolerant.

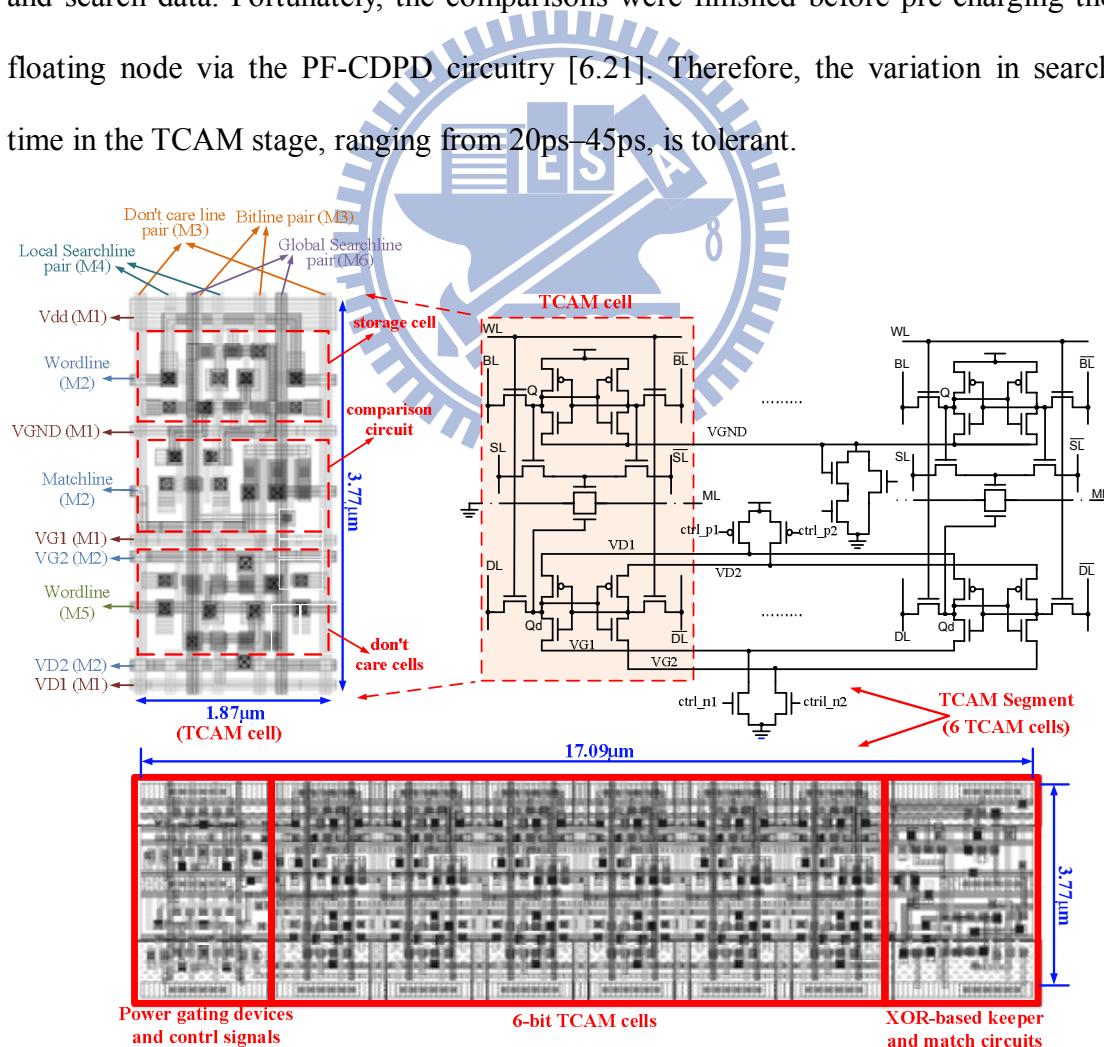


Fig. 6.24 Layout view of 1-bit TCAM cell and a TCAM segment with 6-bit TCAM cells.

Based on the power gating techniques and hierarchy search-line schemes, each TCAM cell needs extra metal layers to route extra virtual ground, virtual Vdd and global search-lines. Fig. 6.24 shows the layout view of a 1-bit TCAM cell and a TCAM segment with 6-bit TCAM cells. A TCAM cell is composed two SRAM cells and a comparison circuit. The size of a TCAM cell is  $3.77 \times 1.87 \mu\text{m}^2$ . To reduce the loading capacitance, the search-line pair and the bit-line pair are separated to reduce both the search-time and the power consumption. However, two more vertical lines must then be added. Therefore, the bit line pair (M4), the don't-care line pair (M3), the LSL pair (M4) and the GSL (M6) pair are routed along the vertical axis. Two horizontal metal lines, M1 and M2, are preserved as power lines, virtual ground and virtual Vdd. Furthermore, the upper word-lines and match-line are along the horizontal axis using M2. The lower word-line is routed through M5. M5 is also utilized to realize the butterfly connection. Nevertheless, the area of the proposed TCAM cell is 3 times as large as a 65nm TCAM cell proposed in [6.35]. The addition of metal tracks further increase the bit-cell area which is a wire-limited design based on butterfly match-line, hierarchy search-line and power gating. Therefore, a trade-off exists between the area efficiency and energy efficiency while adopting energy-efficient schemes. Fig. 6.24 also presents the layout of a TCAM segment that consists of 6-bit TCAM cells, a XOR-based conditional keeper, a match-line circuit, power gating devices and power gating control circuits. The keeper and match-line circuit are placed at the right-hand side of a TCAM segment. The left-hand side contains the power-gating devices and the corresponding control signals. The size of one TCAM segment is  $3.77 \times 17.09 \mu\text{m}^2$ . Fig. 6.25 displays the layout of the 256x144 TCAM array. Fig. 6.25 also presents the floorplan and the test-chip micrograph. The 256x144 TCAM array is divided into eight blocks, each with 32 match-lines. The GSL-to-LSL buffers are placed between pairs of adjacent blocks. The total sizes of the

TCAM array and the test chip are 426x1010 and  $683 \times 1165 \mu\text{m}^2$ , respectively.

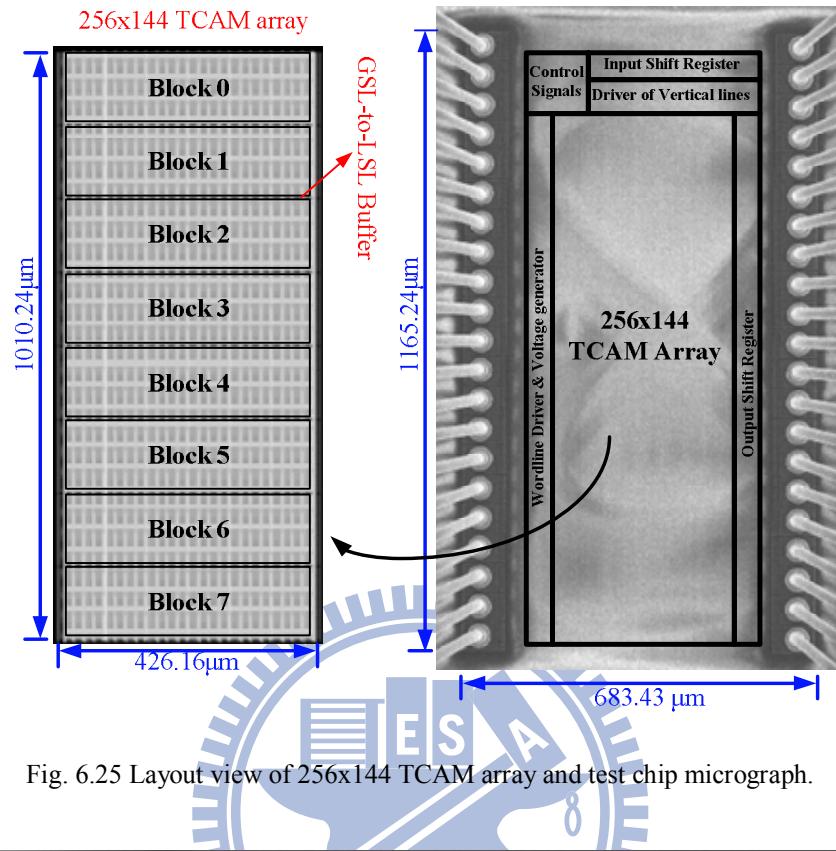


Fig. 6.25 Layout view of 256x144 TCAM array and test chip micrograph.

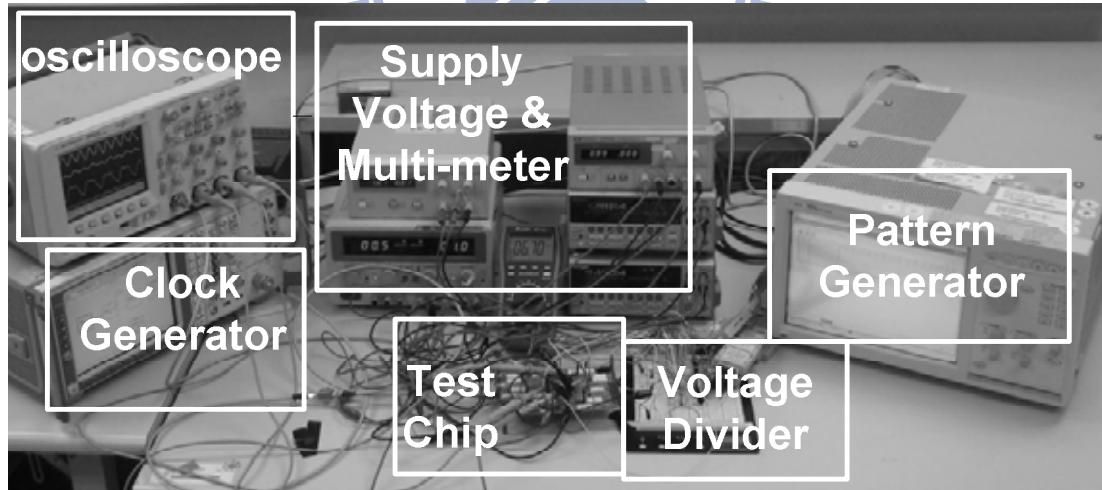


Fig. 6.26 Measurement setup.

Fig. 6.26 displays the measurement setup. A pattern generator and voltage dividers are utilized to set up the TCAM array, and two clocks, the shift clock and the search clock, are generated by the pattern generator and clock generator, respectively. Fig. 6.27(a) shows the block diagram of this test chip. The test strategy is divided into two

phases. The first phase is to input the data in the TCAM array, and the second phase is the search operation. In the first phase, 12x12 shift registers and a word-line shifter are adopted to write data in the 256x144 TCAM array. After the storage data and the don't-care data are set, the search data are inputted into two 12x12 shifter registers. Then, the test enters the second phase. In the second phase, the shift clock is turned off, and the search clock is turned on to perform the search operation. The search data in these two 12x12 shifter registers are thereby transferred into the search-lines. To measure the worst-case patterns, the search data in these two shifter registers are set to be mutually complementary. After the search operation has been complete, the output shifter shifts the results of the match-lines and checks the functionality of the TCAM macro. Fig. 6.27(b) plots the corresponding waveform. However, this test chip cannot measure the search delay because the search frequency is limited by the pad.

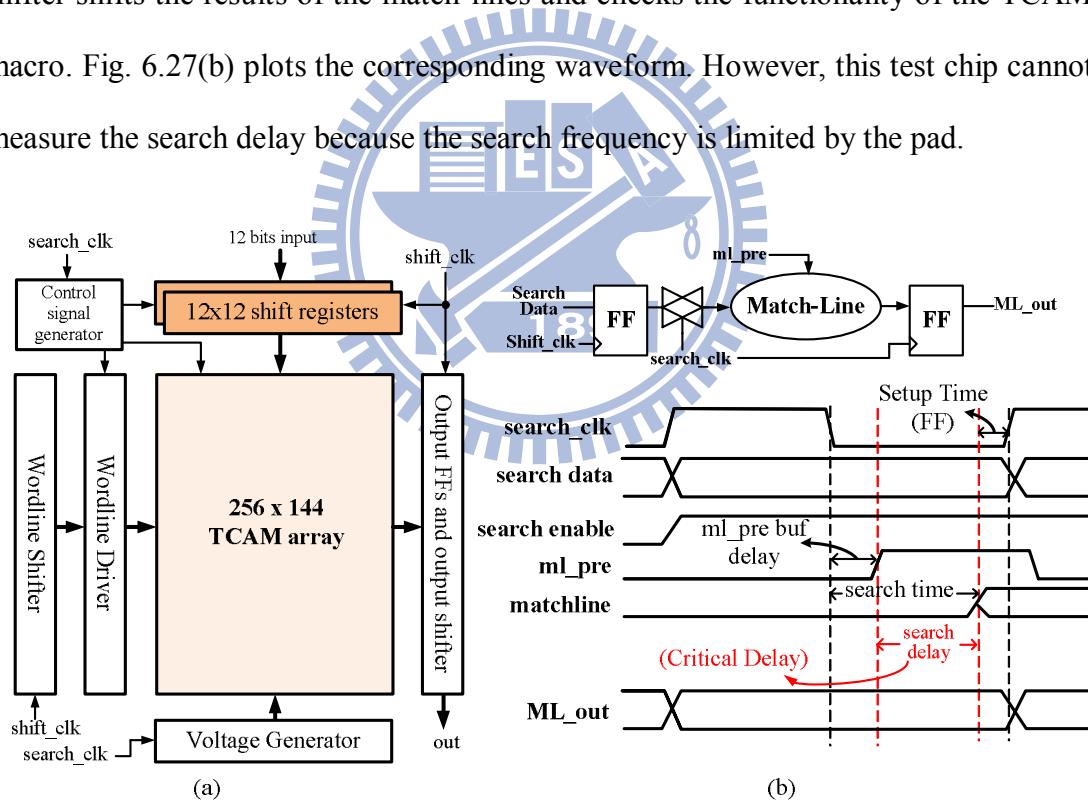


Fig. 6.27 (a) Block diagram of the test chip (b) Test strategy.

The TCAM array is operated at 1V and 400MHz, which frequencies are limited by the pads. Fig. 6.28 shows the energy matrix for various percentages of don't-care data. Additionally, the applying search data are as one pattern in one of the two shift registers and the same pattern or the same one inverted in the other shifter register. As

the proportion of don't-care data increases, the energy dissipation is reduced because the LSLs are disabled. When the same pattern is input repeatedly, the energy consumption is almost independent of the percentage of don't-care data.

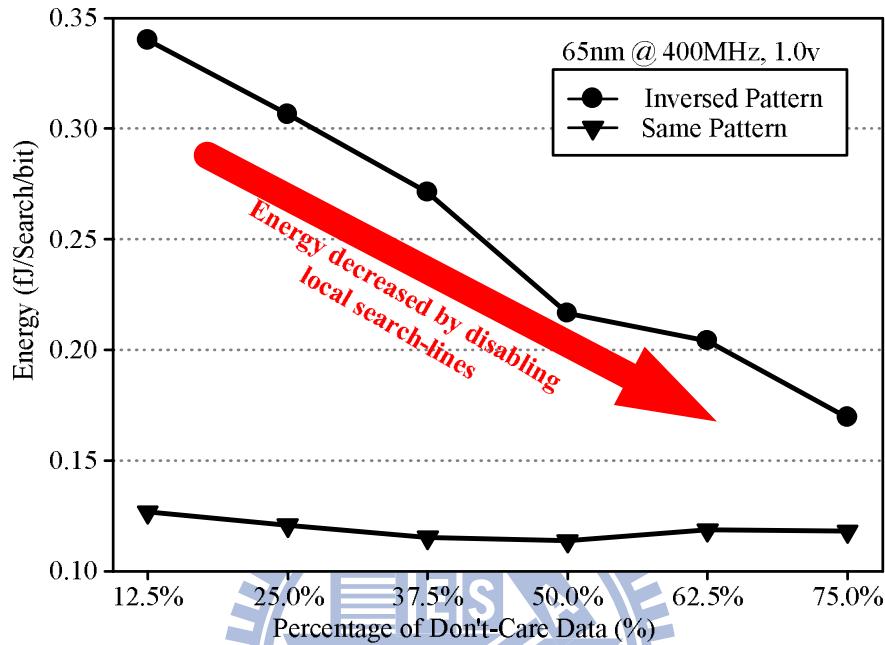


Fig. 6.28 Energy consumption under different don't care patterns.

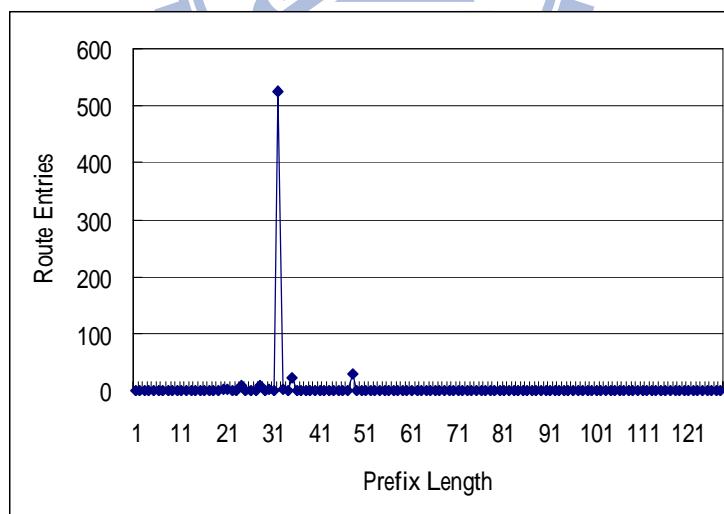


Fig. 6.29 Network address prefix distribution of IPv6.

For achieving the accurate measurement results, the properties of routing tables in IPv6 should be considered [6.36]. From the prefix length distribution in the router of 6Bone, 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% as shown in Fig. 6.29.

Table 6.3 Features Summary and Comparisons.

	Hybrid [19] (JSSC 2005)	PF-CDPD [15] (JSSC 2006)	Range Match [16] (ISSCC 2006)	Tree-style [17, 21] (JSSC 2008)	Charge Recycling [22] (ASSCC 2008)	This Work	
						Simulation	Test Chip Measurement
configuration	1024x144	256x128	512x144	256x128	1024x144	256x144	
Technology	100 nm	0.18 $\mu$ m	0.13 $\mu$ m	0.18 $\mu$ m	0.18 $\mu$ m	65 nm	
Area (mm <sup>2</sup> )	2.8x4.2 (chip)	1.21x0.56 (core)	1.5x1.7 (core)	0.84x0.92 (core)	3.67x0.98 (core)	1.01x0.43 (core)	
Supply voltage (V)	1.2 V	1.8 V	1.2 V	1.8 V	1.8V	1.0 V	
Search time (ns)	2.20 ns	2.10 ns	4.80 ns	1.56 ns	100MHz	0.38ns	400MHz
Energy metric (fJ/bit/search)	0.700	2.330	0.590	1.420	6.300	0.121	0.165
Normalized Search time T* (ns)	1.716	1.365	2.880	1.014	N.A.	0.380	N.A.
Normalized Energy metric E* (fJ/bit/search)	0.316	0.260	0.205	0.158	0.702	0.113	0.165

According to this statistics, more than half of TCAM cells in a routing table are don't-care bits. Moreover, most TCAM segments would not be activated in search operations. Applying 50% don't-care data, Table 1 summarizes the performance of the proposed TCAM and other recently proposed TCAMs. The normalized factors are modified from another investigation [36], and Eq. (6.2) is derived. E and T represent the energy and the delay time, respectively. E\* and T\* denote the normalized factors for 65nm CMOS technology and a supply voltage of 1.0 V. The search time determined from the simulation result is 0.38ns. Additionally, the energy metric of the test chip is 0.165fJ/bit/search. According to Table 1, the comparisons of the proposed energy-efficient TCAM and other TCAM approaches reveal that the proposed TCAM has a very competitive search speed and energy efficiency.

$$\begin{cases} E^* = E \times \left(\frac{65}{\text{Technology}}\right) \times \left(\frac{1.0}{V_{DD}}\right)^2 \\ T^* = T \times \left(\frac{65}{\text{Technology}}\right) \times \left(\frac{V_{DD}}{1.0}\right) \end{cases} \quad (6.2)$$

Although the normalized energy of a tree-style match-line is a little smaller than that of the proposed approach, the tree-style match-line suffers from leakage problems when the technology is shrunk to the nano-scale. Fig. 6.30 presents the standby power

with different power gating modes. The measured standby power differs from the simulated results. The simulation results reveal that the leakage saving in the cut-off mode exceeds that in the in data-retention mode when the multi-mode data-retention power gating is used. The measurements demonstrate that the standby power in the cut-off mode is almost the same as that in the data-retention mode, perhaps because of the floating node of the virtual ground in the multi-mode data-retention power gating.

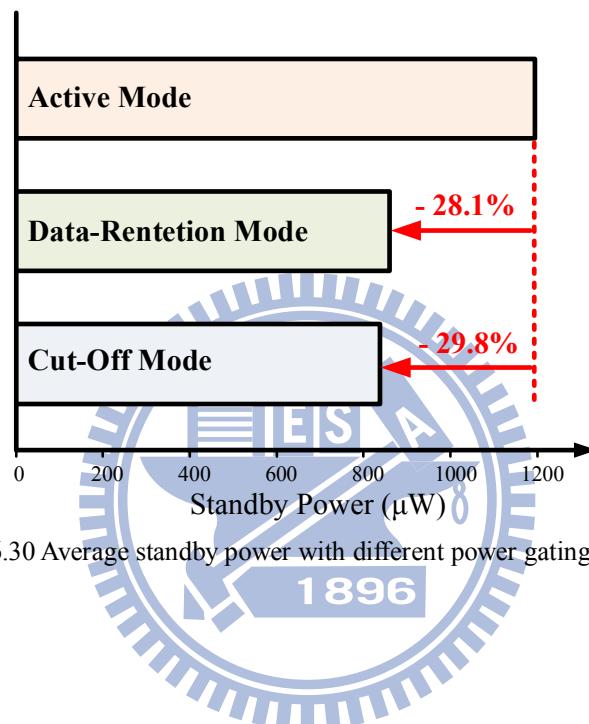
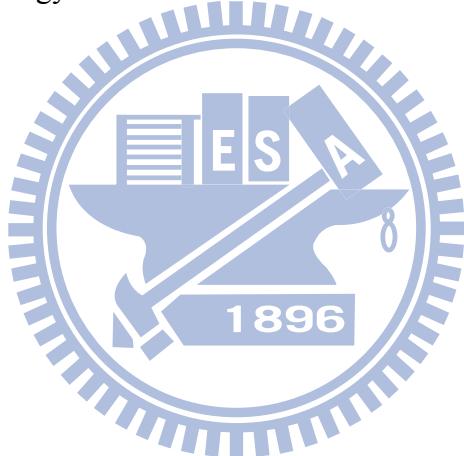


Fig. 6.30 Average standby power with different power gating modes.

## 6.8 Summary

This chapter presents an energy-efficient TCAM design approach, which exploits the co-design of the architecture and circuit. To achieve low-power and high-performance TCAM architecture, a don't-care-based hierarchy search-line scheme and a butterfly match-line scheme are presented. The hierarchy search-line reduces power consumption by reducing not only the switching activity but also the capacitance of the search-lines. The butterfly match-line scheme reduces not only the search time but also the power consumption by exploiting a high degree of parallelism and dependence of sub match-lines. The proposed TCAM match-line is also implemented using a noise-tolerant XOR-based conditional keeper to perform energy-efficient search operations for match-lines. As technologies advance, leakage

currents increasingly dominate the overall power consumption of nano-scale technologies. Accordingly, the super cut-off technique and the multi-mode data-retention power gating technique are utilized to reduce leakage currents significantly in standby mode. Furthermore, the super cut-off power gating technique also reduces the leakage current in the search operations without reducing the search time or destroying the noise margin. Based on UMC 65nm CMOS technology, an energy-efficient 256-word x 144-bit TCAM array is implemented. The experimental results demonstrate that the leakage power is reduced by 19.3% and the energy metric of the TCAM is 0.165fJ/bit/search. The proposed TCAM will be very useful in further nano-scale CMOS technology.



# ***Chapter 7: On-Demand Memory Sub-system for Multi-Task Wireless Video Entertainment Systems***

With increasing demands on ubiquitous wireless high-data-rate multimedia services, it is critical to have efficient processing capability and a merging multi-task system to sustain the growth. . Moreover, green computing design concepts become essential to handle concurrent multimedia services at minimum processing power. In this chapter, an energy-efficient on-demand memory sub-system is proposed to overcome the challenges in the multi-task system design that needs to support wireless video entertainment. Therefore, the proposed on-demand memory sub-system can provide high bandwidth and energy-efficient memory-centric on-chip data communication for wireless video entertainment systems via memory management units (MMUs), consisting of private MMUs (p-MMUs) and a centralized MMU (c-NNU). The p-MMUs can dynamically allocate the memory resource for network data buffering to reduce the stall of processor elements based on the proposed borrowing mechanism. Furthermore, the c-MMU manages centralized on-chip memories (L2 cache) and off-chip memories. For different memory requirements of the processor elements (PEs) in a heterogeneous multi-core system-on-chip (SoC), adaptive memory resource allocation is adopted using the proposed adaptive cache control. Additionally, in order to access off-chip DRAM efficiently, an external memory interface is designed in c-MMU. By considering the characteristics of the wireless video data, an inter-layer pre-fetch mechanism and an efficient data allocation scheme are proposed to reduce the cache miss rate and memory energy consumptions for Scalable Video Coding (SVC).

## 7.1 Background

For multi-task wireless video entertainment systems, a heterogeneous multi-core SoC provides an integrated solution for supporting great amount of data computation [7.1]. With the increasing data computation, the increasing demand of the memory capacity and bandwidth is a bottleneck in multi-core SoCs since the overall performance of PEs is much faster than that of the memory. Additionally, multimedia technologies are usually utilized in multi-task systems for video processing that spawns brand new industries and services, such as digital video recording, video-on-demand services, high-definition TV, and digital home sever. Generally, a great amount of memory requirements are required for high quality or multiple scalable level video processing. Therefore, an efficient memory sub-system is indispensable that provides enough memory space and high data bandwidth for satisfying the video real-time requirement. Furthermore, the strategy of the memory management has become one of the design challenges. Accordingly, the organization of memory sub-system for a multi-task system will affect the system performance dramatically.

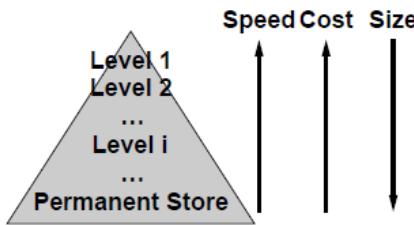
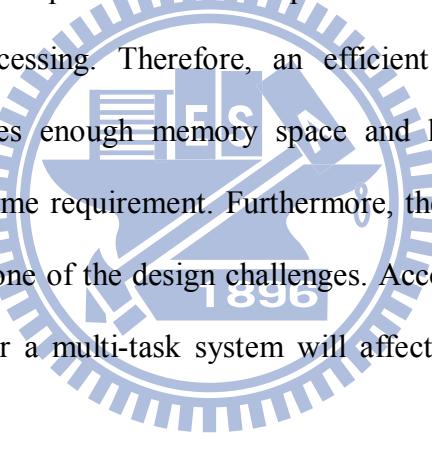


Fig. 7.1 Memory hierarchy.

In multi-core SoCs, a well-organized memory hierarchy realizes both advantages simultaneously, high data bandwidth and small memory capability. The memory hierarchy is base on a principle of locality including temporal and spatial locality. In general, the memory hierarchy is described as a pyramid which is as shown in Fig. 7.1

[7.2]. The higher levels have better performance than the lower levels, but the cost per bit is on the contrary. In ideal, the PE can access the data with the best memory access performance and have large memory space. Nowadays, the hierarchy is formed with Cache (SRAM), DRAM and Disk storage elements, and the performance and energy consumption of different storage elements are list in Table 7.1. No storage element can provide low cost, high bandwidth and low latency simultaneously. Therefore, the memory hierarchy is built to hide the negative characteristics and to gain the positive characteristics of these memory technologies.

Table 7.1 Cost-performance for various memory technologies

Technology	Bytes per Access (typ.)	Latency per Access	Cost per Megabyte <sup>a</sup>	Energy per Access
On-chip Cache	10	100 of picoseconds	\$1–100	1 nJ
Off-chip Cache	100	Nanoseconds	\$1–10	10–100 nJ
DRAM	1000 (internally fetched)	10–100 nanoseconds	\$0.1	1–100 nJ (per device)
Disk	1000	Milliseconds	\$0.001	100–1000 mJ

DRAMs have been widely used for providing additional off-chip memory storage capacity. Compare to the SRAM, the circuit of a DRAM cell is “*dynamic*” because the capacitors storing electrons are not perfect devices, and their eventual leakage requires that, to retain information stored there, each capacitor in the DRAM must be periodically refreshed [7.2]. However, the cost per bit is much cheaper than the SRAM. In the memory hierarchy, DRAM is a level below the on-chip SRAM (cache). DRAM architecture is usually composed of the data memories, address decoders, row buffer, mode register, data buffer. Fig. 7.2 shows a simplified block diagram. In this example, four banks share the address bus and command bus. Each bank has its own row decoder, column decoder, and sense amplifier. The mode register stores the DRAM operation mode, including burst length (BL), column address strobe latency (CL), and burst type, etc. Users can set the value of the mode

register through address bus with proper command.

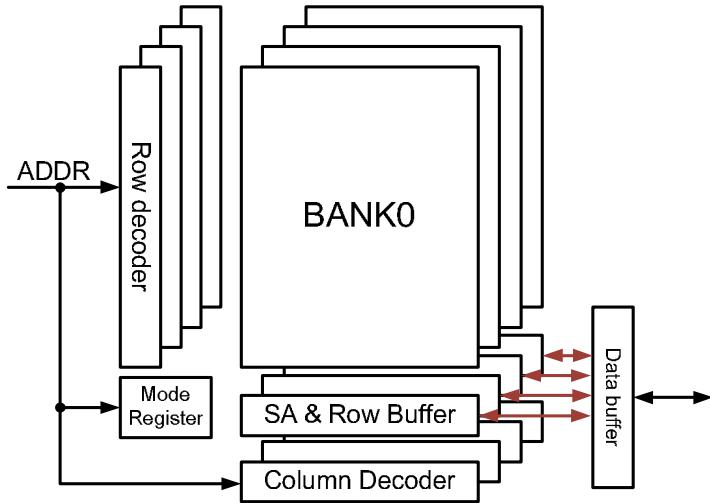


Fig. 7.2 Simplified architecture of a DRAM.

According to different applications or systems, the memory controllers of DRAM reduce the memory access latency to support the real-time video environment. Therefore, efficient external memory controllers were proposed to improve the overall system performance significantly according to the regular memory access behavior in video processing. Moreover, while decoding the video frames, the video decoder generally has regular memory access characteristics. According to the regular behavior, some techniques have been presented to improve the decoding performance via data re-arrangement and history-based predictions [7.3]-[7.7]. Additionally, an interpolation window reuse (IWR) scheme was proposed to reduce data access for the overlapped data [7.8]. Moreover, a cache-based architecture was proposed to reuse intra-MB overlapped data [7.9] or to reuse inter-MB and inter-MB overlapped data [7.10]. Additionally, a multilayer and quality-aware memory controller was proposed to satisfy different memory access requirement [7.11]. Fig. 7.3 shows the configurations of different layers of the multilayer memory controller. Layer 0, memory interface socket (MIS), is a configurable, programmable, and high-efficient SDRAM controller for designers to rapidly integrate SDRAM subsystem into their

designs. And layer 1, quality-aware scheduler (QAS), is a memory controller layer which has the capability to provide quality-of-service guarantees including minimum access latencies and fine-grained bandwidth allocation for heterogeneous processor elements in SoC designs. Moreover, Layer 2 built-in address generator (BAG) designed for multimedia processor elements can effectively reduce the address bus traffic and therefore further increase the efficiency of on-chip communication. An efficient external memory interface was presented for multiprocessor platform with the separated data communication path and memory access path [7.12]. Accordingly, a self-optimizing memory controller base on reinforcement learning concept has been proposed to enhance the memory bandwidth [7.13]. Moreover, for adjusting the memory access scheduling dynamically, a ME-LREQ (Memory Efficiency-Least Request) policy was present [7.14].

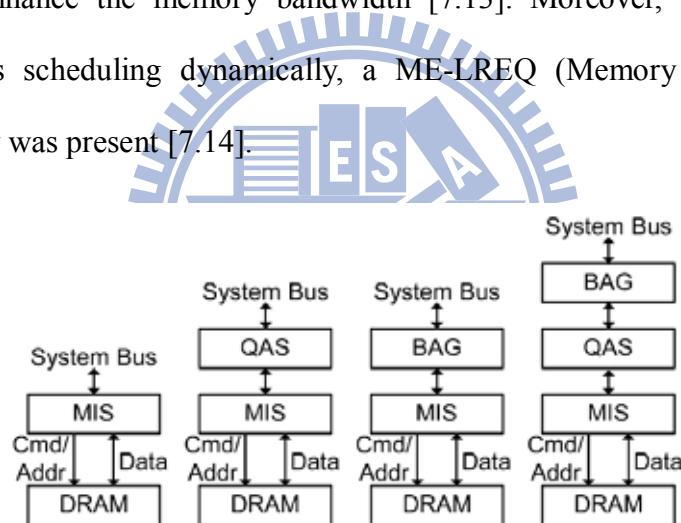


Fig. 7.3 Configurations of different layers of the proposed memory controller.

How to manage and utilize the memory is the most important issue for constructing a multi-core SoC. Accordingly, large amounts of high speed and low power memories are indispensable for multi-task and multi-system emerging. These memories should be able to support diverse memory requirement of different PEs in a system. Therefore, a memory-centric on-chip data communication platform with on-demand memory sub-system is proposed for wireless video entertainment systems. The on-demand memory sub-system provides high bandwidth and low power memory accesses for a heterogeneous multi-core SoC via MMUs. Furthermore, MMUs can

support adaptive memory resource assignment for different PEs according to their memory access behaviors. Based on the regular characteristics of the wireless video data, an inter-layer pre-fetch mechanism and an efficient data allocation scheme are proposed to reduce the cache miss rate and memory energy consumptions for SVC. Before describing the proposed on-demand memory sub-system, a wireless video entertainment system will be introduced in the following section first.

## 7.2 Wireless Video Entertainment System

With the ongoing advancement in digital and communication techniques, digital home service becomes a trend nowadays. In the daily life, home is the personal headquarters for living, keeping personal assets and information. If the digital home services are applied, the residents will effectively participate in any events happening in the local, national and global communities without unnecessary travel. Digital home technique integrates wireless, wired physical transmission and multimedia real-time processes. With wireless communication technique, mobile electronic product, such as cell phone, PDA or notebook, can be used for transmitting or receiving the message by a certain sever. People can monitor and control the situation which something or somebody happens at home remotely or receive immediate video what they want. But nowadays, many kinds of communication protocol have been used such as WLAN, bluetooth, WiMAX or LTE techniques. In order to support a variety of protocols, a heterogeneous network system would be constructed to provide an adaptable PE to process various communications. However, the current technologies and systems cannot effectively meet the requirements of these digital homes for some reasons [7.15]. First, there are too many incompatible and not interoperable systems and standards, and each system only work for one particular application, using a particular physical transmission medium, and incompatible

hardware and firmware. The Second one is the throughput of the future digital home system may require up to 10Gps (gigabit-per-second), but the current home networking technologies is below 1Gps. Therefore, the system bandwidth must be improved. Furthermore, the scalability, security and power are also the problems. To solve these problems, wireless severs and multimedia PEs can be integrated in a multi-core SoC. In order to serve various transmit channels, a multi-task wireless video entertainment system is as shown in Fig. 7.4. The analog front-end system receives and digitizes the wireless signals. Then the data is processed by an integrated, high performance digital system.

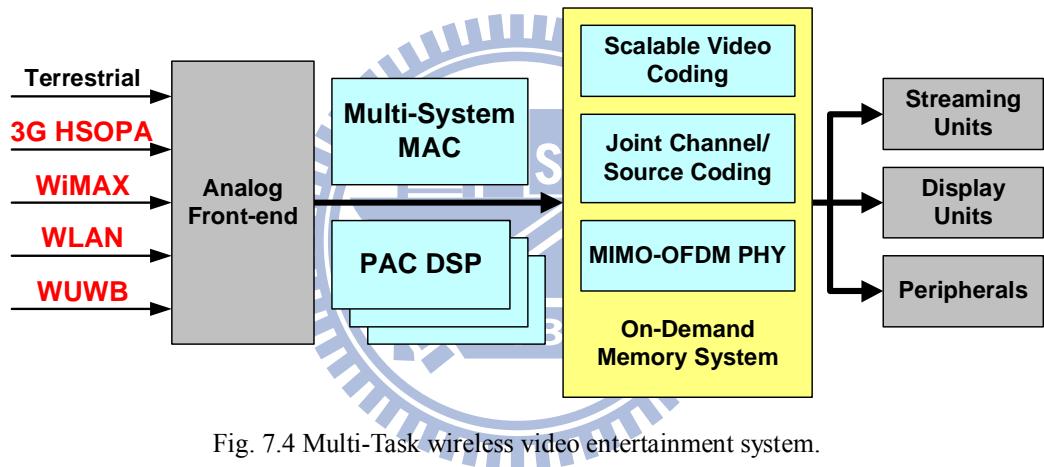


Fig. 7.4 Multi-Task wireless video entertainment system.

In order to support video entertainment and various communication standards for digital home service, a wireless video entertainment system is developed via four functional blocks (PEs). The block diagram of the wireless video entertainment system transceiver is as shown in Fig. 7.5, including the transmitter and the receiver. In this system, SVC is utilized to provide spatial, temporal and quality scalability of the video sequences. Additionally, Luby Transform (LT) coding, the first class of practical fountain codes that are near optimal erasure correcting codes, is applied to have high channel reliability. Media access control (MAC) module is the interface between application layer and the physical layer, and Wireless Processing Unit (WPU)

handles the wireless signal processing including multi-standard baseband control and MIMO-OFDM (multiple-input and multiple-output orthogonal frequency division multiplexing). These functional blocks are developed by other groups in NCTU and integrated via the proposed memory-centric on-chip data communication platform, which is called eH-II platform. The system specification of the receiver is listed in Table 7.2. Additionally, the details of WPU, MAC, LT coding and SVC coding will be described in the following sections.

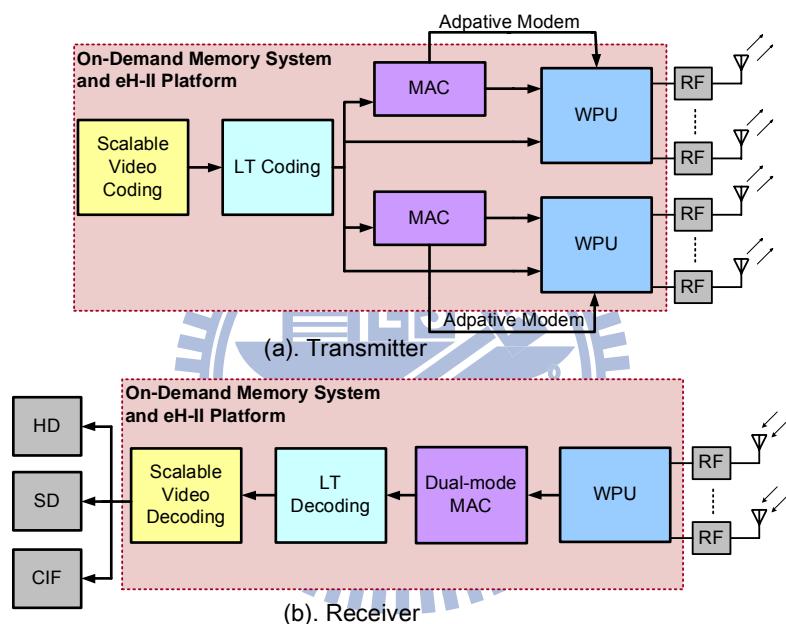


Fig. 7.5 Block diagram of the wireless video entertainment system.

Table 7.2 System Specification (Receiver)

	<b>WPU (4x4 MIMO)</b>	<b>MAC</b>	<b>LT Coding</b>	<b>SVC</b>
Input data rate	160Mbps (4Gb/s)	7.8Mbps	7.8Mbps	1333Kbps
Output throughput	7.8Mbps (with a 64-QAM modulation)	7.8Mbps	7.8Mbps	17.4Mbps
Memory access bandwidth	222.4Mbps	124.8Mbps	124.8Mbps	78.69Mbps
Memory Size (Required)	6.25KB	2MB	1MB	11.34MB (a GOP)

### 7.2.1 Wireless Processing Unit (WPU)

The WPU is designed as a frequency domain (FD) modem with the single-FFT architecture. Additionally, the single-FFT architecture for multi-standard baseband is suitable for IEEE802.11a/b/g/n/VHT and IEEE 802.15.3a/c. The architecture is shown as in Fig. 7.1. There are three key components in this architecture, including frequency-domain (FD) synchronization, FD adaptive sampling and single carrier frequency domain equalizer (SC-FDE). The features of the three components are as follows.

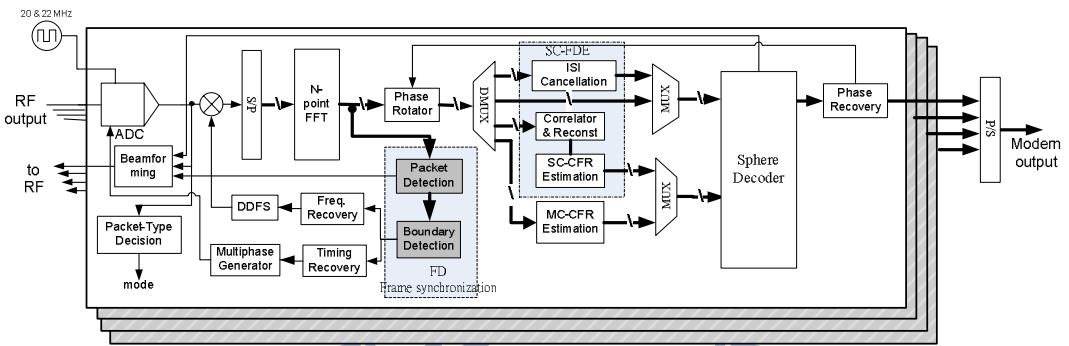


Fig. 7.6 Single-FFT Architecture for MIMO Modem.

- Frequency-domain (FD) synchronization
  1. FD Adaptive Sampling
  2. FD Boundary Decision
  3. FD Anti-I/Q Phase Recovery
- Single carrier frequency domain equalizer (SC-FDE)
  1. Frequency-domain channel estimation (FD-CE)
  2. Frequency-domain ISI cancellation
  3. Frequency-domain data decision
- FD adaptive sampling
  1. 6-symbol Lock
  2. 32 multiphase clocking

3. Boundaryless
4. Tolerance of -30,000~40,000 ppm SCO (sampling clock offset)

Moreover, For the FD boundary decision, it contains the following features, only 1% detection error with low SNR (<5 dB) and high CFO (carrier frequency offset) tolerance. It is a trellis-based detector, and can be used both for DSSS (direct sequence spread spectrum) and OFDM systems. Fig. 7.7 displays the architecture, and it contains 3 key components, including a metric computation, a sorter and an iterative sequence searcher. Additionally, for FD anti-I/Q phase recovery, it contain following features.

1. Pseudo CFO injection
2. Compatible with conventional method (Moose)
3. Robust in IQ mismatch
4. Gain error: 2dB
5. Phase error:20

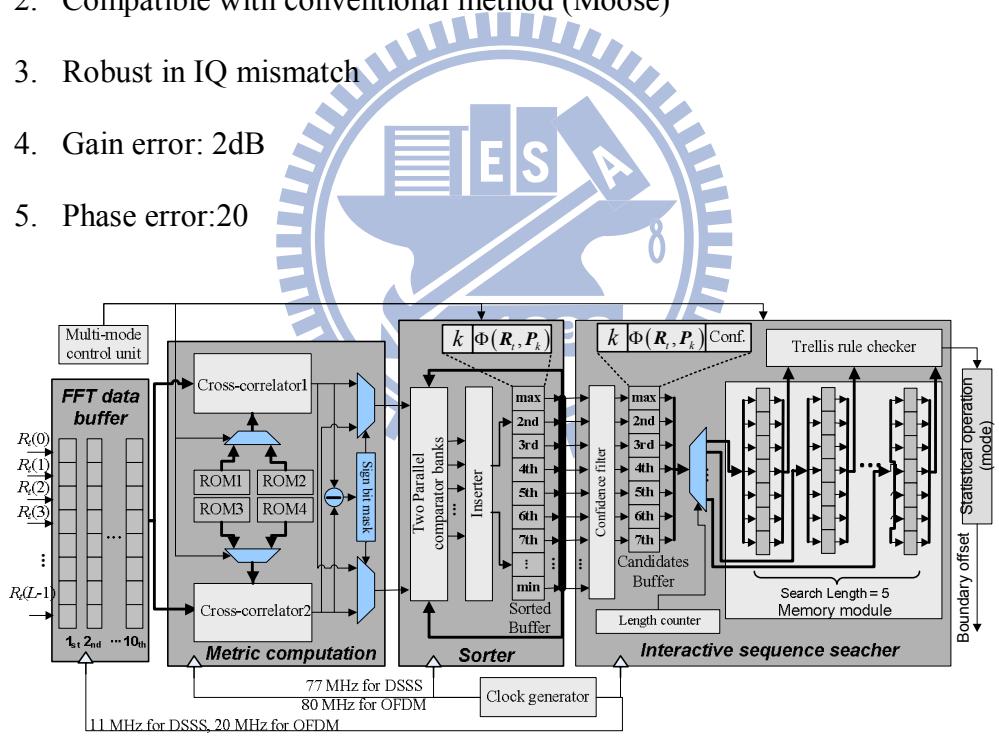


Fig. 7.7 Architecture of FD boundary detector.

### 7.2.2 Medium Access Control (MAC)

MAC protocols play a very important role in wireless node-to-node communication, such as that between base stations and mobile terminals. This work concentrates on quick prototyping, early-stage verification and extensible design of

multi-mode MAC layer systems. Starting from the integrated system of WiMAX/Wi-Fi dual-mode MAC, Object-Oriented Analysis and Design (OOA&D) principle are applied on both protocols, identifying the common and different components between both systems. By using divide-and-conquer and bottom-up design approaches, WiMAX and WiFi MAC are able to be integrated and facilitated reuse and performance optimization of common components between the two systems.

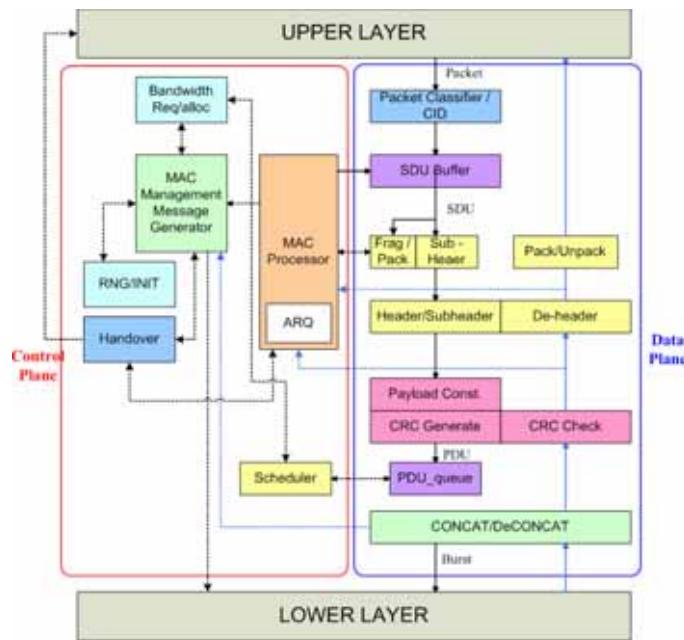
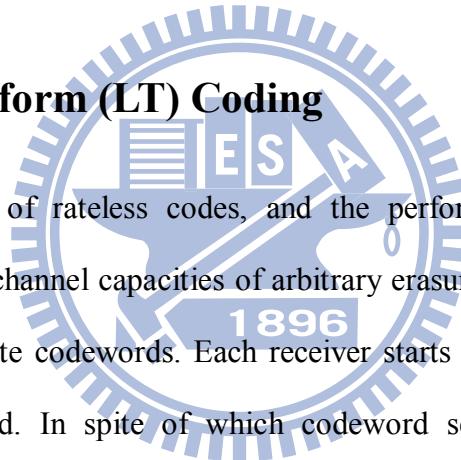


Fig. 7.8 MAC Layer Architecture.

As shown in Fig. 7.8, the MAC protocol layer, in terms of implementation, could be separated in two parts: the Data Plane and the Control Plane. The main function of the Data Plane is production of MAC layer's protocol data units (PDUs). It could either be analyzed with electronic system level (ESL) methodologies, or realized by FPGA hardware solutions. The Control Plane takes control of the Data Plane according to various signal feedbacks. These feedbacks include PHY-to-MAC, Network-to-MAC and inter-BS (base station) or BS-to-MS (mobile station) signaling. In addition to data processing performance that directly relates to software/hardware

co-design, there are other factors that have great impact or overall system performance. For example, the Request/Grant mechanism – the content of MS request shall be properly received and recognized by BS, and then properly responded, vice versa. Some MAC transmission mechanisms including auto retransmission request (ARQ), handover, uplink scheduling, external environmental mechanisms such as BS-end or MS-end channel condition, could deeply influence system performance. Unfortunately, it is difficult to analyze and verify the interaction of MAC functional interactions. The inter-node concepts cover a range even broader than system-level design flows, and traditionally the verification of Control Plane begins at a later stage of design flow.

### 7.2.3 Luby-Transform (LT) Coding



LT code is a class of rateless codes, and the performance of LT coding is approximately close to channel capacities of arbitrary erasure channels. In theory, LT encoder generates infinite codewords. Each receiver starts decoding when sufficient codewords are collected. In spite of which codeword set is collected, the high recovery probability of source symbols is guaranteed. Consequently LT codes are channel independent and require no retransmission. For block codes, when there are too many codewords erased within a block, codewords in this block are undecodable and retransmission is needed. However, retransmission can jam the transmission and paralyze multicasting servers in multicasting. In comparison with block codes, LT codes are more suitable for multicasting. Recently, pre-codes concatenated with LT codes are standardized in 3GPP MBMS (3rd Generation Partnership Project, Multimedia Broadcast/Multicast Service).

LT codes conduct BP (belief propagation) algorithm as decoding scheme. The

advantage of BP decoding is its low decoding complexity. It trades decoding ability for decoding complexity. The performance of LT codes are determined by two factors. One is the degree distributions derived based on BP algorithm. The other is the number of source symbols  $K$ . Theoretically,  $K$  approaches infinity and an LT encoder generates infinite codewords. In practice, with the same degree distribution, the performance of LT codes degrades with the decrement of  $K$ . BP decoding process fails when source symbols are not decoded completely but there are not codewords with degree one left. The information contained in these codewords is unable to be exploited by BP algorithm. This follows that the recovery probability of source symbols is not optimal. Codewords transmitted but not efficiently decoded results in the waste of transmission bandwidth.

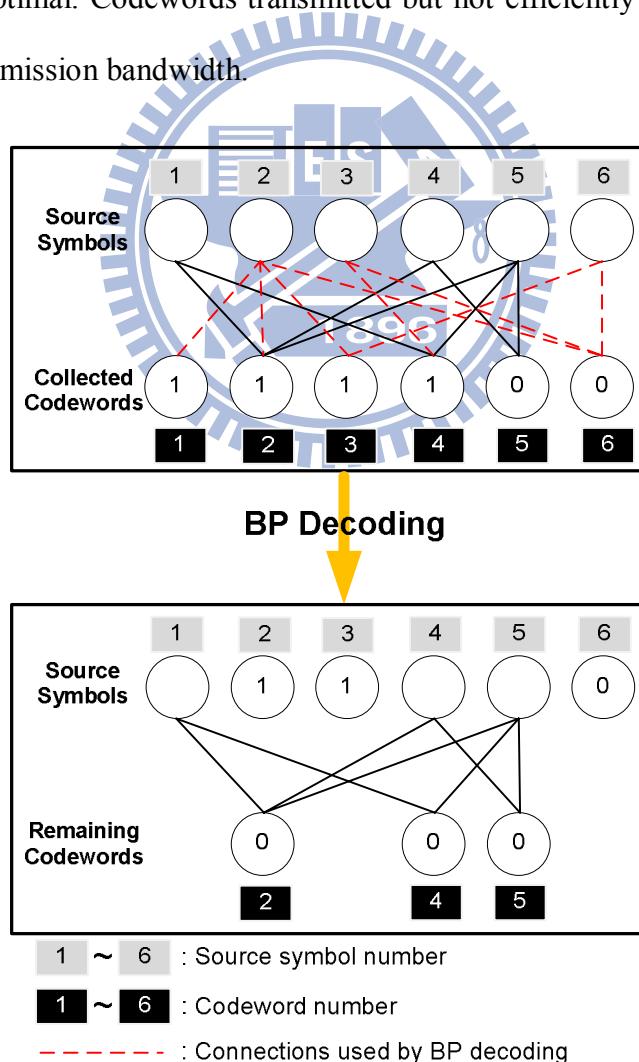


Fig. 7.9 An example of decodable codewords which BP decoding fails to decode.

Fig. 7.9 is a simple example to show this condition. Now, there are six source symbols and six codewords. The red dash line stands for the connections that can be exploited by BP decoding. After BP decoding, codeword 2, 4, and 5 are left. Notice that, the source symbol 1 can be recovered by performing exclusive-or on codeword 2 and codeword 5. Similarly, source symbol 4 can be recovered by performing exclusive-or on codeword 2 and codeword 4. Finally, source symbol 5 is recovered by performing exclusive-or on codeword 2, codeword 4, and codeword 5. For rateless codes, decoding complexity is proportional to the total number of codeword degrees. After BP decoding, most of the codewords are removed. Besides, the average degree of remaining codewords is decreased. For example, with  $K=1000$  and  $N=1120$ , the average degree of the received codewords is 43.6. After BP decoding, the average degree of remaining codewords is 8.3 and the corresponding degree distribution is shown in Fig. 7.9. In addition, the average number of remaining codewords is 85.9. The total number of codeword degrees are  $(43.6 \times 1120) / (8.3 \times 85.9) = 68.5$  times less after BP decoding. It is efficient to conduct more complicated decoding methods to recover the information in the remaining codewords.

#### 7.2.4 Scalable Video Coding (SVC)

Recently, with the prosperity of the Internet video, digital television, and portable devices, the demand of digital video becomes more and more diversified. To deal with those diversified video applications, Scalable Video Coding, the latest video coding standard inherited from the state-of-art H.264/AVC, is formed to provide different scalabilities (temporal, spatial, and quality) in a single bit-stream. Fig. 7.10 shows the SVC encoder architecture with two spatial layers. To generate scalable bitstream, the input images are first downsampled to lower spatial resolution and encoded by H.264/AVC compatible video encoder. Afterward, the higher spatial resolution images

are encoded by H.264/AVC encoder with additional advanced inter-layer prediction techniques to fully utilize the relationship between two consecutive spatial layers and consequently improve the coding performance. In addition, the quality and temporal scalabilities are achieved in each spatial layer by the approaches of Coarse Granular Scalability (CGS) and Hierarchical B structure, respectively. Finally, all generated bitstreams corresponding to different quality scalabilities are grouped into a single SVC bitstream. However, in addition to the primitive coding complexities of H.264, the extra scalabilities of SVC also contribute significant computational complexity and memory requirement in hardware realization. Therefore, in order to minimize the computational complexity and memory requirement for realizing SVC codec, this project first analyzes the internal memory requirement and external memory access to find out the best coding method which can achieve best tradeoff between internal memory usages and external memory accesses and several efficient techniques are also proposed to improve the coding performance of SVC codec.

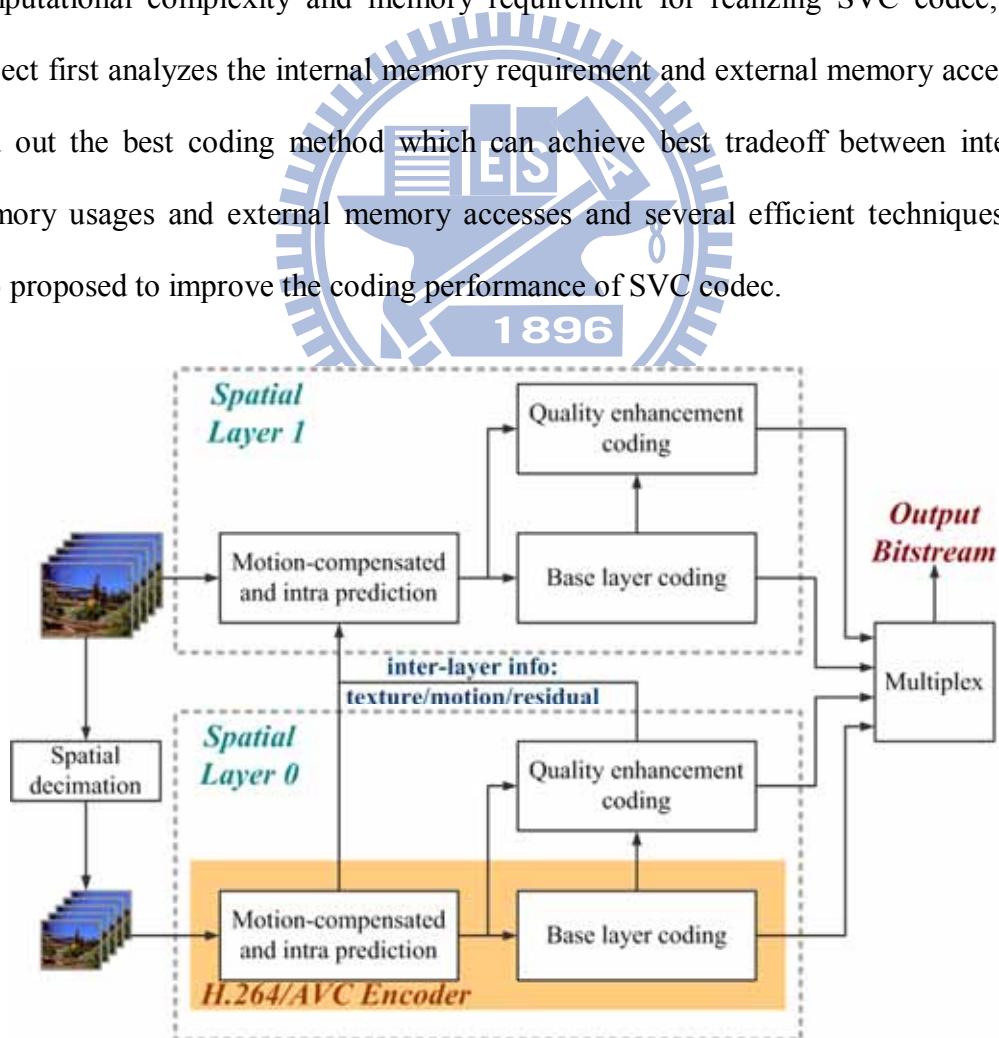


Fig. 7.10 Architecture of an SVC encoder.

### 7.3 Architecture of On-Demand Memory Sub-System

The wireless video entertainment system requires real-time and huge memory accesses. However, the performance gap between the memory and PEs is large in a multi-core SoC platform. A well-organized memory management can significantly reduce the memory access latency. Accordingly, in order to provide high memory bandwidth and capacity for the wireless video entertainment system, a memory-centric on-chip data communication platform (eH-II) is proposed and the overall architecture is as shown in Fig. 1.5, including a memory-centric on-chip interconnection network (OCIN) and an on-demand memory system. Heterogeneous PEs, such as microprocessors, application-specific stream processors and IP (intellectual property) cores, can be integrated in this platform. In eH-II platform, each PE owns a private memory management unit (p-MMU). The p-MMU includes local caches (D-cache and I-cache) and a cache controller which can efficiently handle all memory requests generated by the PEs. It can dynamically allocate unused space in cache for buffering the transmitting data. If the PE requires extra memory resources, the centralized memory resources, including centralized cache and off-chip DRAM, can be used and controlled by a centralized memory management unit (c-MMU). The c-MMU can dynamically allocate and manage the memory resources according to different memory requirements. For the data communication between PEs, message-passing technique is applied for this platform. The PEs transmit/receive the data to/from others through the memory-centric OCIN. Therefore, Network interfaces (NIs) are designed to packetize the transmitted data.

In this heterogeneous multi-core platform, different PEs would have quite different memory requirements with different specific functions. For instance, the memory requirement of the video decoding is larger than that of the WPU. Moreover, different

system environment factors may affect memory utilizations for the applications in platform during runtime. In view of this, different qualities of wireless channels may have different memory behavior in the wireless video entertainment system. Thus, a multilevel memory hierarchy is adopted for the on-demand memory sub-system in eH-II platform. Hence, the memory sub-system enables PEs to dismember the centralized memory resources dynamically.

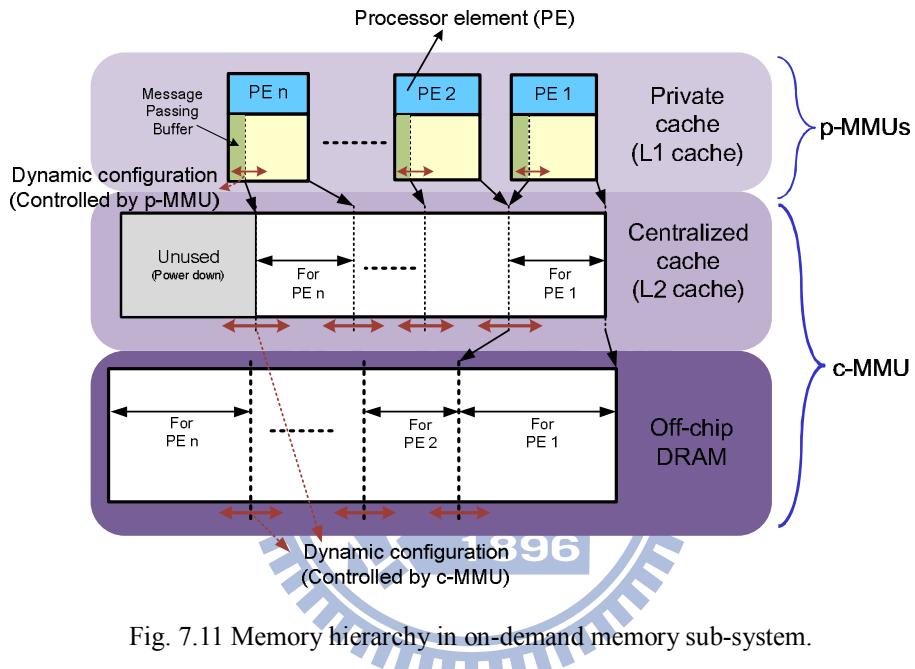


Fig. 7.11 Memory hierarchy in on-demand memory sub-system.

In the on-demand memory sub-system, a three-level memory hierarchy is constructed as shown in Fig. 7.11. For the first hierarchy level, p-MMUs are applied to control the memory accesses. Furthermore, in order to improve the network efficiency of the OCIN, p-MMUs can dynamically allocate unused space in distributed caches to store blocked packets. For the second level hierarchy of the on-demand memory sub-system, a c-MMU is constructed by a cache controller and centralized cache to provide more memory resources for PEs. Consequently, the configuration of the centralized cache can be dynamically adjusted according to the different memory requirement from PEs. Adaptive cache controller controls the adaptive allocation and cache operation in the c-MMU. Furthermore, unused

memories can be power down to reduce power consumptions via the Adaptive cache controller. For supporting enough memory spaces, off-chip DRAM is applied as the third memory hierarchy level in the memory sub-system that is managed by the DRAM controller in the c-MMU. This DRAM controller includes an external memory interface and an address translator to improve the memory access efficiency.

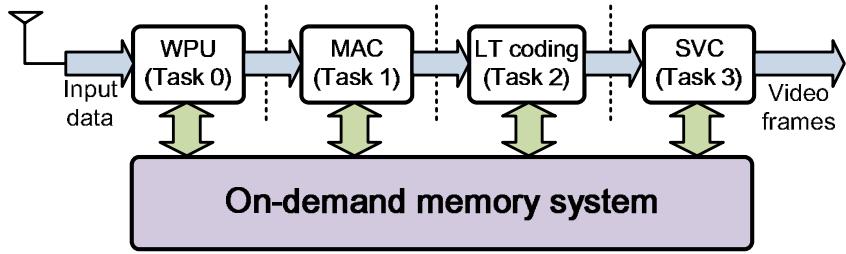


Fig. 7.12 The data stream of wireless video entertainment systems.

According to the receiver of the wireless video entertainment system, the processing sequence of the multiple tasks is generally step by step. Hence, the data stream of this wireless video entertainment system is as shown in Fig. 7.12. In eH-II platform, on-demand memory sub-system can support heterogeneous and real-time memory requirements for wireless video entertainment systems. Based on different memory requirements of PEs, the c-MMU can dynamically allocate memory resources to enhance the flow rate of the data stream. Therefore, with the adaptive memory resource arrangement for different PEs, the execution efficiency of the streaming processing in wireless video entertainment systems can be improved.

The overall architecture of the on-demand sub-system in eH-II platform is as shown in Fig. 7.13. The system components can be categorized into data computation, data communication and data storage. For data computation, WPU, MAC, LT coding and SVC are PEs in the wireless video entertainment system. Wrappers are utilized as core interfaces to satisfy the specification of the pre-defined protocol. Subsequently, the memory-centric OCIN are developed for data communication that includes

network interfaces (NIs) and an interconnection network. In eH-II platform, message-passing mechanism is adopted. With this mechanism, the transmitting data are packetized into packets by NIs, and through the interconnection network using a pre-defined message-passing protocol.

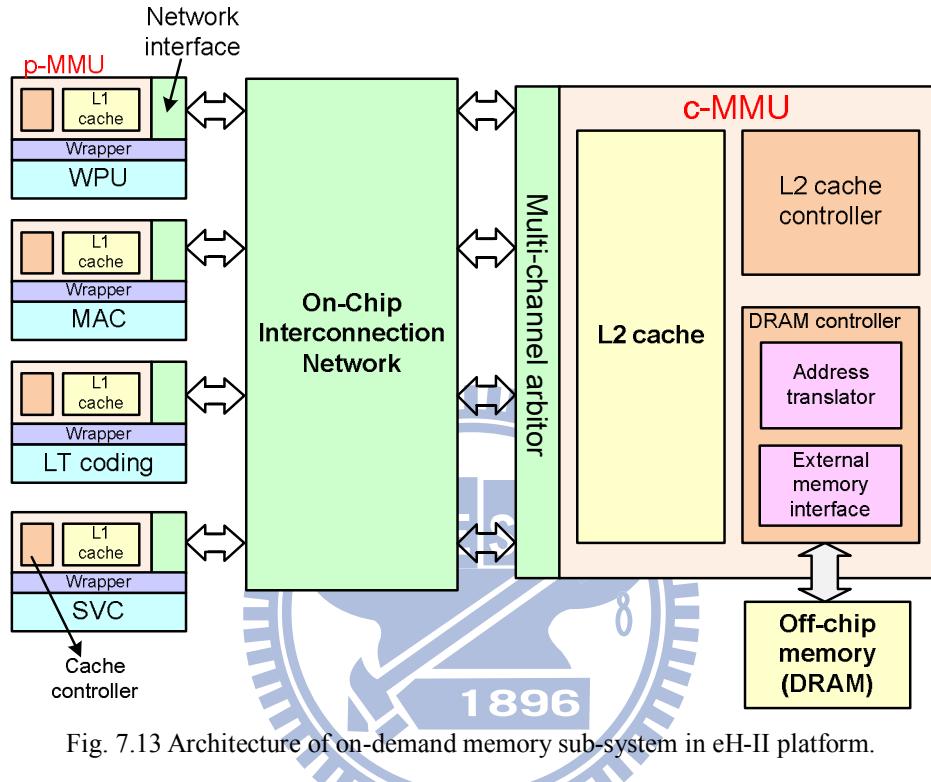


Fig. 7.13 Architecture of on-demand memory sub-system in eH-II platform.

For data storage, each PE owns a p-MMU, which is constructed a distributed cache (L1 cache) and a cache controller for accessing memory and managing the cache usage. When the packet queue in NI is insufficient, the p-MMU can borrow unused cache blocks for NI. Moreover, the c-MMU, consisting of a centralized cache (L2 cache), a cache controller and a DRAM controller, is designed to provide more memory resources for PEs. The cache controller can support dynamical cache re-organization for allocating different cache resources. Consequently, the DRAM controller is designed to efficiently access off-chip DRAM. In the DRAM controller, the address translator rearranges and translates address to have efficient memory allocation, and the external memory interface reschedules the commands to reduce

memory access latency.

## 7.4 Private Memory Management Unit (p-MMU)

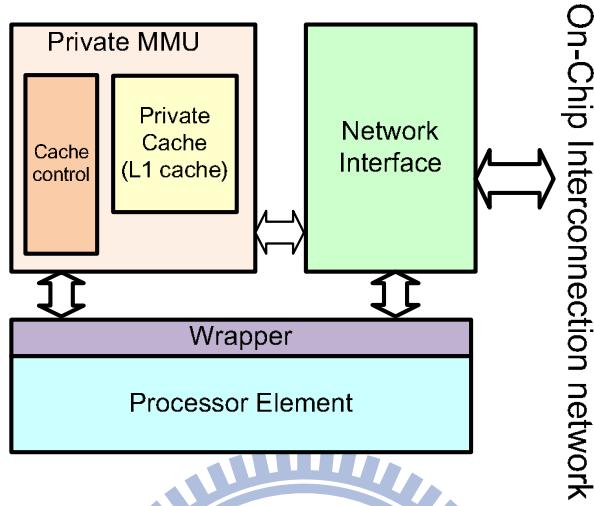


Fig. 7.14 Block diagram of a local node.

In the eH-II platform, a p-MMU is designed in a local node with a NI, a wrapper and a PE. The block diagram in a local node is as shown in Fig. 7.14. To provide local memory resources for PEs, efficient p-MMUs are developed to process the memory requests and to store the temporal data of the tasks. Distributed cache performs as a high level cache for the dedicated PE in the on-demand memory sub-system. In this on-demand memory sub-system, the burst-based memory access protocol is adopted for PEs to access memory. Therefore, distributed cache (L1 cache) and a cache controller are kernel blocks in a p-MMU. Additionally, NI is designed as a bridge between PEs and OCIN. NI contains the input queue and output queue for buffering packets. However, the sizes of the queues dominate the area and the performance. If the buffer is insufficient, the PE will be stall until the head-of-line blocking releases. Therefore, if the utilization of the distributed memory is low, the p-MMU can borrow the memory resources for buffering the blocking packets from the PEs, and the PEs can execute their tasks without any stalls. Restated, when the packet buffer in NI is

crowded, unused cache blocks can be borrowed for buffering the blocking packets from PEs.

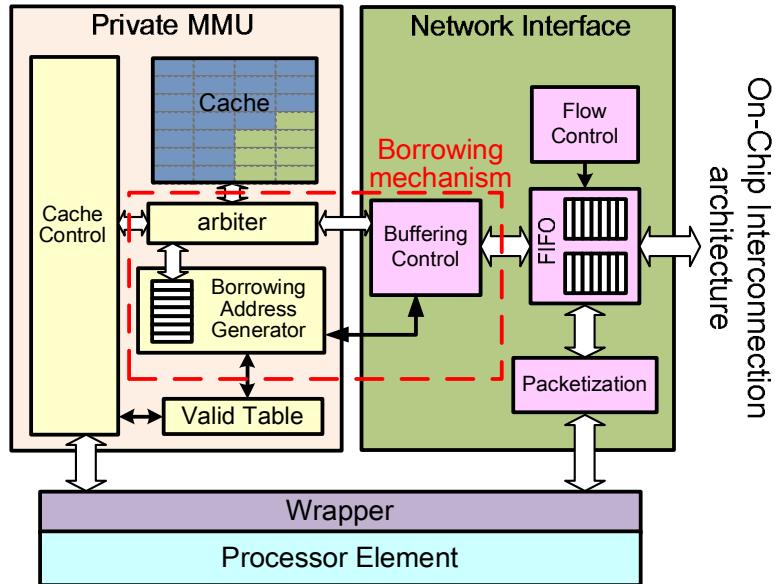


Fig. 7.15 p-MMU and efficient network interface.

### 7.4.1 Buffer Borrowing Mechanism

The architecture of proposed p-MMU and efficient Network Interface with buffer borrowing mechanism is shown in Fig. 7.15. The NI uses a buffering control to generate a borrowing request to the p-MMU for borrowing memory resources. And thus, the p-MMU checks the valid table and generates the borrowing address for the NI. Fig. 7.16 presents the buffer borrowing interface between the NI and p-MMU. The operations of the buffer borrowing include *write*, *read* and *release*. For the write operation, the buffering control should send a buffer request to the p-MMU first, and send the blocking data until receiving a grant signal. However, the head-of-line blocking may release while waiting the grant from p-MMU or setting the data. Therefore, a *release* operation can release the extension memory resources. The details of the borrowing address generator and buffering control will be described as follows.

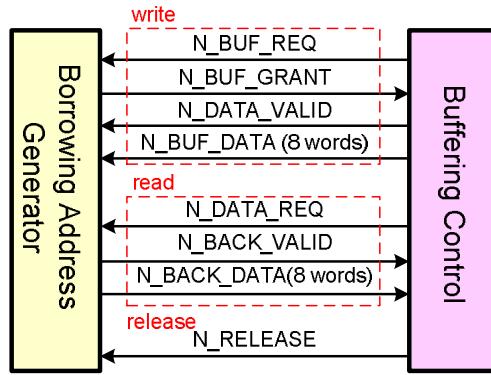


Fig. 7.16 Buffer borrowing interface between NI and p-MMU

### 7.4.2 Borrowing Address Generator

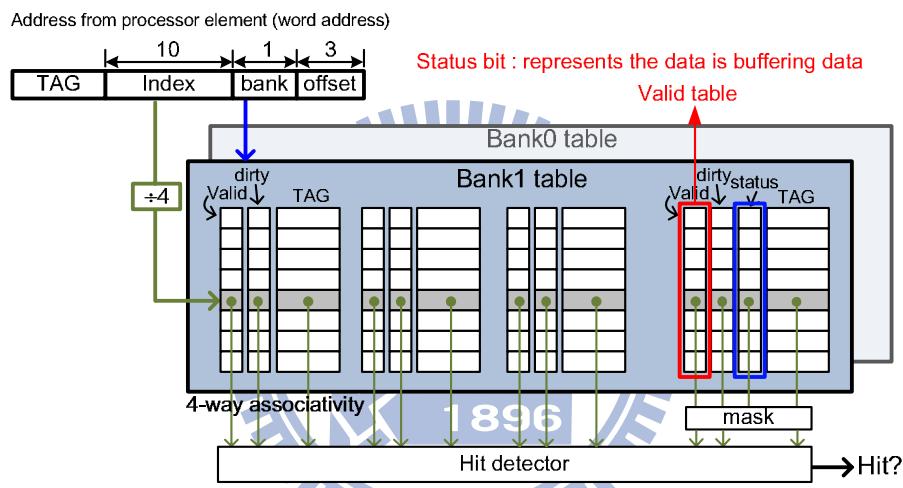


Fig. 7.17 Borrowing mechanism in p-MMU.

When the NI requests an extend buffer to store the blocking packet, the borrowing address generator searches an empty space in the distributed memory via checking the valid table. This valid table is attached in the cache tables as shown in Fig. 7.17. The distributed memories are divided into two banks with four-way association. The memories corresponding to the last associated table in bank 0 and bank 1 are infrequently used in opposition to others. Therefore, the p-MMU can borrow the empty spaces corresponding to this table. Moreover, each cache line in the four-way association contains 4x8 words. Therefore, the maximum payload of a packet can be stored in a memory block (8 words) in one cycle. If a memory block is borrowed, the

p-MMU asserts the status bit that represents the borrowing data. Depending on the status bit, the cache control can mask the searching of this table in a searching operation.

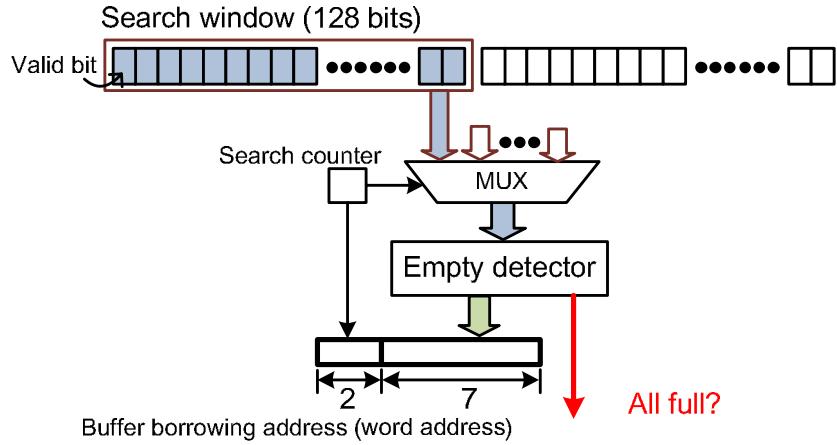


Fig. 7.18 Architecture of the empty memory block searching.

After the NI send a borrowing request to the p-MMU, the NI should take 2-8 cycles for collecting the payload. Most packets contain 8 flits in their payloads, and the average size of payload is about 4 words. Therefore, the p-MMU has to search the empty memory block in 4 cycles. Additionally, the last associated tables in bank 0 and bank 1 contains 512 valid bits. To search the empty memory block, a 128-bit searching window is adopted. Fig. 7.18 shows the architecture of the empty memory block searching. The searching window is controlled by a search counter. The empty detector detects an empty memory block and generates the borrow address with the search counter. If all memory blocks in a searching window are full, the searching windows will move to the next 128 bits. Fig. 7.19 shows the searching flow chart of the borrowing mechanism. The flow can be divided into three steps, which are empty memory block searching, borrowing status setting, and data writing. The operations of empty memory block searching and borrowing status setting are described above. While writing data in the borrowing memory block, the borrowing address should be stored in the address queue for reading operations. After writing the payload into the

memory block, the grant signal is changed to 0 for the next borrowing request.

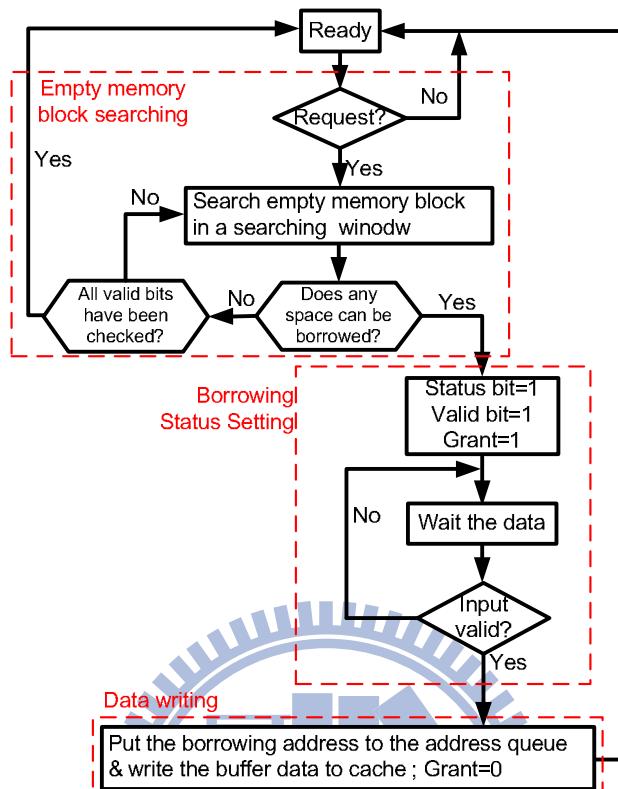


Fig. 7.19 Searching flow chart of the borrowing mechanism in p-MMU.

### 7.4.3 Buffering Control

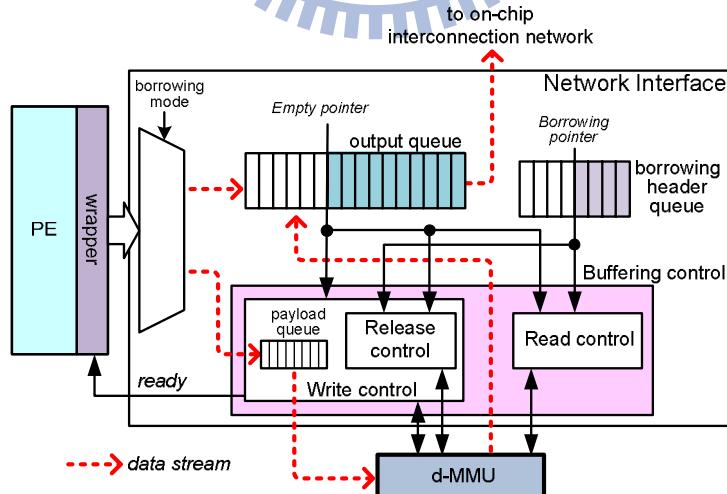


Fig. 7.20 Block diagrams of borrowing mechanism in network interface.

The buffering control in NI detects the empty size of the output queue and sends the borrowing operations to p-MMU. Fig. 7.20 shows the block diagrams of

borrowing mechanism in the buffering control. The buffering control sends the write, read, and release operations depending on an empty pointer of the output queue and a borrowing pointer of the borrowing header queue. The empty pointer and borrowing pointer indicate the number of the occupied buffers in the output queue and borrowing header queue, respectively. In addition, the write control contains a payload queue for collecting the payload, and then writing this payload to the borrowed memory block. The borrowing control policy of the buffering control is presented as shown in Fig. 7.21. The borrowing mode indicates whether the blocking data stored in the p-MMU or not. Therefore, after receiving data from the PE, the data should be stored in the p-MMU in the borrowing mode. Otherwise, the data can be stored in the output queue when the size of the empty slots is larger than the payload. While waiting the borrowing grant from p-MMU and collecting the payload, the head-of-line blocking may be released. Therefore, the borrowing mechanism can also be released if the borrowing mode equals to zero. The release signal will interrupt the search operation of p-MMU.

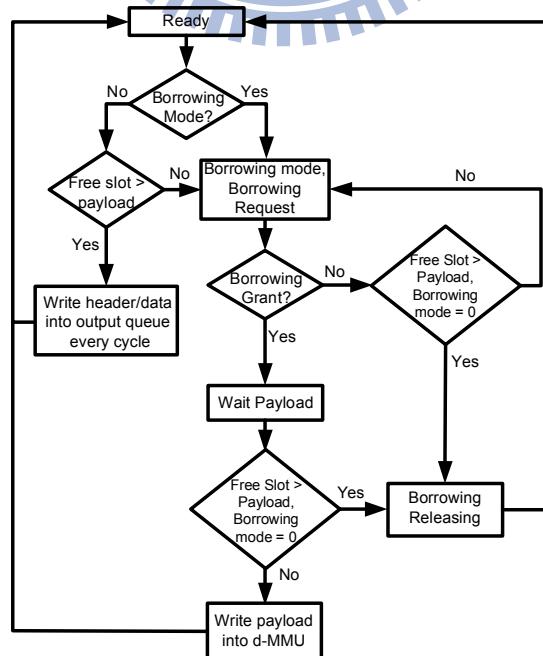


Fig. 7.21 Borrowing control policy of the buffering control.

#### 7.4.4 Simulation Results of Buffer Borrowing Mechanism

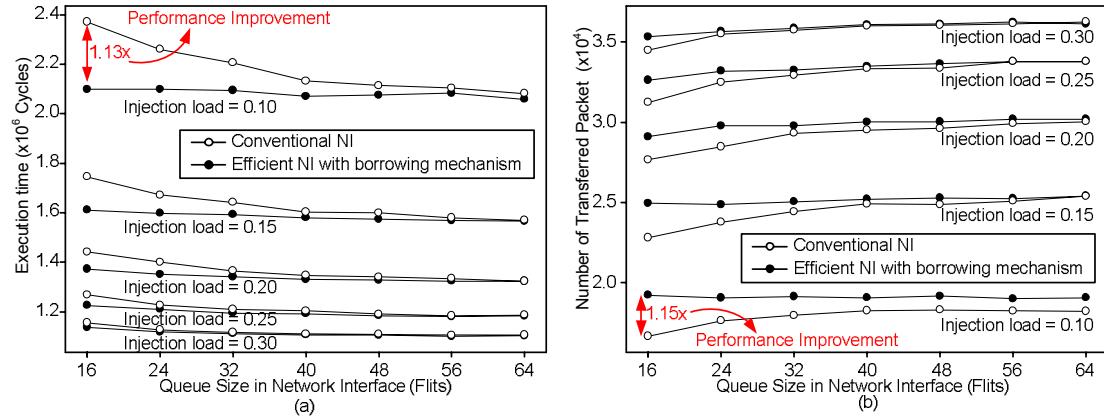


Fig. 7.22 (a) Execution time under various injection loads and queue sizes (b) Transferred packets under various injection loads and queue sizes.

The proposed p-MMU, NI and memory-centric OCIN are implemented in SystemC for the cycle-driven simulation. Thereby, the simulation environment is set as a 4x4 router with 4 PEs to evaluate the performance improvement via the efficient NIs. Fig. 7.22(a) shows the execution time of transferring 200000 packets under various injection loads and queue sizes. With the increasing injection load, the execution time decreases because the transferred packets are fixed. Additionally, Fig. 7.22(b) shows the number of transferred packets in 300000 cycles under various injection loads and queue sizes. Based on the simulation results, the proposed borrowing mechanism can achieve the similar performance with different queue sizes. Moreover, the proposed efficient NI can realize about 1.15x performance improvement compared to the conventional one with 16flits.

#### 7.5 Centralized Memory Management Unit (c-MMU)

The distributed memory resources may be insufficient for PEs. Therefore, lower level cache is utilized to provide larger on-chip memory resources in a c-MMU. According to distinct memory resource requirements from different PEs, the proposed c-MMU can allocate different cache resources for PEs via an adaptive cache

controller. In addition, the external memory is required for storing the huge data such as video frames in video processing. A DRAM controller is designed in the c-MMU to access DRAM device.

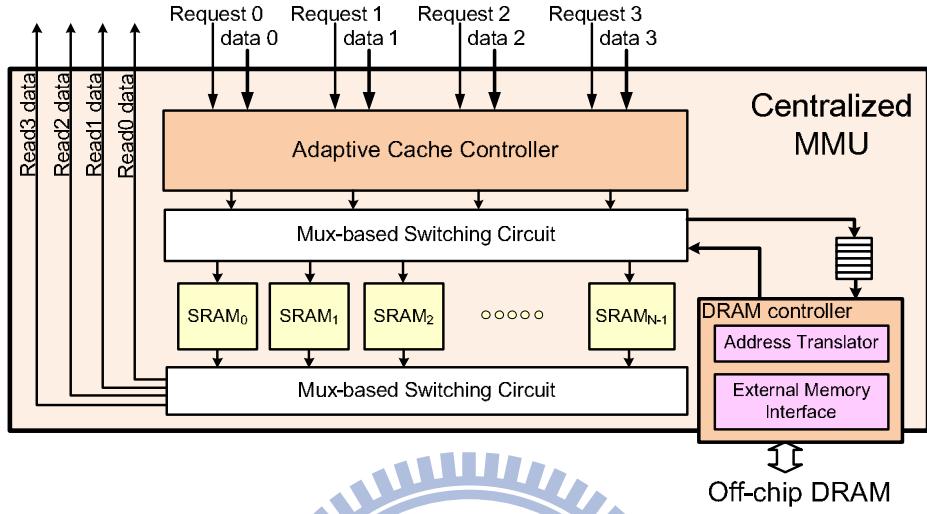


Fig. 7.23 Block diagram of the c-MMU.

The block diagram of the c-MMU is shown in Fig. 7.23, consisting of an adaptive cache controller, MUX-based switches, SRAM sub-blocks and a DRAM controller. The adaptive cache controller receives the memory requests from different p-MMUs which can be executed simultaneously. The proposed adaptive cache controller checks the selected cache tables to determine whether the data is in the cache or not. According to the checking result, the corresponding data and addresses are forwarded to the SRAM sub-block or DRAM controller by the MUX-based switches. For the read requests, the read data forward to the output switch and send back to p-MMUs. Consequently, the DRAM controller, consisting of the address translator and external memory interface, is designed to efficiently access the external memory. The address translator is applied to re-generate DRAM address for improving memory bandwidth efficiency and reducing the DRAM energy consumption. The design strategy strongly depends on the memory access behavior of the applications. The detail design of the address translator will be described in Section 7.6.

### 7.5.1 Adaptive Cache Controller

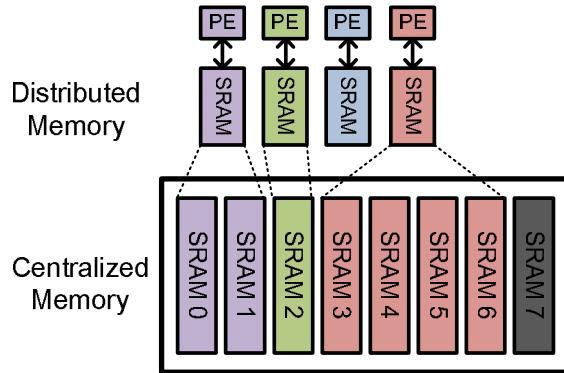


Fig. 7.24 oncept of the adaptive memory resource allocation.

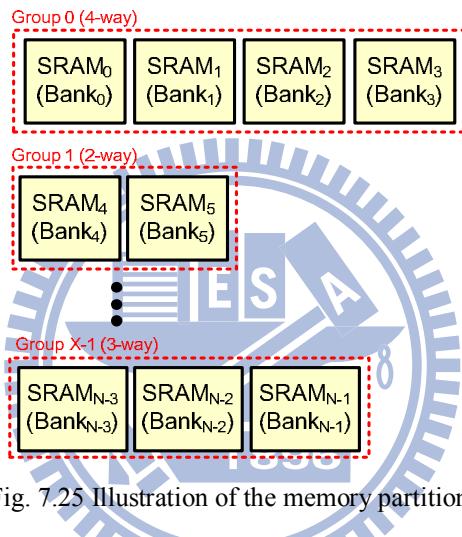


Fig. 7.25 Illustration of the memory partition.

The proposed c-MMU can dynamically adjust and allocate memory resources for PEs during the runtime. The concept of the adaptive memory resource allocation is as shown in Fig. 7.24. Base on different memory requirement of PEs, unequal memory resources are allocated. The principle of adjusting the cache size is base on selective cache ways which had been proposed in [7.16]. Based on the selection of different ways, the cache size can be assigned for PEs dynamically associated with small area and timing overhead for the cache reconfiguration. In the proposed c-MMU organization, the associativity-based partitioning scheme is adopted for the cache partition. Each SRAM sub-block represents a *way* and forms a bank for the cache organization. Therefore,  $N$  SRAM sub-blocks in the c-MMU represent  $N$ -way

associativity capacity in the centralized cache. For different PEs, the SRAM sub-blocks can be grouped into several groups for PEs. Fig. 7.25 shows an example of SRAM bank partition. Assume the c-MMU has  $N$  SRAM banks for  $X$  PEs. The memory partition can be realized as illustrated in Fig. 7.25.

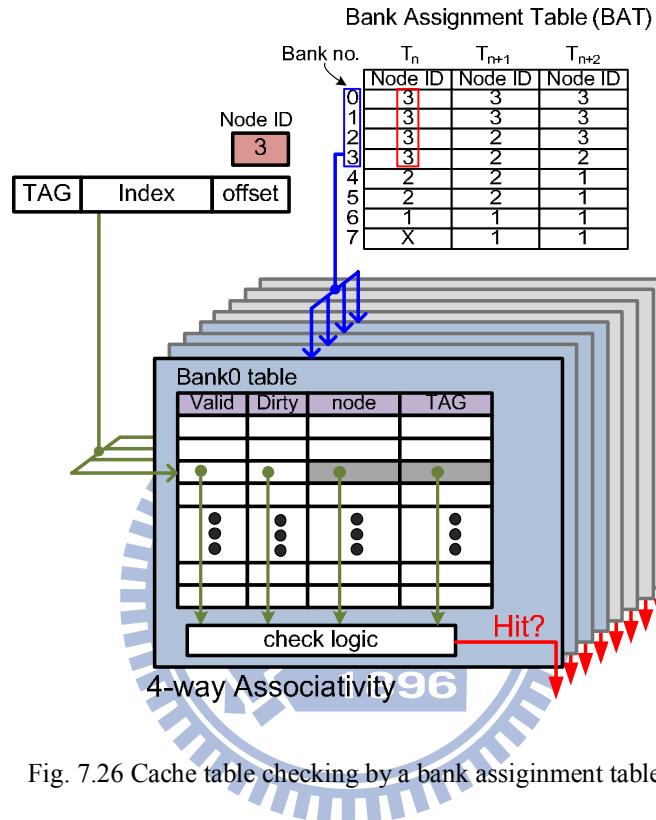


Fig. 7.26 Cache table checking by a bank assignment table.

In order to dynamically allocate the memory resources for PEs during runtime, a bank assignment table (BAT) is designed to record the memory usage information in three time intervals. Fig. 7.26 illustrates the cache table checking method using the BAT. According to the corresponding ID of the PE, the adaptive cache controller searches the BAT and returns the assigned bank numbers indicating the associated bank tables. In Fig. 7.26, four memory banks are assigned for PE3 in the first time interval. When a request from PE3 is served, Bank0, Bank1, Bank2 and Bank3 tables will be selected for hit checking. Based on this configuration, a 4-way associativity L2 cache memory resource can be accessed by PEs.

For a multi-task system, multiple memory requests from different PEs can be served simultaneously in the c-MMU because the checking tables are independent for different PEs generally. Fig. 7.27 shows the illustration of checking multiple requests. The associated bank tables are selected in accordance with BAT information, and the check functions are operated independently. Additionally, the BAT can recode the configuration in different time interval that is updated by profiling the memory requirements of the systems off-line. For the wireless video entertainment system, the effective bandwidth of the channel can be detected by MAC. According to the detection of the wireless channel, the transmitter can determine the scalable level of SVC bit-stream to satisfy the effective bandwidth. Based on various bit-stream, the memory requirement of different quality levels is also various and can be profiled off-line. In view of these, the BAT can be controlled via the detection of MAC and the profiled memory requirements.

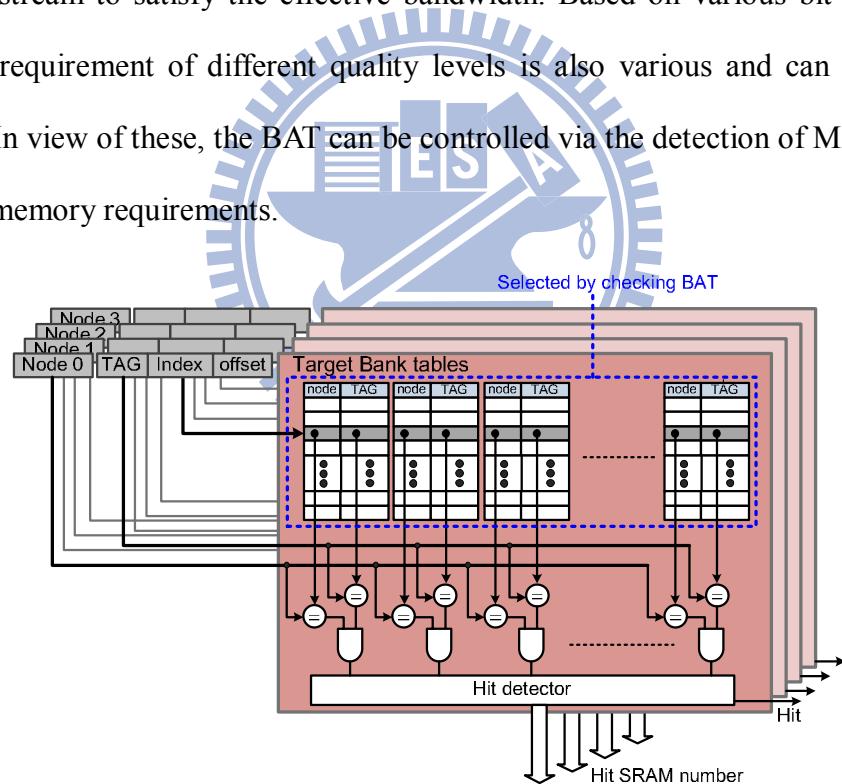


Fig. 7.27 Illustration of checking multiple requests.

While changing time intervals, the bank assignment for each PE is reorganized. The bank configurations may be different from the previous configurations. Therefore, the Lazy-based transitioning scheme [7.17] is utilized for maintaining data consistency. If a miss occurs, the data may remain in the other banks. The bank tables

which are assigned in previous time interval need to be checked again. The flow chart of the adaptive cache control is as shown in Fig. 7.28.

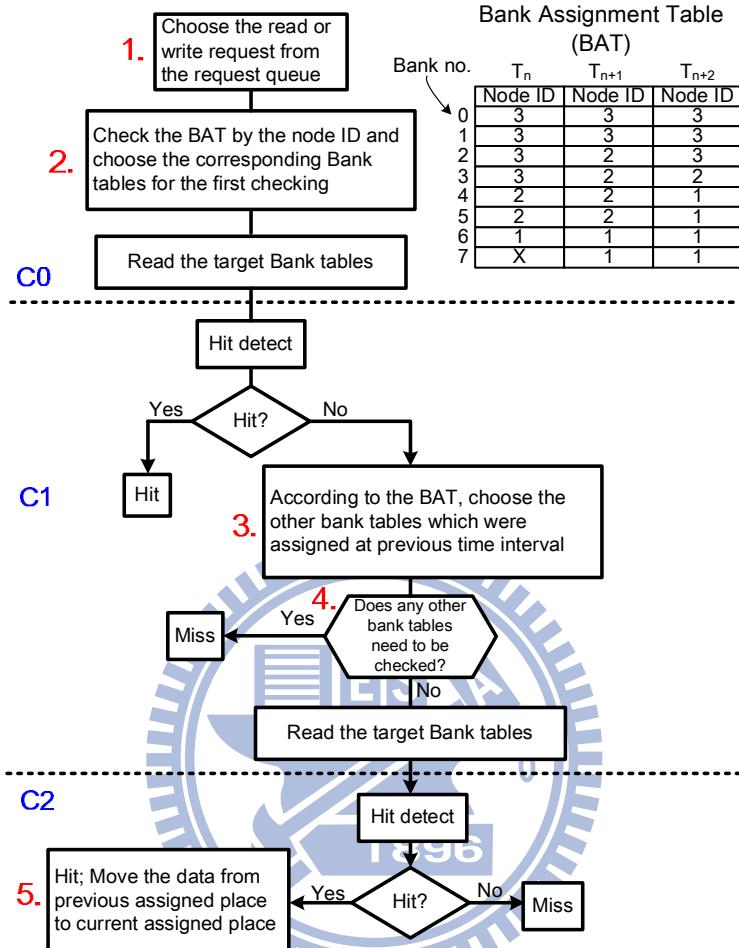


Fig. 7.28 Flow chart of adaptive cache control.

In step 1, the read or write request are received from the request queues. The priority of read request is higher than the write request unless the data dependency is detected or the write queue is full. In step 2, the corresponding tables are selected by the BAT information. If a miss occurs, the other tables would be checked based on the BAT information of the last time intervals in step 3 and step 4 because the corresponding data may still be reserved in the centralized cache. If other tables should be checked according the assignment of the last time intervals, the checking operation will be lunched again. Otherwise, the miss operation will be executed. After finishing the second hit detection, additional data movement will be executed if hit

occur in step 5. The corresponding data will be moved from original location to the new location. Accordingly, these five steps are executed in three cycles as shown in Fig. 7.28. Therefore, the proposed adaptive cache controller has one cycle overhead.

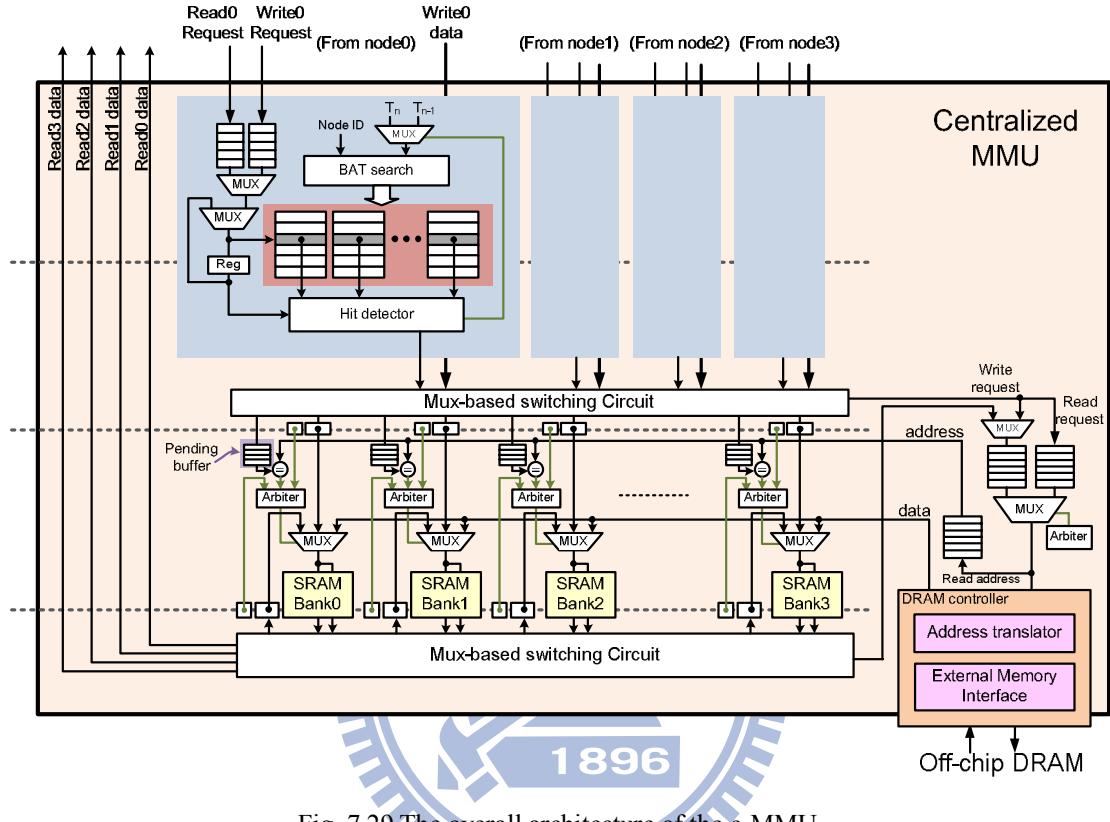


Fig. 7.29 The overall architecture of the c-MMU.

The overall architecture of the c-MMU is divided into four execution stages as shown in Fig. 7.29. The blue blocks indicate the adaptive cache controller, and the tables in red block are the selected bank tables for tag checking. In the first execution stage, the BAT search engine searches the bank assignment information according to the Node ID and selected time interval. In the second execution stage, the hit detector outputs the corresponding read/write address and bank destination to the MUX-based switch. If the data does not exist in the c-MMU for a read request, the address will be transferred to the corresponding bank pending buffer and the external memory read queue. When the read data is read from off-chip DRAM, the data can be directly stored into the corresponding bank and transmitted back to the corresponding

p-MMUs via comparing the address in the pending buffer in third execution stage. Since a bank can be accessed by the cache controller, external memory and other banks, an arbiter at the third execution stage is adopted to determine the priority of access requests for a memory bank. The requests from external memory have the highest priority to minimize the miss penalty, and the priority of additional write requests generated by cache reorganization is set as the lowest one. At the final execution stage, the cache data are read out and forwarded to the PEs, external memory or another SRAM banks by a MUX-based switch. Additionally, for the external memory request arbitration, the read requests are served prior to the write requests.

### 7.5.2 External Memory Interface in the DRAM Controller

The external memory interface (EMI) is developed to communicate with the external memory for the on-demand memory sub-system. To deal with tremendous data transfer and storage of the video processing, the external memory must provide high data bandwidth to guarantee the real time request. However, the bandwidth of the external memory is limited by the finite pin number of I/O. Accordingly, the external memory interface should provide high data bandwidth utilization.

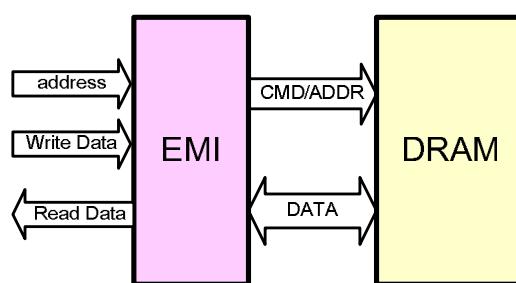


Fig. 7.30 Connection between the external memory interface (EMI) and DRAM.

The EMI receives the physical addresses of DRAM from the address translator, and thus generates DRAM commands to access DRAM data. The connection between

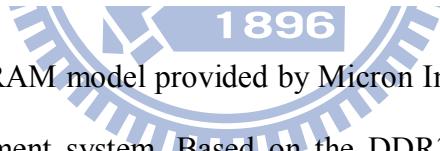
the EMI and DRAM is as show in Fig. 7.30. EMI generates the appropriate commands based on the DRAM specification without any DRAM timing violation. Moreover, a command scheduling is applied to reschedule the DRAM commands to improve the bandwidth efficiency. Because the banks in the DRAM can be operated in parallel, the commands with different banks would enable issued without timing constrain. In view of this, rescheduling DRAM commands can realize higher bandwidth utilization than in-order issuing.

Table 7.3 Micron DDR3 configurations

Speed Grade	Data Rate (MT/s)	Target $t_{RCD}$ - $t_{RP-CL}$	$t_{RCD}$ (ns)	$t_{RP}$ (ns)	$CL$ (ns)
-125 <sup>1, 2</sup>	1600	11-11-11	13.75	13.75	13.75
-125E <sup>1, 2</sup>	1600	10-10-10	12.5	12.5	12.5
-15 <sup>3</sup>	1333	10-10-10	15	15	15
-15E <sup>1</sup>	1333	9-9-9	13.5	13.5	13.5
-187	1066	8-8-8	15	15	15
-187E	1066	7-7-7	13.1	13.1	13.1

Parameter	256 Meg x 4	128 Meg x 8	64 Meg x 16
Configuration	32 Meg x 4 x 8 banks	16 Meg x 8 x 8 banks	8 Meg x 16 x 8 banks
Refresh count	8K	8K	8K
Row addressing	16K (A[13:0])	16K (A[13:0])	8K (A[12:0])
Bank addressing	8 (BA[2:0])	8 (BA[2:0])	8 (BA[2:0])
Column addressing	2K (A[11, 9:0])	1K (A[9:0])	1K (A[9:0])

 1 Giga-bit DDR3 SDRAM model provided by Micron Inc. [7.18] is adopted in the wireless video entertainment system. Based on the DDR3 SDRAM, several speed grades and configurations can be selected as listed in Table 7.3. 15E speed grade and 64Megx16 configuration is chosen for the wireless video entertainment system. There are 8 independent banks in a DDR3 device. The EMI recodes the bank status and generates appropriate commands according to the corresponding bank states. However, different speed grades and configurations have different timing constrain, and the designer should follow these timing rules to build the EMI.

The architecture of EMI is as shown in Fig. 7.31, consisting of three finite state machines, FIFOs, a command scheduler, timing counters and I/O control circuit. The operation of the proposed EMI can be briefly separated into three parts, including

command generating, command issuing and I/O controlling, and each part is controlled by a finite state machine.

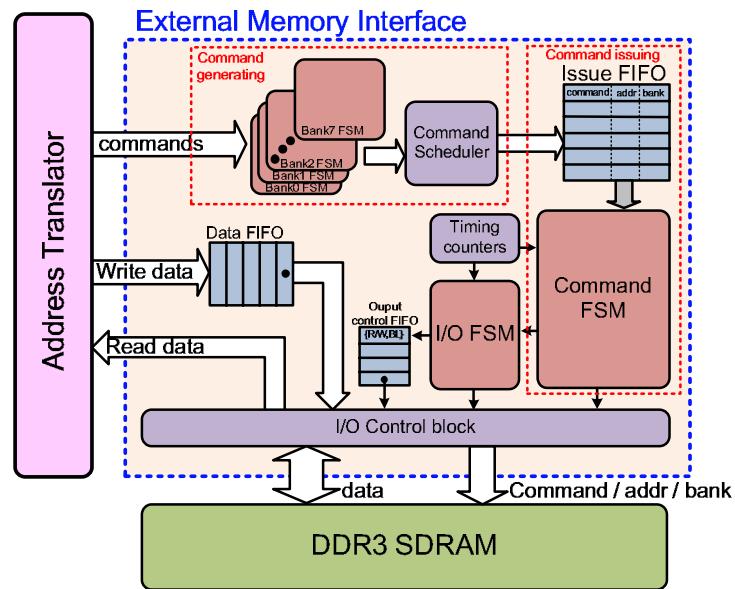


Fig. 7.31 Architecture of the external memory interfaces.

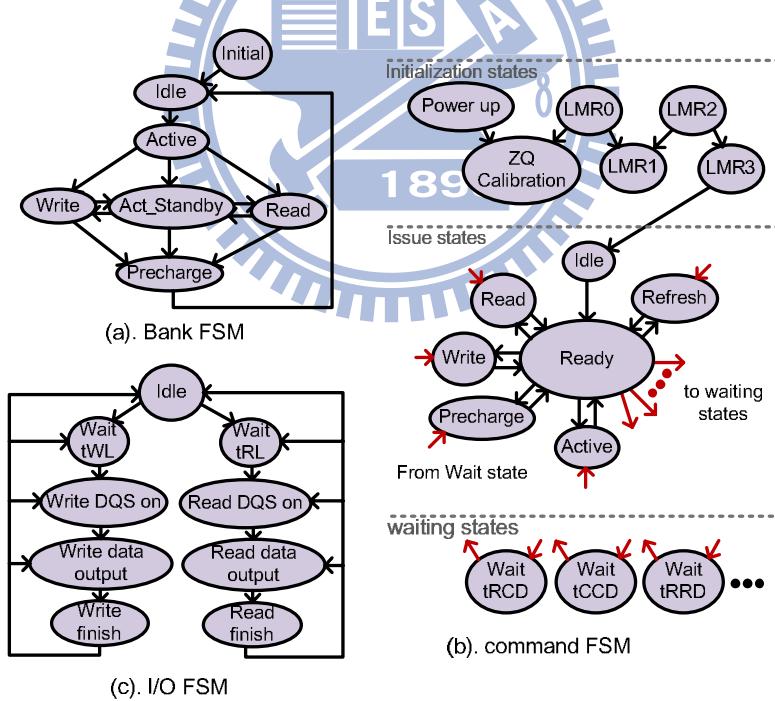


Fig. 7.32 State diagram of EMI Finite State Machines.

In the part of command generating, 8 Bank Finite State Machines (FSM) are constructed to recode the status of eight DRAM internal banks for generating the commands to access DRAM. The state diagram of the Bank FSM is as shown in Fig.

7.32(a). When an input command addresses to one of DRAM banks, the state of the corresponding Bank FSM is checked. According to different bank status, the correct commands are issued to the command scheduler for rescheduling. After rescheduling the command, the DRAM commands are stored in the issue FIFO. When issuing these commands to DRAM device, complex timing rules must be strictly observed. The command FSM can issue the commands in the right time without any timing violations that is controlled by timing counters recoding the cycle margins of different timing constraints. When a command is issued, the relative timing counters are set to the corresponding values and activated. Additionally, when issuing new commands from issue FIFO to DRAM, the timing violation will be checked via the timing counters. If a timing variation exists, the commands will be stalled. During the waiting time, the EMI issues NOP commands to DRAM. Consequently, DRAM requires a long latency for the initialization, including ZQ calibration and mode register loading. Fig. 7.32(b) shows the state diagram of the command FSM, including initialization states, issue states and waiting states. The initialization states handle the DRAM initializations, and the issue states generate the appropriate DRAM commands to the I/O control block. Additional waiting states will stall the command issuing until the following command can be issued legally based on the timing counters.

The command scheduler is utilized to reschedule the command sequence for improving the bandwidth efficiency. Therefore, parallelizing the accesses which address to different banks is realized to fully utilize the DRAM bandwidth. When the successive accesses address to different banks, the bank-miss is induced. Fig. 7.33(a) presnets the original command sequence without any scheduling, and the bandwidth efficiency is the worst. Fortunately, the banks in a DRAM device can be operated in parallel. Therefore, the banks can be activated first, and then the column access

commands are issued as shown in Fig. 7.33(b). The bank activation time can be hidden. However, no more than four bank ACTIVATE commands can be issued in a given tFAW (MIN) period. If the number of successive accesses with different banks exceeds four, the sequence can be optimized by interleaving ACTIVATE and column access commands as shown in Fig. 7.33(c). The proposed command scheduler can schedule the ACTIVATE and column access command with different banks to the optimal sequence.

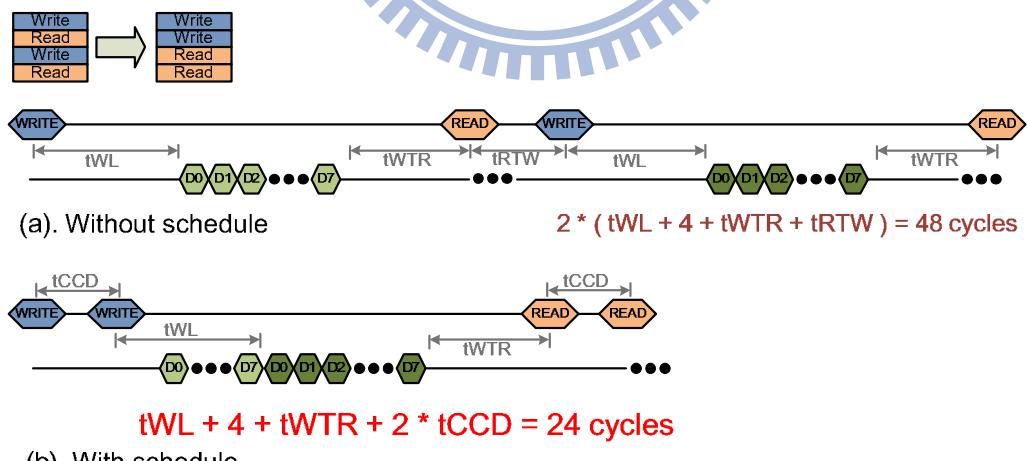
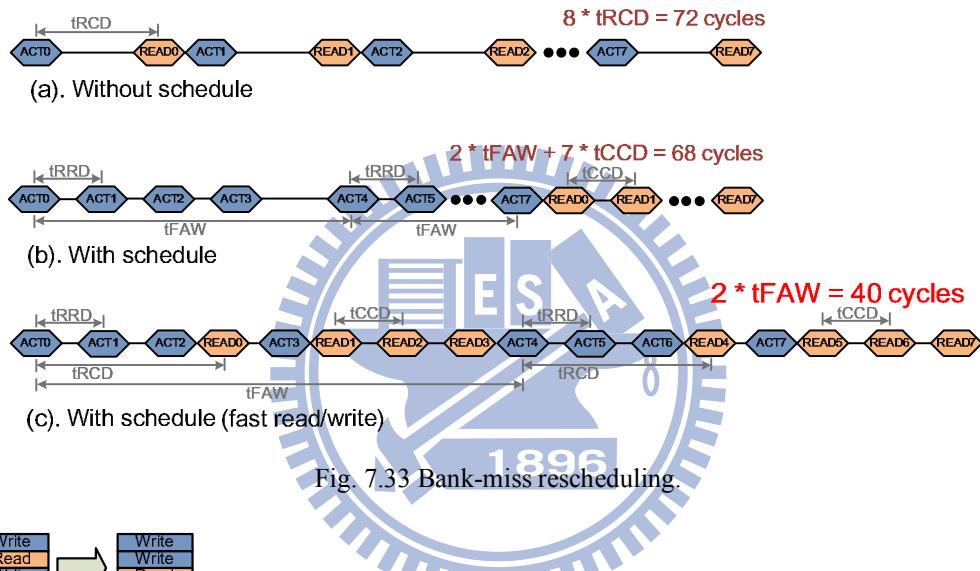
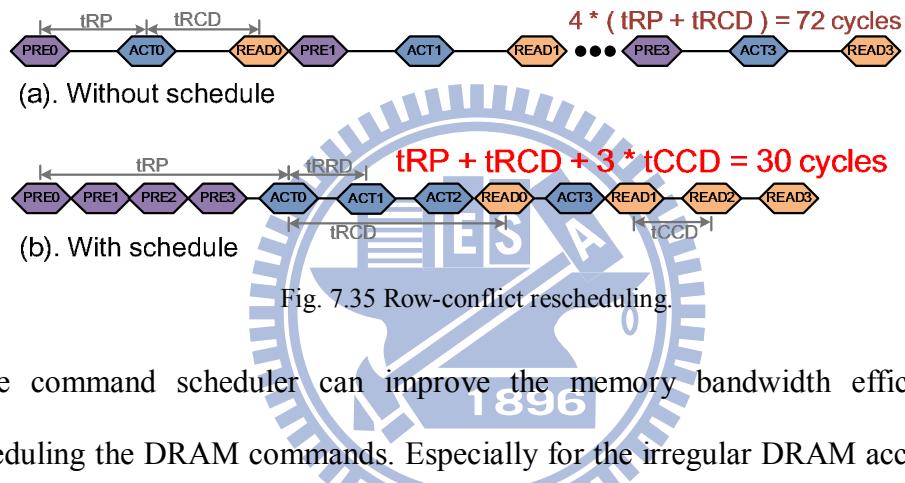


Fig. 7.34 Read/Write rescheduling.

For accessing DDR3 devices, writing a burst data may be followed by a subsequent read command. The bandwidth efficiency is reduced while the read and write commands are interleaved frequently. Fig. 7.34(a) illustrates an example of

issuing the read bursts after write bursts. If no data dependency exist in the successive read and write commands, the issue sequence can be exchanged to improve the bandwidth efficiency as shown in Fig. 7.34(b). Furthermore, when row-conflict occurs, the PRECHARGE and ACTIVATE commands should be issued to deactivate the open row and re-activate new row. Fig. 7.35 shows an example of four successive row-conflict reads for different banks. Based on the row-conflict rescheduling, the PRECHARGE and ACTIVATE commands can be issued in advance so that the precharge and activate time can be hidden.



The command scheduler can improve the memory bandwidth efficiency by rescheduling the DRAM commands. Especially for the irregular DRAM accesses, the command scheduler significantly reduces the average access time by hiding additional cycles which are caused by bank and row conflicts. In order to measure the DRAM memory access efficiency, the bandwidth utilization is defined as Eq. (7.1).

$$\text{Bandwidth Utilization} = \frac{\text{Total cycles of input/output data between DRAM}}{\text{Total cycles of processing access commands}} \times 100\% \quad (7.1)$$

The random access pattern is applied for measuring the DRAM bandwidth utilization with/with rescheduling. Table 7.4 lists the summary of simulation configurations, and the successive access requests are random. Base on the definition of the DRAM bandwidth utilization, the bandwidth utilization can be improved by 1.42x, from 18.8% to 26.53% via the proposed command scheduler.

Table 7.4 Simulation environment.

Test Pattern configuration	
Burst Length	8 bits
Number of random r/w command	2000
EMI configuration	
Clock rate	666.67MHz
Data FIFO depth	32 bits
Issue FIFO depth	32 bits
DRAM configuration	
Channel/Rank/Bank	1/1/8
Reference Model	Micron DDR3-1333 MT41J128M8BY-15E
Operating clock rate	666.67MHz

## 7.6 Pre-fetch Mechanism and Address Translator for SVC

In this section, a pre-fetch mechanism and an address translator are proposed in the on-demand memory sub-system to improve the performance for SVC. The SVC PE requires the largest memory resources than other PEs in the wireless video entertainment system. Thus, a pre-fetch approach for SVC is proposed to improve the memory access performance. The proposed pre-fetch mechanism can pre-fetch the residual data and motion vectors which will probably be read for inter-layer prediction decoding presently. Furthermore, the DRAM data arrangement for SVC data is utilized in the address translator to reduce the DRAM access latency and power consumption. The address translator can increase the probability of row-hit and bank-hit status in the DRAM controller to reduce both the activated power and the pre-charge time. The details of the pre-fetch mechanism and address translator will be described in the following sections.

### 7.6.1 Inter-Layer Pre-Fetch Scheme

For realizing the wireless video entertainment system, power consumption has

been the most critical issue. However, the property of intensive memory data accessing, especially for external memory accesses, contributes the significant part of the power consumption in the system. Therefore, if the external memory accesses can be reduced, both of the power saving and system performance can be improved. To reduce the overhead of external memory accesses, several works have been proposed to reduce the search range for motion estimation [7.20]-[7.23] in the video encoder. For the video decoder, the literatures explored the possibility of data reuse for motion compensation [7.24]-[7.26]. Moreover, the pre-fetch schemes were adopted to early acquire the data which will be used for reconstructing the video frames [7.27], [7.28]. In the conventional single layer video codec structures, such as H.264, the mechanism of pre-fetch schemes is usually utilized on motion estimation and compensation. However, the performance of pre-fetch is usually inefficient for the irregular moving of video content. The video decoder cannot guarantee where the object in video content will move to before decoding the motion vectors and failed to pre-fetch the appropriate data.

In SVC, three scalabilities are supported to achieve spatial, temporal and quality adaptation for satisfying the application diversities. Additionally, three inter-layer prediction modes are utilized to further increase the coding performance. Based on the inter-layer prediction, the motion information, residuals and reconstructed video signals in the base layer are used as the references to predict the current macroblock. The three inter-layer prediction modes in SVC are described as follows, including inter-layer motion prediction, inter-layer residual prediction and inter-layer intra prediction.

In inter-layer motion prediction mode, when the enhancement layer as well as the base layer is inter prediction mode, the motion information of base layer can be used

as reference for prediction in enhancement layer as shown in Fig. 7.36. Therefore, the macroblock partition of the enhancement layer is acquired from the corresponding 8x8 block of the base layer associated with a scaling operation. In addition to the block size, the motion vectors of the enhancement layer are obtained by multiplying the motion vectors of corresponding 8x8 block size in base layer by 2. Furthermore, the up-sampled motion information is used to refine the search results.

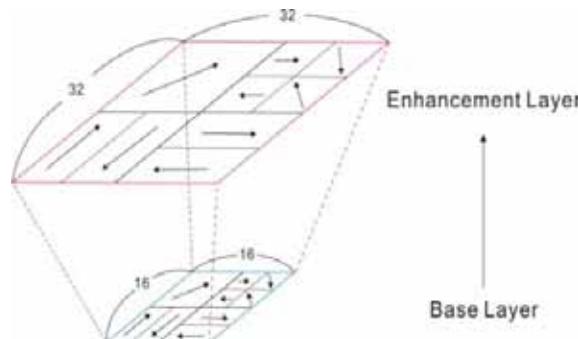


Fig. 7.36 inter-layer motion prediction [7.29].

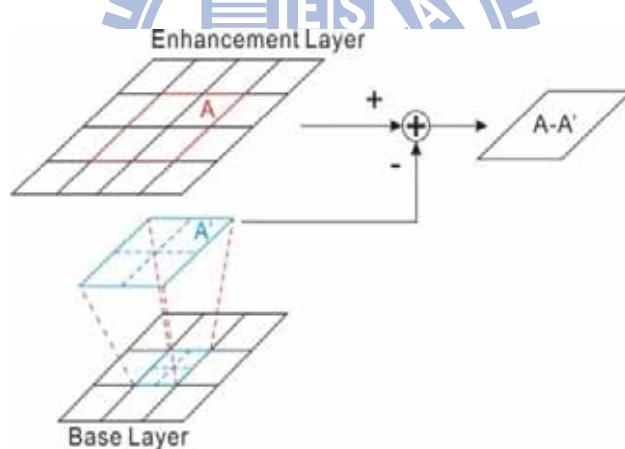


Fig. 7.37 Illustration of inter-layer residual prediction [7.29].

Fig. 7.37 presents the concept of the inter-layer residual prediction mode. Depending on the inter-layer residual prediction, the residual data is up-sampled from the corresponding 8x8 block of the base layer by bilinear interpolation. Afterward, the up-sampled residuals are used for predicting the residuals of the current macroblock in the enhancement layer.

Inter-layer intra prediction can be employed for the macroblock in the

enhancement layer if the corresponding block in base layer is intra mode block. Restated, the enhancement layer macroblock can be predicted by up-sampling the reconstructed macroblock of the base layer as shown in Fig. 7.38. For up-sampling the reconstructed macroblock in base layer, one-dimensional four-tap and bilinear filer are used for up-sampling the luminance and chrominance components, respectively.

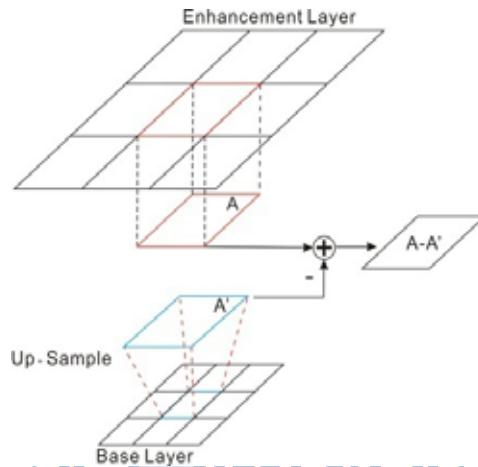


Fig. 7.38 Illustration of inter-layer intra prediction [7.29].

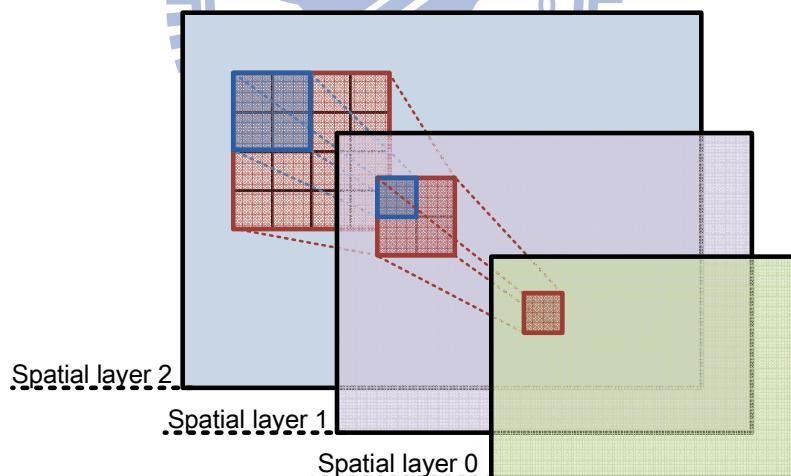


Fig. 7.39 Data relations of three spatial layers for inter-layer prediction.

For supporting spatial scalability, inter-layer prediction mechanisms are developed to increase the usage of lower layer information for improving the rate-distortion efficiency of the enhancement layers [7.30]. Fig. 7.39 shows the data relations for the inter-layer prediction. When decoding advanced spatial layer of video frames, low layer frames will be referenced frequently. From the prediction flow of inter-layer

prediction, the required data for inter-layer prediction can be obtained ahead before checking the inter-layer prediction mode for current encoding macroblock. Therefore, an Inter-layer pre-fetch scheme (IPS) is proposed for the inter-layer prediction. Based on the proposed IPS, all information required for inter-layer prediction are pre-fetched ahead before the inter-layer prediction. In order to increase processing time of the inter-layer prediction by reducing memory access latency, IPS is designed to load lower layer signals to the cache in advance.

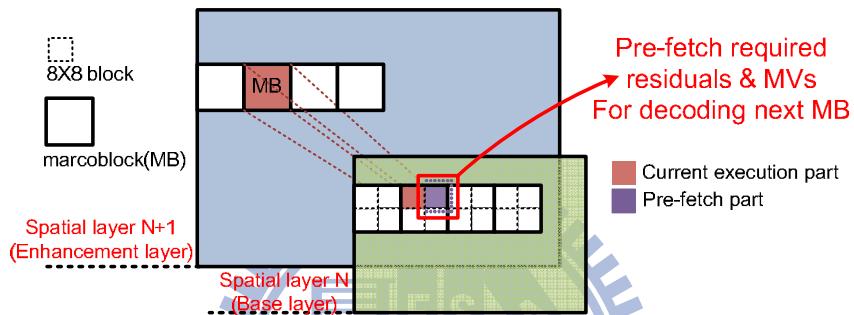


Fig. 7.40 Inter-layer pre-fetch scheme (IPS).

For the inter-layer prediction, the prediction signals are usually generated by the motion-compensated prediction in the enhancement layer or by upsampling the reconstructed lower layer signal. When decoding the advanced layer by inter-layer motion prediction and residual prediction, SVC reads the residual and motion vector(MV) signals of lower layer from memory in regular. Accordingly, IPS pre-fetches the required residual and MV data which will be referenced for decoding the following macroblock by inter-layer prediction. Fig. 7.40 gives an explanation of the proposed IPS. The green and blue frames represent the base layer frame and the enhancement layer frame, respectively. When reconstructing a macroblock, the pre-fetch engine will pre-fetch the necessary residuals and MVs in base layer for the next macroblock reconstructing. For inter-layer residual prediction, IPS loads an 8X8 block and additional residual signals which will need by bilinear interpolation. For inter-layer motion prediction, all MVs in 8X8 block are also pre-fetched by IPS.

The purple block illustrates the pre-fetch data for IPS. Based on the proposed scheme, the cache hit rate can significantly be improved for inter-layer prediction.

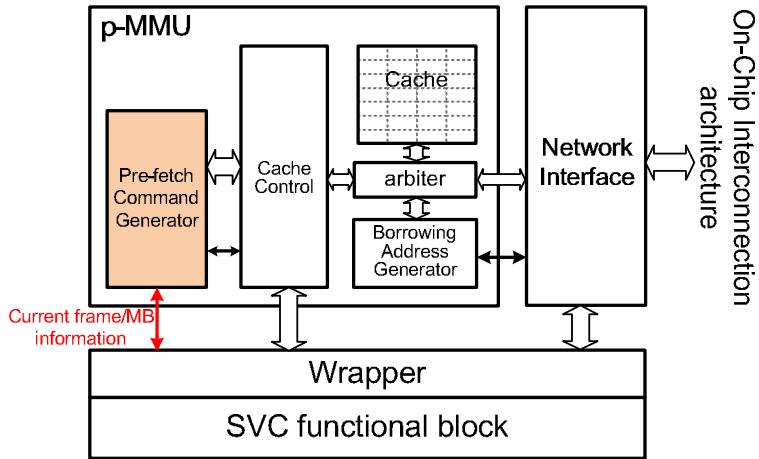


Fig. 7.41 p-MMU architecture with the pre-fetch command generator.

For the proposed IPS, a pre-fetch command generator (PCG) is constructed in p-MMU as shown in Fig. 7.41. The PCG receives the frame and macroblock information from the SVC and generates the corresponding pre-fetch commands. According to the pre-defined address allocation, PCG can produce the pre-fetch addresses and send the addresses to the cache control unit. When the cache control unit is idle or waiting the missed data, the pre-fetch commands can be served. The cache control unit checks the cache table to determine whether the pre-fetch data is in the cache. If not, an additional read miss would occur, and it will read the pre-fetch data from the lower level memory. The amount of pre-fetching data is about 128bytes (10x9 residuals and 8x8 MVs) that can be pre-fetched easily while reconstructing a macroblock.

The proposed IPS may induce the power overhead of the cache in p-MMU produced by additional pre-fetch command accesses. However, the proposed scheme can effectively reduce the cache miss rate by pre-fetching the base layer residuals and MVs. Moreover, this scheme can also reduce unnecessary cache misses of L1 cache in

p-MMU by retaining useful data in the p-MMU and further reduce the accesses of L2 cache in c-MMU. Therefore, the proposed IPS can significantly reduce the total memory energy consumption, including on-chip cache and off-chip memory.

### 7.6.2 Address Translator for SVC

The features of SDRAM and memory-access patterns of video-processing are considered to find an appropriate address translation for improving the performance of the overall system. As the increasing resolution and compressing efficiency in video-processing, the video processor should deal with a great amount of data within a tightly bounded time. However, video data are stored in the external memory that is much slower than the on-chip memory. And thus, the system performance strongly depends on the memory bandwidth between the video processor and external memory. To meet the requirement of the video processing, the regularity of memory-access patterns can be effectively exploited to reduce the execution cycles. Additionally, row-activation and pre-charge operations dominate the dynamic power consumption of the external memory. Therefore, to improve memory bandwidth and energy consumption for SVC, a new address translator is proposed to reduce the timing overhead of row-activations in DDR3 DRAM.

In a general cache memory system, adjacent blocks would address to the same row, same bank and same rank as much as possible to reduce the row conflicts and bank conflicts based on the open-page DRAM policy. With the conventional memory mapping method [7.10], [7.31], [7.32], the address translation between cache address and DRAM physical address can be accomplished as shown in Fig. 7.42. The last one bit indicates the byte offset since the data width of the selected DRAM configuration is 2-bytes (16-bits). Consequently, 10 bits for DRAM 1K column address, 3 bits for 8

bank address and 13 bits for 8K row address are defined. Based on this mapping scheme, a block access can be located in the same bank and the same row to prevent the row-conflict and bank-conflict. Additionally, the row-conflict and bank-conflict probability can be minimized for the adjacent block access.

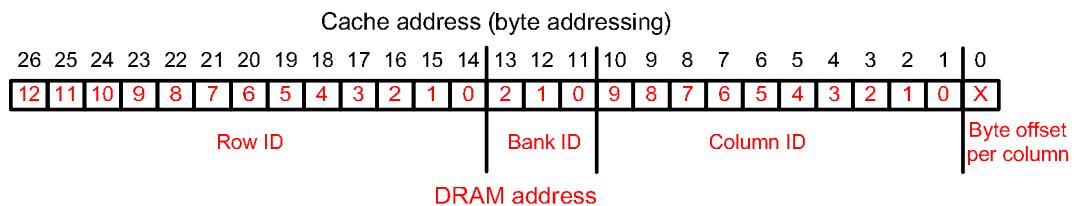


Fig. 7.42 Conventional mapping scheme for DRAM.

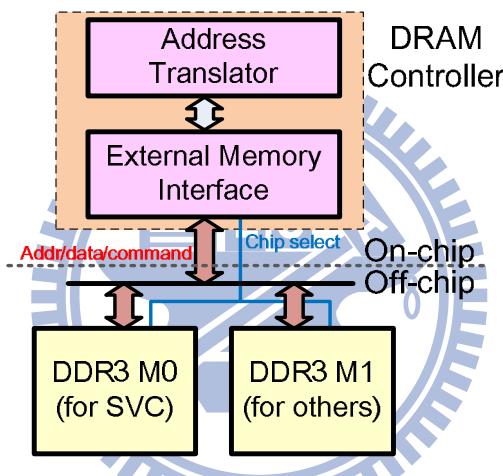


Fig. 7.43 DRAM organization in the wireless video entertainment system.

In this wireless video entertainment system, DDR3 SDRAM provided by Micron Inc. is utilized [7.18]. In order to have high DRAM bandwidth, the 64MegX16 configuration is selected because the DRAM data bus width of this configuration is larger than that of others. Additionally, two DDR3 devices are adopted that share the same bus. The DRAM organization in the wireless video entertainment system is as shown in Fig. 7.43. In order to reduce the chip I/O port, these two DRAMs share the same address, data and command bus. Because SVC requires great amount of memory accesses, a DRAM device is arranged for SVC, and the other DRAM is arranged for other PEs.

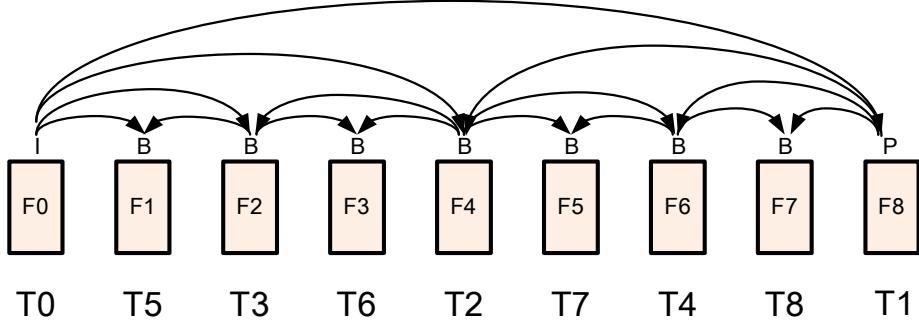


Fig. 7.44 Video frame arrangement of a GOP.

In SVC, bidirectional prediction is utilized to reduce bit rate. Fig. 7.44 shows the temporal and decoding relations of video frames in a group of pictures (GOP). For reducing the DRAM row miss rate, the video frames are allocated to different banks according to the decoding references. For instance, F4 will probably refer F0 or F8 to reconstruct the frame. In view of this, these three frames are allocated to different banks. Hence, the video decoder writes the reconstruct data to the new DRAM bank in regular. This mapping can achieve high row-hit rate for data write because of the regular write behavior for reconstructing a frame. Compared to the conventional memory mapping methods which have been adopted in many works for video frames [7.10], [7.31], [7.32], the proposed memory mapping can increase the row-hit rate of the DRAM write for the cache-based hierarchy memory sub-systems. Although the row-miss rate of the DRAM read is higher than that of other mapping methods, the performance would not be degraded because the DRAM data can be cached and reused in the cache system. Furthermore, for motion compensation, the locality of reference data strongly depend on the range of the search window defined in the encoder. While reconstructing frames, the write is more regular than the read because the write sequence is in raster-scan but the read sequence depends on the motion estimation. Accordingly, the proposed mapping can achieve low row-miss rate via reducing the DRAM row-conflicts caused by writing reconstruct frame data.

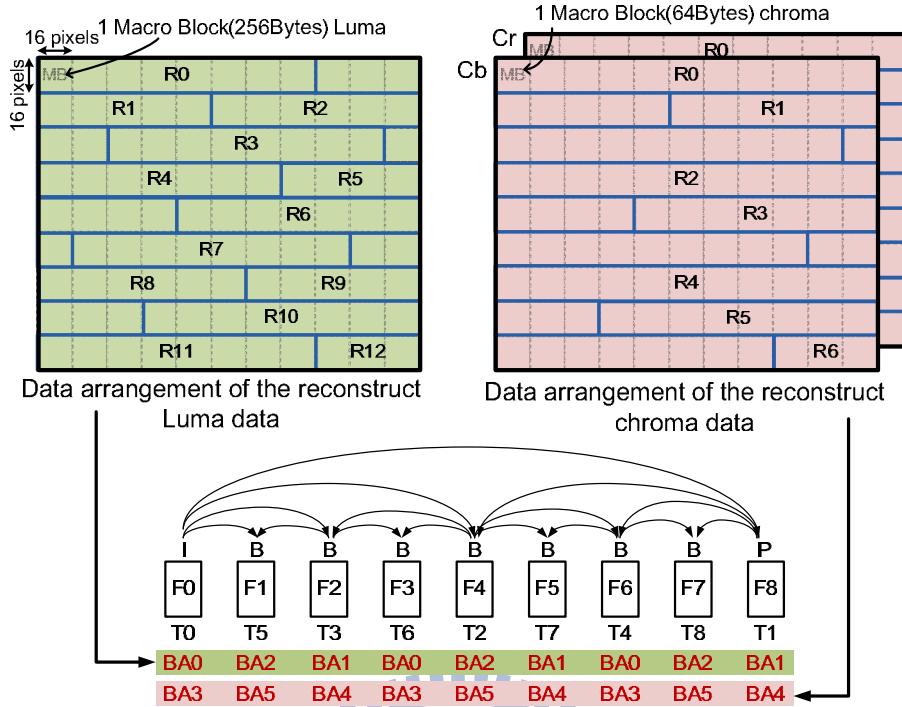


Fig. 7.45 Memory mapping for a QCIF frame.

The proposed memory mapping for a QCIF frame is as shown in Fig. 7.45. The reconstructed Luma and Chroma data are stored into the bank which is assigned by the bank interleaved scheme. Luma data will be assigned to bank 0, bank 1 or bank 2 and the Chroma data will be assigned to bank 3, bank 4 or bank 5 according to the temporal relations of the decoding video frames. In order to increase the row-hit rate of the DRAM write, the frame data is allocated in raster-scan with MB unit because the decoding sequence is also in raster-scan. The row size of the DRAM is 2Kbytes, so 8 macroblock Luma data or 16 macroblock Chroma data can be stored in to the same row. The read data for motion compensation may be irregular, but the produced reconstructed data is in raster-scan sequence. Based on the proposed mapping scheme, the data read and write for the reconstructed processing address to different banks. The produced reconstructed data can store into DRAM and have the minimum row-conflict. In addition, the residual data are placed into bank 6 and MV data are placed into bank 7 respectively.

## 7.7 Analysis of On-Demand Memory Sub-System

In this section, the simulation results and analysis of the proposed on-demand memory sub-system will be given based on the cycle-driven SystemC simulator. For measuring the energy consumptions of the on-chip caches, CACTI model [7.33], which is provided by HP Labs, is utilized to characterize the energy consumption of memory elements in p-MMU and c-MMU. CACTI is a powerful model that enables users to measure cache and memory access time, cycle time, area, leakage, and dynamic power. According to the selected cache parameters, the corresponding dynamic energy and standby leakage power can be generated, and the total energy consumption can be calculated. Additionally, the system power calculators are provided by Micron Technology Inc. [7.19] to measure the DRAM energy consumption. These power calculators can estimate the power requirement of SDRAM devices in a system environment. With the accurate estimation of power consumption, the system designer can quickly handle complex system trade-offs to optimize the system performance.

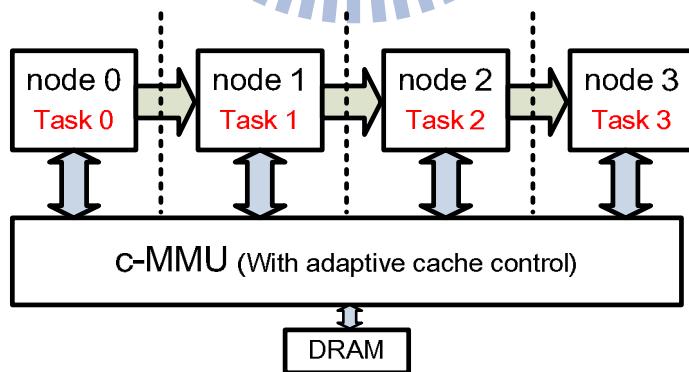


Fig. 7.46 Task-level parallel organization and flow of data stream.

For various memory demands required by different PEs in the eH-II platform, the c-MMU can support reconfigurable bank assignment for PEs. In order to simulate the behavior of the stream applications in a heterogeneous system, a task-level parallel organization is utilized as shown in Fig. 7.46. Additionally, Fig. 7.46 also presents the

flow of the data stream of these four tasks. According to different memory behaviors of nodes, the c-MMU allocates different number of SRAM banks for different nodes. The tasks with random memory accesses are applied for these four nodes, and the BAT in the c-MMU can be updated by the system for the bank assignment. Additionally, the memory requirements are different during runtime, and the requirements for three time intervals (500 tasks in a time interval) are listed in Table 7.5. With profiling the memory requirement and re-allocating the bank assignment as list in Table 7.5, the memory resources in c-MMU can be utilized effectively. When finishing 1500 tasks in three time intervals, 40.41% execution cycles and 48.54% memory energy reductions can be achieved compared to the fixed bank assignment with equal number of banks.

Table 7.5 Memory requirement assumption and corresponding bank assignment for c-MMU

Time	T0	T1	T2
Memory requirements (node x → Memory usage percentage)	node 0 → 25% node 1 → 19% node 2 → 37% node 3 → 19% Unused → 0%	node 0 → 19% node 1 → 19% node 2 → 50% node 3 → 12% Unused → 0%	node 0 → 19% node 1 → 19% node 2 → 44% node 3 → 18% Unused → 0%
Bank Assignment (node x → # of banks)	node 0 → 4 node 1 → 3 node 2 → 6 node 3 → 3 Turn-off → 0	node 0 → 3 node 1 → 3 node 2 → 8 node 3 → 2 Turn-off → 0	node 0 → 3 node 1 → 3 node 2 → 7 node 3 → 3 Turn-off → 0

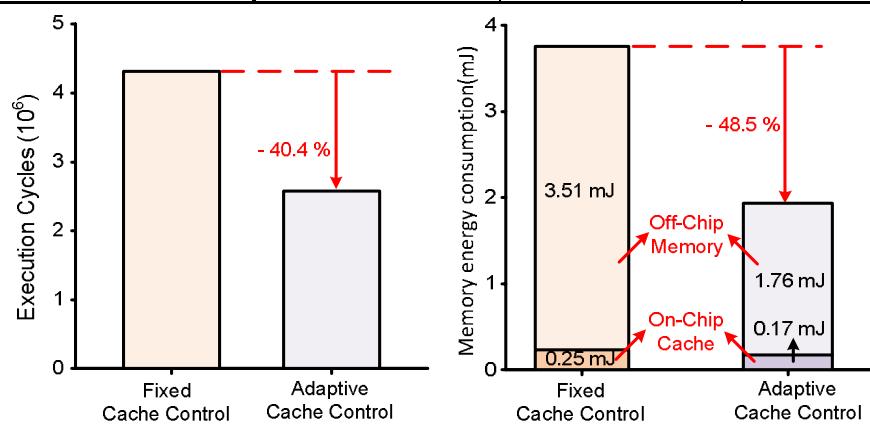


Fig. 7.47 Total execution cycles and memory energy consumption.

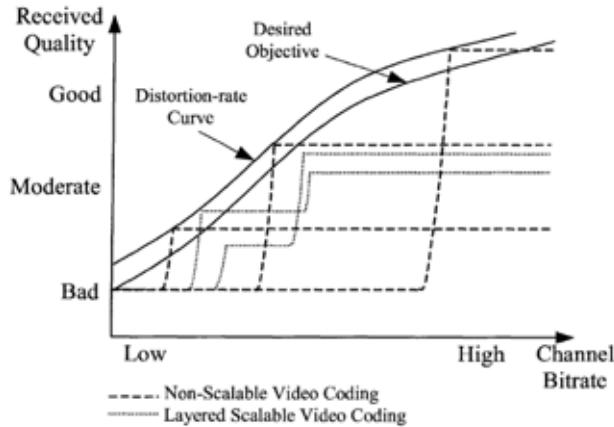


Fig. 7.48 Video quality versus channel bit-rate [7.34]

For the wireless video entertainment system, SVC can optimize the video quality over a given bit rate range. Generally, a non-scalable video encoder generates the compressed bitstream with a fixed resolution and quality. In contrast, a scalable video encoder compresses a raw video sequence into multiple layers [7.34]. One of the compressed layers is the base layer, which can be independently decoded and provide coarse visual quality. Other compressed layers are enhancement layers, which can only be decoded according to the base layer and can provide better visual quality. The complete bitstream (i.e., including all layers) provides the highest quality. In the receiver, the quality level and resolution of the video can be reconstructed by SVC depending on the wireless channel bandwidth or the application in the end-user device. Fig. 7.48 presents the video qualities for different channel bitrates, and the distortion-rate curve represents the upper bound in quality for any coding technique at the given bit-rate. The non-scalable single staircase curve is changed to a curve with several stairs by SVC. In the wireless video entertainment system, SVC requires great amount of memory resource to store the frame data. However, decoding different scalable levels has different memory requirements for storing the reconstruct frames of different layers. The memory requirements of the deterministic scalable layers for a GOP are shown in Fig. 7.49, and the summary of the SVC parameters are listed in

Table 7.6.

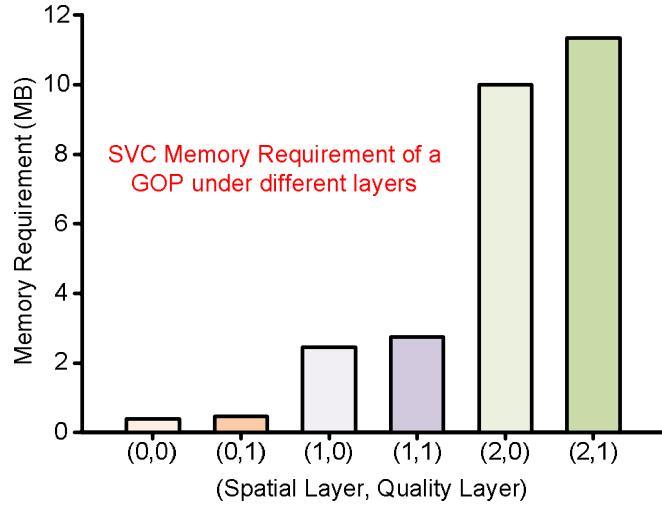


Fig. 7.49 SVC memory requirements of different scalable layers for a GOP.

Table 7.6 Summaries of SVC parameters

SVC parameters	
Number of Spatial layer	3
Spatial layers	QCIF(177x146)-CIF(352-288)-4CIF(704x576)
Number of Quality layer	2
Quality layers	QPBL : 32 - QPEL : 16
Frame rate	30fps
GOP	8 (I-B-B-B-B-B-B-P)
Sequence	Stephen

In the wireless video entertainment system, the effective bandwidth of the wireless channel can be detected by MAC. According to the detection of the wireless channel, the transmitter can determine the scalable level of SVC bitstream to satisfy the effective bandwidth. Based on various bitstream, the memory requirement of different quality and resolution levels is also various and can be profiled off-line. With profiling the SVC memory requirements and dynamically updating the BAT in c-MMU by MAC, the appropriate bank assignments for SVC in different effective bandwidths can be achieved. The memory configurations of different quality levels are listed in Table 7.7. The bank assignment in the c-MMU is also been profiled under different SVC decoding levels. When SVC is required to decode high spatial and quality layer frames, the adaptive cache controller assigns more banks than that for

low spatial and quality layer frames. The adaptive cache controller can reduce the miss rate of frame reconstructing. And thus, the number of DRAM accesses can also be decreased. In contrast, the adaptive cache controller can turn-off unused banks in the c-MMU when decoding low spatial and quality layer frames. Fig. 7.50 and Fig. 7.51 present the energy comparison and execution time of memory for decoding a GOP under different SVC levels, respectively. Compared to the fixed bank assignment with equal banks, the adaptive bank assignment can reduce both the execution time and energy consumption for decoding enhancement layers in SVC.

Table 7.7 c-MMU bank assignment for wireless video entertainment systems

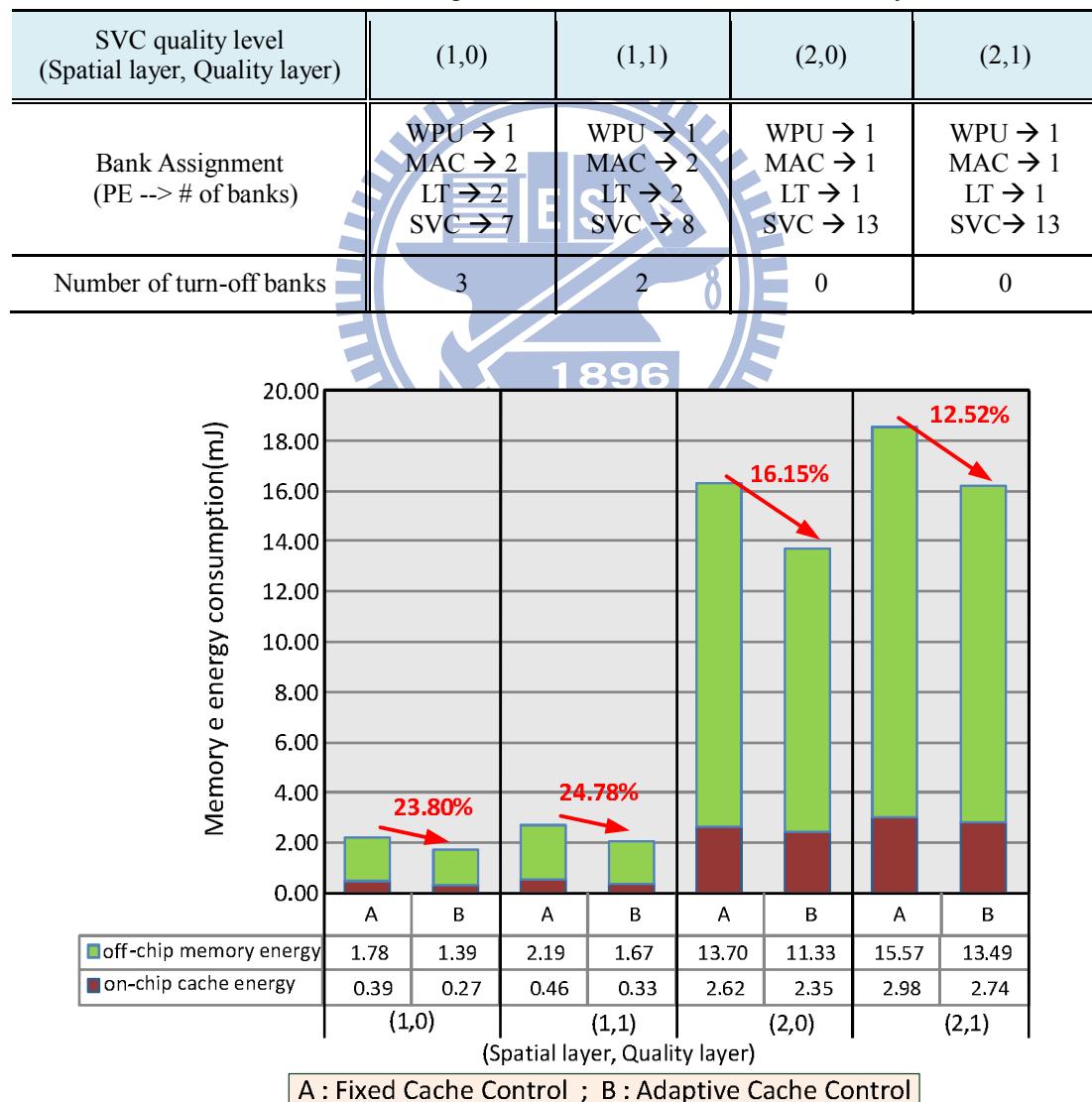


Fig. 7.50 Memory energy consumption for different SVC levels.

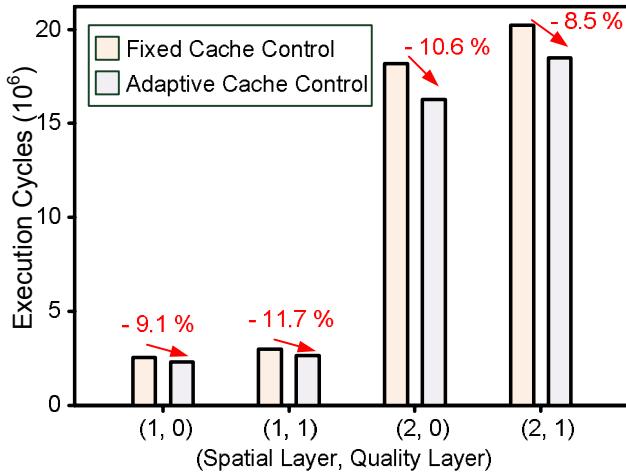


Fig. 7.51 Execution cycles for different SVC levels.

The proposed adaptive cache controller in the c-MMU can support cache reconfiguration for different bank assignments in different time intervals by checking BAT, which can be updated by MAC at runtime. Fig. 7.52 gives an example of various bit-rates in the wireless channel, and presents the corresponding SVC quality levels. Based on the bank assignment of different SVC quality levels listed in Table 7.7, the total execution cycles and memory energy consumption are as shown in Fig. 7.53. Therefore, 7.13% execution time reduction and 10.53% memory energy consumption reduction can be realized using the adaptive cache control.

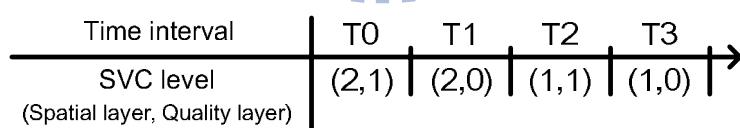


Fig. 7.52 Various bit-rates in the wireless channel and the corresponding SVC quality levels.

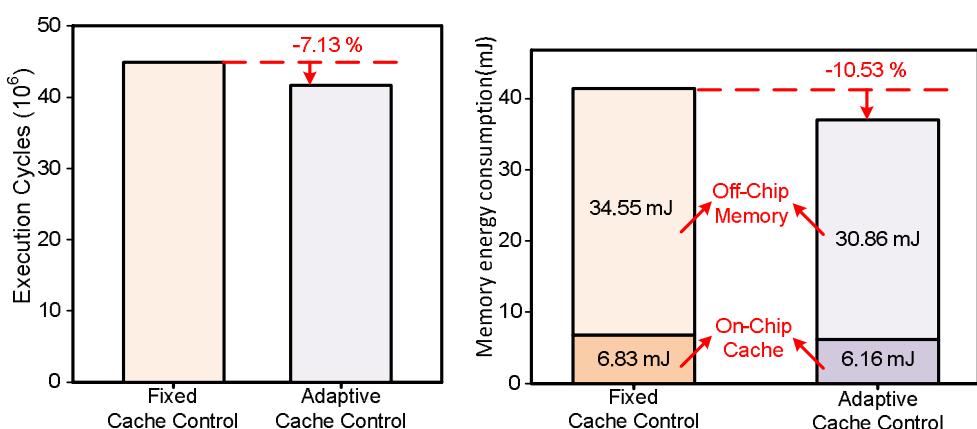


Fig. 7.53 Total execution cycles and memory energy consumption.

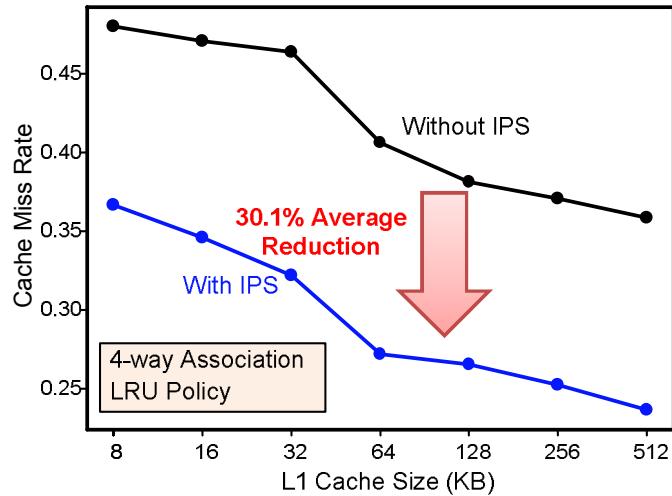


Fig. 7.54 Miss rate of the L1 cache versus L1 cache size with/without inter-layer pre-fetch scheme.

Based on the proposed IPS, the miss rate of the distributed cache (L1 cache) can be reduced for the SVC. Based on 4-way associativity cache configuration and least recently used (LRU) replacement policy of the p-MMU, the IPS can reduce 30.01% miss rate on average in the p-MMU as shown in Fig. 7.54. Furthermore, IPS can reduce unnecessary cache misses in L1 cache by retaining useful data, which are accessed recently. Therefore, the number of L2 memory accesses and DRAM accesses can also be reduced by 24.6% and 34% on average via the reduction of the miss rate in the p-MMU as shown in Fig. 7.55 and Fig. 7.56, respectively. Moreover, the number of DRAM accesses caused by L2 cache data replacement also can be reduced.

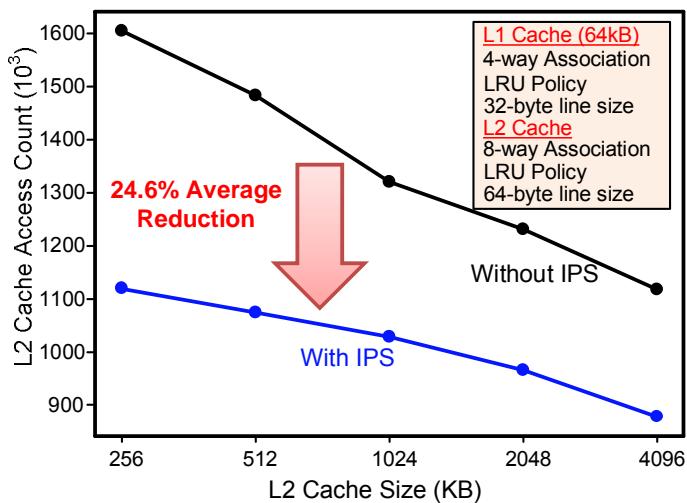


Fig. 7.55 Memory accesses of L2 cache with/without inter-layer pre-fetch scheme.

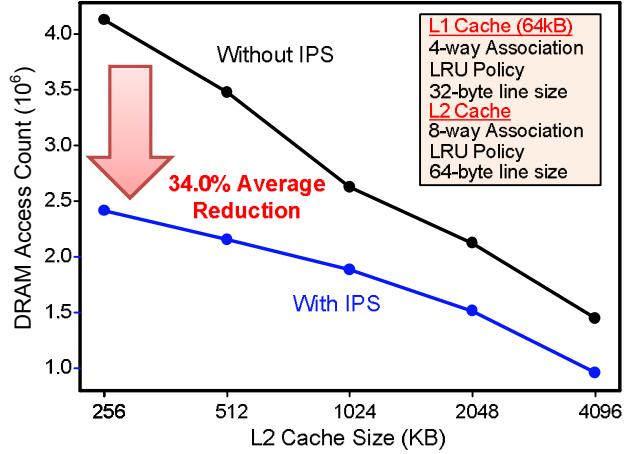


Fig. 7.56 Memory accesses of DRAM with/without inter-layer pre-fetch scheme.

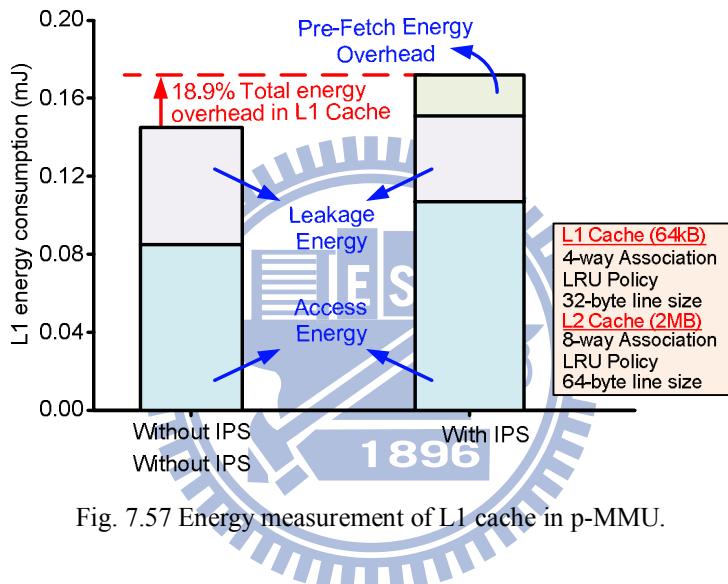


Fig. 7.57 Energy measurement of L1 cache in p-MMU.

Additional energy overhead is induced using the proposed IPS because of extra cache accesses generated by the pre-fetch requests. Fig. 7.57 presents the energy measurement of L1 cache via the CACTI model. The green part is the pre-fetch energy overhead produced by the miss pre-fetch requests. The red and the blue parts indicate the standby leakage energy consumption and the access energy, respectively. The access energy with IPS is larger than that without the pre-fetch scheme since the additional pre-fetch requests. However, the standby leakage energy can be saved because IPS reduces the cache miss rate and the execution time. The total energy overhead produced by IPS is 18.9%.

The address translator converts the original addresses to appropriate DRAM

addresses to reduce the DRAM row-miss rate. Fig. 7.58 and Fig. 7.59 present the DRAM row-miss rate and number of DRAM row-conflict with different L2 cache size, respectively. Compared to the conventional data allocation, the proposed address translator can further reduce row-miss rate and number of row-conflict in DRAM for video application. Additionally, the row-miss rate and row-conflict also can be reduced using IPS since the number of DRAM accesses is decreased. On average, 60.64% row-miss rate and 74.35% number of row-conflict reduction can be achieved by the combination of the proposed IPS and address translator.

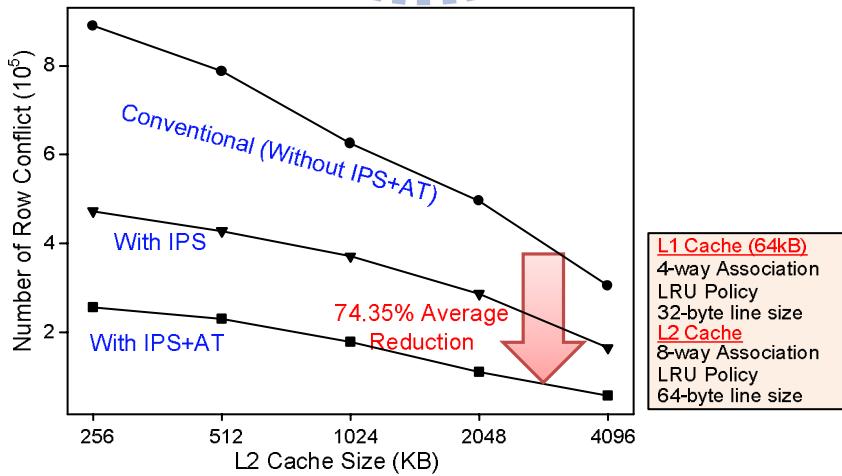
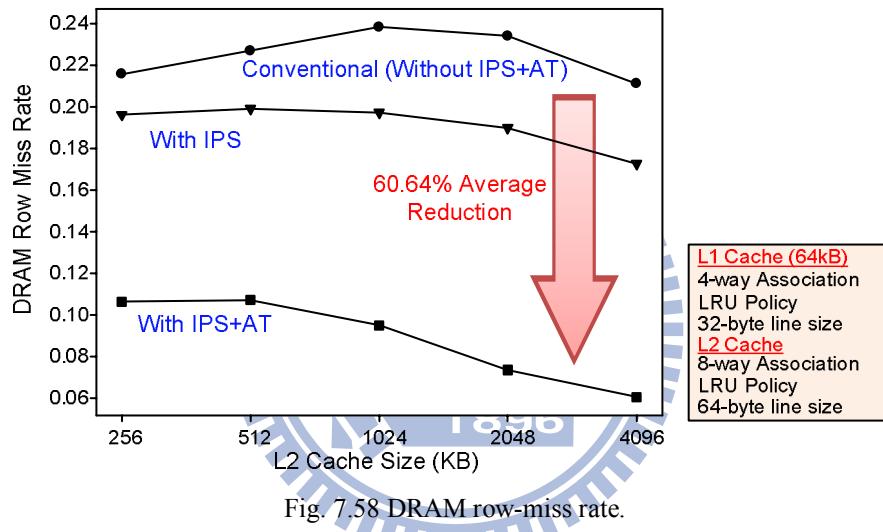


Fig. 7.59 Number of DRAM row-conflict.

Reducing row-miss rate can decrease the DRAM activate power and improve the DRAM access bandwidth utilization. Fig. 7.60 shows the simulation results of the

DRAM activate power. The proposed pre-fetch and data allocation method can reduce the activate power about 57.19% on average. Furthermore, 24.87% DRAM bandwidth utilization, which is defined as Eq. (7.1), can be improved via the combination of the proposed IPS and address translator compared to the conventional mapping method.

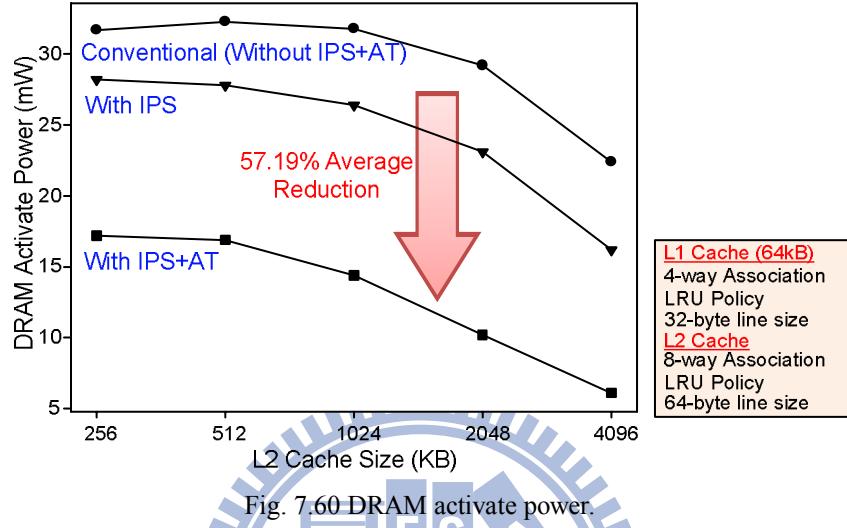


Fig. 7.60 DRAM activate power.

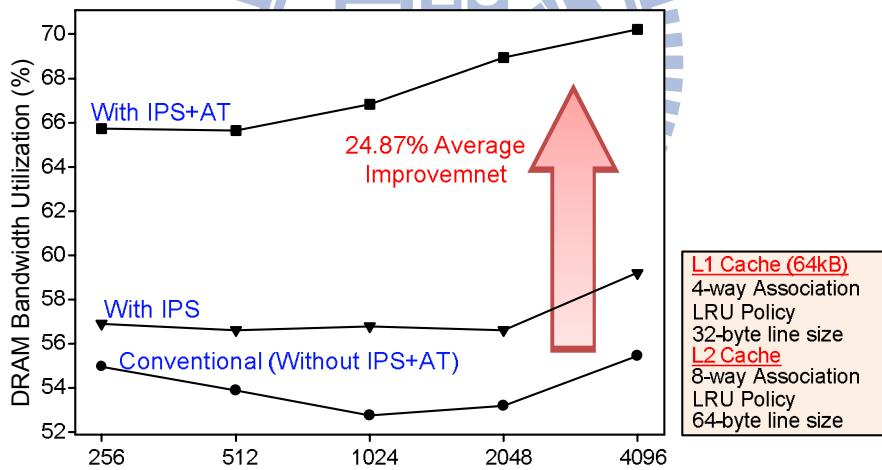


Fig. 7.61 DRAM bandwidth utilization.

Consequently, DRAM energy consumption can also be reduced via the proposed mechanisms as shown in Fig. 7.62. The DRAM read/write power dominates the DRAM power consumption. Reducing number of DRAM accesses can decreases the total DRAM energy consumption. Therefore, the proposed IPS can reduce the execution time and energy consumption significantly by directly decreasing the number of DRAM accesses with low cache miss rate. Furthermore, the DRAM energy

can be reduced using the proposed address translator by saving the activate power of DRAM.

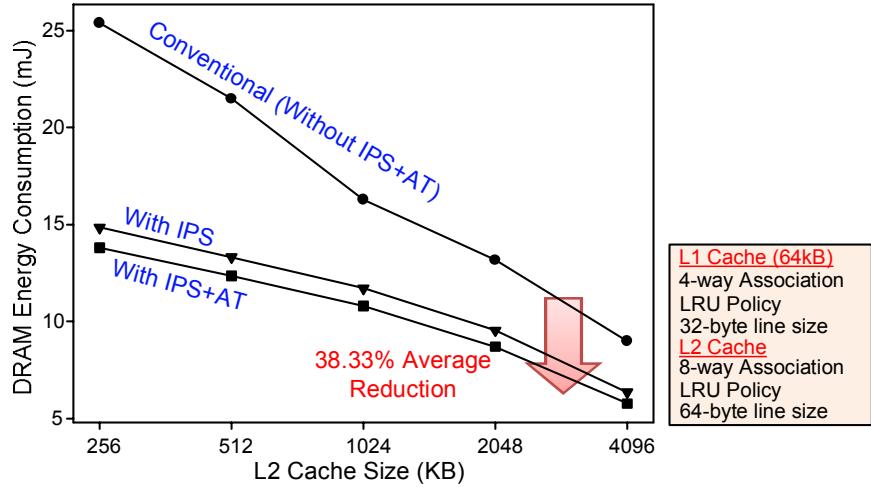


Fig. 7.62 DRAM energy consumption.

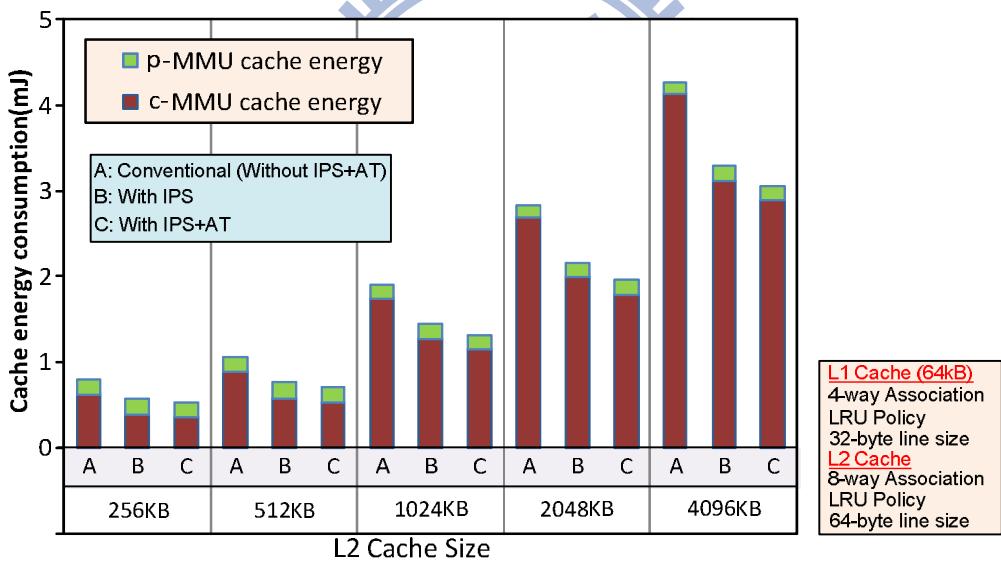


Fig. 7.63 On-chip cache energy consumption.

The total on-chip cache energy consumption with different L2 cache sizes, ranging from 256KB to 4MB, are simulated as shown in Fig. 7.63. Increasing L2 cache size increases the cache energy consumptions and decreases the cache miss rate. Therefore, The DRAM energy consumption can be reduced via the decreasing of the cache miss-rate. Fig. 7.64 illustrates the total memory energy consumption including on-chip cache and off-chip DRAM with different L2 cache sizes. Although the IPS

induces 18.9% energy overhead in the p-MMU, the execution time and number of L2 cache and DRAM accesses can be reduced. Therefore, the total memory energy consumption can be reduced. Based on the combination of the proposed IPS and address translator, 37.53% energy reduction can be achieved on average.

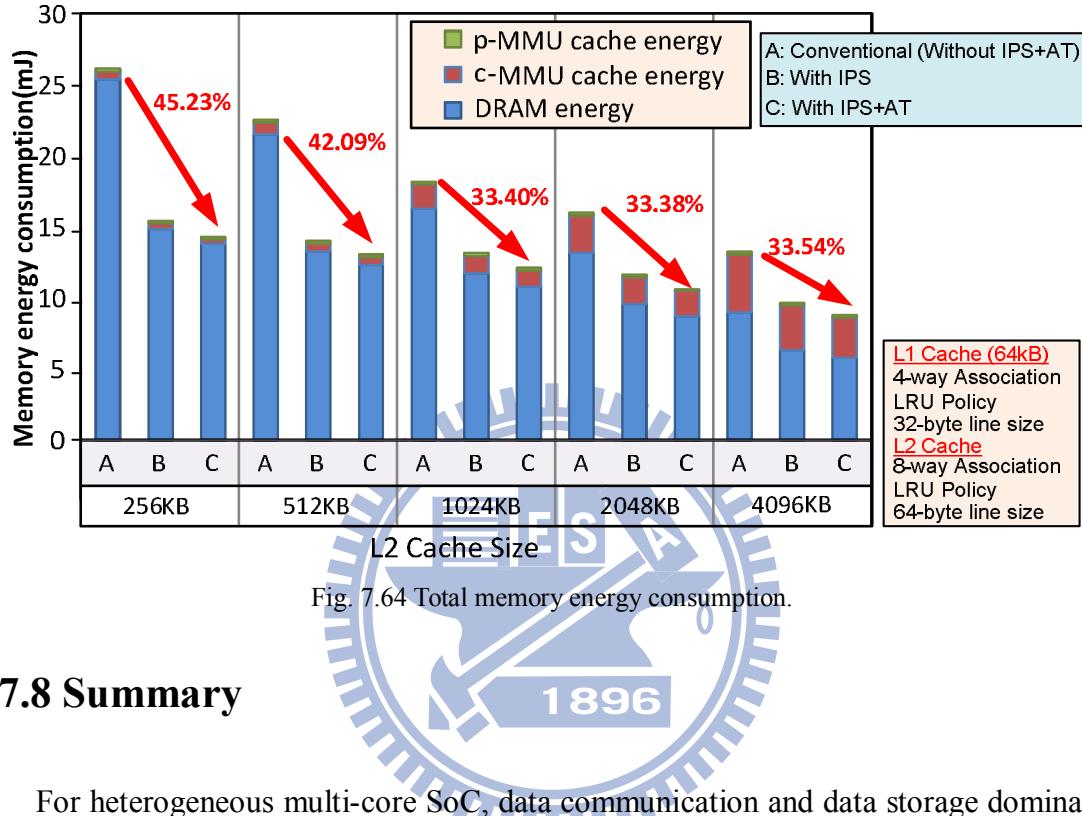


Fig. 7.64 Total memory energy consumption.

## 7.8 Summary

For heterogeneous multi-core SoC, data communication and data storage dominate the overall performance. In this chapter, an on-demand memory sub-system is presented to construct a memory-centric on-chip data communication platform, eH-II platform, for wireless video entertainment systems. In the proposed on-demand memory sub-system, MMUs, including p-MMUs and a c-MMU, can efficiently control the memory access and memory resource allocation for PEs. The p-MMU performs as a high level cache for the dedicated PE in the on-demand memory sub-system. Furthermore, for reducing the stall caused by heavy traffic in OCIN, a buffer borrowing mechanism is proposed. The p-MMU can dynamically allocate the memory resources for buffering the blocking network data. By considering the borrowed memory blocks and p-MMU, the size of the output queue in NI can be

dynamically scheduled. According to the cycle-driven simulation results in SystemC, the proposed efficient NI can achieve performance improvement by 1.15x compared to the conventional one. Therefore, the proposed efficient NI can increase the performance of the memory-centric OCIN.

The c-MMU is designed for managing and providing larger centralized memory resources for the system. Based on the adaptive cache controller, the proposed c-MMU can dynamically assign variable number of SRAM banks for PEs to optimize the utilization of centralized on-chip cache can be optimized. By profiling the memory requirements for different SVC spatial and quality layers, the appropriate bank assignment can be determined and controlled by MAC at runtime. From the simulation results, 7.13% execution time reduction and 10.53% memory energy consumption reduction can be realized using the adaptive cache control. Additionally, an EMI of the DRAM controller in the c-MMU is applied to access external memory efficiently. By re-scheduling the DRAM commands, the effective bandwidth of DRAM can be improved by 1.42x.

For SVC in the wireless video entertainment system, the IPS and address translator are proposed to improve the decoding performance and integrated in the p-MMU and c-MMU, respectively. While reconstructing the frames, the required information for the inter-layer prediction in SVC is pre-fetched ahead to reduce the cache miss significantly using the proposed IPS. The simulation result shows that the proposed IPS can reduce both the overall memory energy consumptions and the execution time. Furthermore, the address translator for video applications in the DRAM controller is realized to improve DRAM memory bandwidth efficiency and reduce DRAM activated power consumption. The proposed IPS and address translator not only reduce cache miss rate but also reduce total memory energy consumptions. From the

simulation results, 38.83% execution time reduction and 37.53% memory energy reduction can be achieved via the combination of the IPS and address translator.



# ***Chapter 8:***

## ***Conclusions and Future Works***

### **8.1 Conclusions**

As design complexity of multi-core SoC (system-on-chip) continues to increase, a global approach is needed to effectively transport and manage on-chip communication traffic, and to optimize wire efficiency. Based on the crucial issues in multi-core SoCs, including the energy bound and the increasing requirements of data communication and data storage, an energy-efficient on-chip data communication platform is presented in this dissertation. This on-chip data communication platform is constructed using a memory-centric on-chip interconnection network (OCIN) and an on-demand memory sub-system. The memory-centric OCIN provides building blocks and the micro-architecture for the platform with the on-demand memory sub-system, including energy-efficient and reliable channels, congestion-aware routing algorithm, energy-efficient routing table, two-level FIFO buffer and buffer-efficient NIs. In addition, the on-demand memory sub-system provides high bandwidth and low power memory accesses for multi-core SoCs via a centralized memory management unit (c-MMU) and private memory management units (p-MMUs). The on-demand memory sub-system can support variety memory resources for different PEs based on the memory behaviors. Moreover, when decoding the video frames, memory access characteristics of video decoders are generally regular and repeat. Therefore, the on-demand memory sub-system can improve the decoding performance via efficient memory management. Table 8.1 summarizes the contributions in this dissertation, including the building blocks of the memory-centric OCIN and the on-demand

memory sub-system.

Table 8.1 The summary of this dissertation

Memory-Centric On-Chip Interconnection Network (OCIN)	
Self-calibrated energy-efficient and reliable channel	<ul style="list-style-type: none"> <li>• Self-corrected green (SCG) coding scheme</li> <li>• Self-calibrated voltage scaling technique</li> </ul>
Two-level FIFO buffer router	<ul style="list-style-type: none"> <li>• Shared buffer mechanism</li> <li>• Multiple accesses</li> </ul>
An adaptive congestion-aware routing algorithm	<ul style="list-style-type: none"> <li>• Adaptive routing to tolerate hot spots</li> </ul>
Energy-efficient ternary content addressable memories (TCAM) for routing tables	<ul style="list-style-type: none"> <li>• Butterfly match-line scheme</li> <li>• XOR Conditional Keeper</li> <li>• Hierarchy search-line scheme</li> <li>• Super cut-off power gating</li> <li>• Multi-mode data-retention power gating</li> </ul>
Efficient Network Interfaces (NIs)	<ul style="list-style-type: none"> <li>• Memory borrowing mechanism</li> </ul>
On-Demand Memory Sub-System	
Private memory management unit (p-MMU)	<ul style="list-style-type: none"> <li>• Dedicated L1 Cache</li> <li>• Memory borrowing mechanism</li> <li>• Inter-layer pre-fetch scheme for SVC</li> </ul>
Centralized Memory Management Unit (c-MMU)	<ul style="list-style-type: none"> <li>• Centralized L2 Cache</li> <li>• Adaptive cache control</li> <li>• Efficient external memory interface (EMI) for DDR3 DRAM</li> <li>• Address translator in the DRAM controller for SVC</li> </ul>

In near future, with the advancement of the wireless communication and multimedia techniques, great amount of digital electronic devices will be developed in human life. The proposed energy-efficient on-chip data communication platform provides an integrated solution for heterogeneous multi-core SoCs and wireless video entertainment systems.

## 8.2 Futures Works

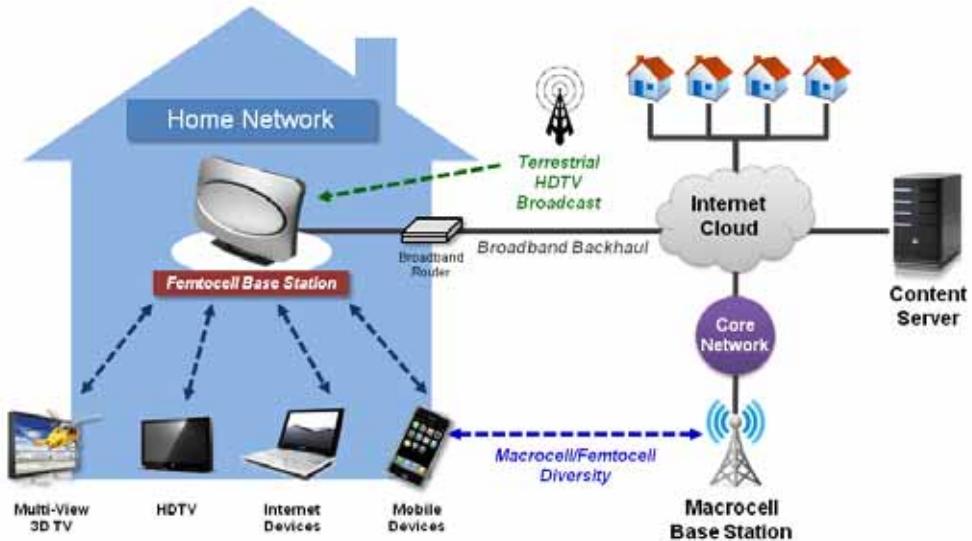


Fig. 8.1 A femtocell home multimedia center.

With the accelerated progress in video sizes and quality of mobile multimedia services, multi-view 3D video, is now regarded as the future star in global multimedia industry. The conventional macro base stations designed for wide service coverage cannot meet the increasing demands in 3D high-quality transmission especially for indoor applications. The emerging Femtocell with wire-line backhaul has been proposed as one of the key deployment architectures in 4G wireless systems to achieve 1 Gbps transmission rates as shown in Fig. 8.1. In order to meet these future transmission challenges and to support multiple wideband 3D video streaming services, an energy efficient reconfigurable multi-core platform for intelligent multimedia femtocell systems should be developed. This multimedia femtocell is expected to support massive computation for high definition multi-view 3D video and to provide intelligent resource allocation schemes for achieving multi-user high-quality transmission without interfering by other macrocell base stations or self-deployed femtocells. Furthermore, beyond 4G requirements, an over 3 Gbps transmission platform, based on reconfigurable multi-streaming multi-antenna

wireless signal processing and multi-mode multi-streaming resource controls, will be developed. Moreover, including hierarchical memory management, the power control, and cross layer hardware/software co-design, this multi-core platform will be able to optimize the application-specific interconnection performance through the reconfigurable memory centric and fault tolerant system designs. Finally, in order to have optimal power control, a novel power management unit will be provided and integrated with the on-chip data communication platform.

For the research of building blocks of OCINs, the joint effect of combining different proposed techniques did not analyze in this dissertation. However, the overall performance would be decreased since the proposed techniques in different layers are mutually exclusive. Therefore, a systematic analysis of the building blocks in this dissertation should be developed to point out the optimized design among cross layers. Additionally, triplication error correction stage for reliable channels increases the area overhead significantly. The SCG coding scheme can be improved by considering the characteristics of all flits in a packet.

All the simulation and measurement results in this dissertation were based on UMC 65nm CMOS technology. However, the circuit design will be influenced by high-k metal gates while shrinking below 45nm CMOS technologies. Moreover, PVT (process, voltage, temperature) variations increases rapidly, and further degrades the functionality of our designs. Therefore, a PVT-tolerant circuit design has been considered to provide a reliable on-chip data communication platform.

# Bibliography

---

## References of Chapter 1

- [1.1] K. Ahmad and A. C. Begen, "IPTV and Video Networks in the 2015 Timeframe: The Evolution to medianets," *IEEE Communications Magazine*, Vol. 47, No. 9, Sept. 2009.
- [1.2] (2005-2009) International Technology Roadmap for Semiconductors. Semiconductor Industry Assoc. [Online]. Available: <http://public.itrs.net>
- [1.3] V. Chandra, A. Xu, H. Schmit and L. Pileggi, "An interconnect channel design methodology for high performance integrated circuits," in *Proceeding of Design Automation Test European Conference Exhibition*, Vol. 2, pp. 1138-1143, Mar. 2004.
- [1.4] P. Wodey, G. Caarque, F. Baray, R. Hersemeule and J.P. Cousin, "LOTOS Code Generation for Model Checking of STBus Based SoC: The STBus Interconnect," in *Proceeding of ACM/IEEE International Conference on Formal Methods and Model for Co-design*, pp. 204-213, 2003.
- [1.5] K.-C. Chang, J.-S. Shen and T.-F. Chen, "Evaluation and Design Trade-Offs between Circuit Switched and Packet Switched NOCs for Application-Specific SOCs," in *Proceeding of Design Automation Conference*, pp. 143-148, Jun. 2006.
- [1.6] V. Chandra, A. Xu, H. Schmit and L. Pileggi, "An interconnect channel design methodology for high performance integrated circuits," in *Proceeding of IEEE Design, Automation and Test in Europe Conference and Exhibition*, Vol. 2, pp. 1138 -1143, 2004.
- [1.7] J. Muttersbach, T. Villiger, and W. Fichtner, "Practical design of globally asynchronous locally synchronous systems," in *Proceeding of International Symposium on Advanced Research of Asynchronous Circuits and System*, pp. 52-59, Apr. 2000.
- [1.8] L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computer*, Vol. 35, pp. 70-78, Jan. 2002.
- [1.9] R.I. Bahar, D. Hammerstrom, J. Harlow, W. H. Joyner Jr., C. Lau, D. Marculescu, A. Orailoglu, and M. Pedram, "Architectures for Silicon Nanoelectronics and Beyond," *IEEE Computer*, Vol. 40, No. 1, pp. 25-33, Jan. 2007.

[1.10] D. Zydek, N. Shlayan, E. Regentova, and H. Selvaraj, “Review of Packet Switching Technologies for Future NoC,” in *Proceeding International Conference on System Engineering*, pp. 306-311, Aug. 2008.

[1.11] P.P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, “Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures,” *IEEE Transaction on Computer*, Vol. 54, No. 8, pp. 1025-1040, Aug. 2005

[1.12] L. Benini and G. De Micheli, *Network on Chips: Technology and Tools*, Morgan Kaufmann, 2006.

[1.13] W. J. Dally and B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann, 2004.

[1.14] P. Schaumont, B.C. Lai, W. Qin, I. Verbauwhede, “Cooperative multithreading on embedded multiprocessor architectures enables energy-scalable design,” in *Proceeding of Design Automation Conference*, pp. 27-30, June 2005.

[1.15] G. Delagi, “Harnessing technology to advance the next-generation mobile user-experience,” *Digest of Technical IEEE International Solid-State Circuits Conference Papers*, pp.18-24, Feb. 2010.

[1.16] J. Semiao, et al., “The Management for Low-Power Design of Digital Systems,” *Journal of Low Power Electronics*, Vol. 4, No. 3, pp. 410-419, Dec. 2008.

[1.17] C. Kutter, “Design Challenge for Mobile Communication Devices,” in *Proceeding of IEEE International Symposium on Low Power Electronics and Design*, pp. 1-2, Oct. 2006.

[1.18] R. Kakerow, “Low power design methodologies for mobile communication,” in *Proceeding of IEEE International Conference on Computer Design*, pp. 8-13, Sept. 2002.

## References of Chapter 2

[2.1] J. Howard, S. Dighe, Y. Hoskote, S. Vangal, D. Finan, G. Ruhl, D. Jenkins, H. Wilson, N. Borkar, G. Schrom, F. Paillet, S. Jain, T. Jacob, S. Yada, S. Marella, P. Salihundam, V. Erraguntla, M. Konow, M. Riepen, G. Droege, J. Lindemann, M. Gries, T. Apel, K. Henriss, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, R. Van Der Wijngaart, and T. Mattson, “A 48-Core IA-32 message-passing processor with DVFS in 45nm CMOS,” in *Proceeding of IEEE International Solid-State Circuits Conference*, pp.108-109, Feb. 2010.

[2.2] N. A. Kurd, S. Bhamidipati, C. Mozak, J. L. Miller, T. M. Wilson, M. Nemani, and M. Chowdhury, “Westmere: A family of 32nm IA processors,” in

*Proceeding of IEEE International Solid-State Circuits Conference*, pp.96-97, Feb. 2010.

- [2.3] J. L. Shin, K. Tam, D. Huang, B. Petrick, H. Pham, C.-K. Hwang, H.-P. Li, A. Smith, T. Johnson, F. Schumacher, D. Greenhill, A. S. Leon, and A. Strong, “A 40nm 16-core 128-thread CMT SPARC SoC processor,” in *Proceeding of IEEE International Solid-State Circuits Conference*, pp.98-99, Feb. 2010.
- [2.4] Y. Yuyama, M. Ito, Y. Kiyoshige, Y. Nitta, S. Matsui, O. Nishii, A. Hasegawa, M. Ishikawa, T. Yamada, J. Miyakoshi, K. Terada, T. Nojiri, M. Satoh, H. Mizuno, K. Uchiyama, Y. Wada, K. Kimura, H. Kasahara, and H. Maejima, “A 45nm 37.3GOPS/W heterogeneous multi-core SoC,” in *Proceeding of IEEE International Solid-State Circuits Conference*, pp.100-101, Feb. 2010.
- [2.5] C. Johnson, D. H. Allen, J. Brown, S. Vanderwiel, R. Hoover, H. Achilles, C.-Y. Cher, G. A. May, H. Franke, J. Xenedis, and C. Basso, “A wire-speed power<sup>TM</sup> processor: 2.3GHz 45nm SOI with 16 cores and 64 threads,” in *Proceeding of IEEE International Solid-State Circuits Conference*, pp.104-105, Feb. 2010.
- [2.6] L. Benini and G. De Micheli, “Networks on Chips: A New SoC Paradigm,” *IEEE Computer*, Vol. 35, pp. 70-78, Jan. 2002.
- [2.7] L. Benini and G. De Micheli, *Network on Chips: Technology and Tools*, Morgan Kaufmann, 2006.
- [2.8] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufmann, 2004.
- [2.9] W. J. Dally and B. Towles, “Route Packets, Not Wires: On-Chip Interconnection Networks,” in *Proceeding of Design Automation Conference*, pp.684-689, 2001.
- [2.10] M. Drinic, D. Kirovski, S. Megerian, and M. Potkonjak, “Latency-Guided On-Chip Bus-Network Design,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, No. 12, pp. 2663-2673, Dec. 2006.
- [2.11] V. Chandra, A. Xu, H. Schmit and L. Pileggi, “An interconnect channel design methodology for high performance integrated circuits,” in *Proceeding of IEEE Design, Automation and Test in Europe Conference and Exhibition*, Vol. 2, pp. 1138-1143, 2004.
- [2.12] H. Lekatsas and J. Henkel, “ETAM++: extended transition activity measure for low power address bus designs,” in *Proceeding of VLSI Design*, pp. 113-120, 2002.

[2.13] K.-H. Baek, K.-W. Kim and S.-M. Kang, “A low energy encoding technique for reduction of coupling effects in SoC interconnects,” *IEEE Midwest Symposium on Circuit and Systems*, Vol.1, pp. 80-83, 2000.

[2.14] C.-G. Lyuh and T. Kim, “Low power bus encoding with crosstalk delay elimination,” in *Proceeding of IEEE ASIC/SoC Conference*, pp. 389-393, 2002.

[2.15] T. Lv, J. Henkel, H. Lekatsas and W. Wolf, “An adaptive dictionary encoding scheme for SOC data buses,” in *Proceeding of IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp. 1059-1064, 2002.

[2.16] Kang Min Lee, Se-Joong Le and Hoi-Jun Yoo, “Low energy transmission coding for on-chip serial communications,” in *Proceeding of IEEE System-on-Chip Conference*, pp. 177-178, 2004.

[2.17] (2005-2009) International Technology Roadmap for Semiconductors. Semiconductor Industry Assoc. [Online]. Available: <http://public.itrs.net>

[2.18] C. Kretzschmar, A. K. Nieuwland and D. Muller, “Why Transition Coding for Power Minimization of on-Chip Buses does not work,” in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 512-517, Feb. 2004.

[2.19] N.K. Patel and I. L. Markov, “Error-Correction and Crosstalk Avoidance in DSM Busses”, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No.10, pp. 1076-1080, Oct. 2004.

[2.20] S. R. Sridhara, and N. R. Shanbhag, “Coding for reliable on-chip buses: fundamental limits and practical codes,” in *Proceedings of IEEE International Conference on VLSI Design*, pp. 417-422, Jan. 2005.

[2.21] F. Worm, P. Ienne, P. Thira and G. DeMicheli, “A Robust Self-Calibrating Transmission Scheme for On-Chip Networks,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 13, No. 1, pp. 126-139, Jan. 2004.

[2.22] T. Chelcea and S.M. Nowick, “Robust Interfaces for Mixed-Timing Systems,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 8, pp. 857-873, Aug. 2004.

[2.23] J. Muttersbach, T. Villiger, and W. Fichtner, “Practical design of globally asynchronous locally synchronous systems,” in *Proceeding of International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 52-59, Apr. 2000.

[2.24] J. Mekie, S. Chakraborty and D. K. Sharma, “Evaluation of pausable clocking for interfacing high speed IP cores in GALS Framework”, in *Proceeding of the 17th International Conference on VLSI Design*, pp.559-564, 2004.

[2.25] S. Lee, C. Lee and H.-J. Lee, “A new multi-channel on-chip-bus architecture for system-on-chips,” in *Proceeding of IEEE International SoC Conference*, pp. 305-308, Sep. 2004.

[2.26] M. Ariyamparambath, D. Bussaglia, B. Reinkemeier, T. Kogel and T. Kempf, “A highly efficient modeling style for heterogeneous bus architectures,” in *Proceeding of IEEE International SoC Conference*, pp. 83-87, Nov. 2003.

[2.27] A. Ivanov and G. De-Micheli, “Guest Editors’ Introduction: The Network-on-Chip Paradigm in Practice and Research,” *IEEE Design and Test of Computers*, Vol. 22, Issue 5, pp. 399-403, Sep. 2005.

[2.28] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. Meerbergen, P. Wielage and E. Waterlander, “Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip,” in *Proceeding of IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp. 350-355, 2003.

[2.29] M. Dehyadgari, M. Nickray, A. Afzali-kusha and Z. Navabi, “A new protocol stack model for network on chip,” *IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*, 2006.

[2.30] C. Nicopoulos, V. Narayanan and C.R. Das, *Network-on-Chip Architectures*, Springer, 2009.

[2.31] T. Bjerregaard and S. Mahadevan, “A Survey of Research and Practices of Network-on-Chip,” *ACM Computing Surveys*, Vol. 38, Article 1, Mar. 2006.

[2.32] I. Cidon, *Tutorial in IEEE Network-on-Chip Symposium*, May. 2009.

[2.33] M. Tudruj and L. Masko, “Dynamic SMP Clusters with Communication on the Fly in NoC Technology for Very Fine Grain Computations”, in *Proceeding of International Symposium on/Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks*, pp. 97-104, 2004.

[2.34] P.G. Paulin, C. Pilkington, E. Bensoudane, M. Langevin and D. Lyonnard, “Application of a multi-processor SoC platform to high-speed packet forwarding”, in *Proceeding of Design, Automation and Test in Europe Conference and Exhibition*, pp. 58-63, 2004.

[2.35] A. Adriahantaina, H. Charlery, A. Greiner, L. Mortiez, and C. A. Zeferino, “SPIN: a scalable, packet switched, on-chip micro-network,” in *Proceeding of Design, Automation and Test in Europe Conference and Exhibition*, pp. 70-73, 2003.

[2.36] W.J. Dally and B. Towles, “Router Packets, “Route Packets, Not Wires: On-Chip Interconnection Networks,” in *Proceeding of Design Automation Conference*, pp. 683-689, 2001.

[2.37] W.J. Dally and C.L. Seitz, "The Torus Routing Chip," *Technical Report 5208: TR: 86, Computer Science Department, California Institute of Technology*, pp. 1-19, 1986.

[2.38] F. Karim, A. Nguyen and S. Dey, "An Interconnect Architecture for Networking Systems on Chips," *IEEE Micro*, Vol. 22, No. 5, pp. 36-45, Sept./Oct. 2002.

[2.39] P. P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Design of a switch for network on chip applications," in *Proceeding of IEEE International Symposium on Circuits and Systems*, pp.217-220, 2003.

[2.40] Y.-H. Kao, N. Alfaraj, M. Yang, and H. J. Chao, "Design of High-Radix Clos Network-on-Chip," in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp. 181-188, 2010.

[2.41] F. Gilabert, S. Medardoni, D. Bertozzi, L. Benini, M.E. Gomez, P. Lopez and J. Duato, "Exploring High-Dimensional Topologies for NoC Design through an Integrated Analysis and Synthesis Framework," in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp. 107-116, 2010.

[2.42] D. Bertozzi and L. Benini, "Xpipes: a network-on-chip architecture for gigascale systems-on-chip," *IEEE Circuits and Systems Magazine*, Vol. 4, pp 18-31, 2004.

[2.43] M. Tudruj and L. Masko, "Fine-Grain Numerical Computations in Dynamic SMP Clusters with Communication on the Fly," in *Proceeding of International Conference on Parallel Computing in Electrical Engineering*, pp. 386-389, 2004.

[2.44] M. Saneei, A. Afzali-Kusha, and Z. Navabi, "Low-power and low-latency cluster topology for local traffic NoCs," in *Proceeding of IEEE International Symposium on Circuits and Systems*, pp. 1727-1730, 2006.

[2.45] S. Bourduas and Z. Zilic, "A Hybrid Ring/Mesh Interconnect for Network-on-Chip Using Hierarchical Rings for Global Routing," in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp. 195-204, 2007.

[2.46] A. Guerre, N. Ventroux, R. David and A. Merigot, "Hierarchical Network-on-Chip for Embedded Many-Core Architectures," in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp. 189-196, 2007.

[2.47] Z. Marrakchi, H. Mrabet, C. Masson, and H. Mehrez, "Mesh of Tree: Unifying Mesh and MFPGA for Better Device Performances," in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp. 243-252, 2007.

[2.48] K. Lee, S.-J. Lee and H.-J. Yoo, “Low-power network-on-chip for high-performance SoC design,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 14, pp. 148-160, 2006.

[2.49] Yole Development. (2007). 3DIC & TSV Report [Online]. [http://www.yole.fr/pagesan/products/report\\_sample/3dic.pdf](http://www.yole.fr/pagesan/products/report_sample/3dic.pdf)

[2.50] V. F. Pavlidis and E. G. Friedman, “3-D Topologies for Networks-on-Chip,” in *Proceeding of IEEE International SOC Conference*, pp. 285-288, 2006

[2.51] C. Addo-Quaye, “Thermal-aware mapping and placement for 3-D NoC designs,” in *Proceeding of IEEE International SOC Conference*, pp. 25-28, 2005.

[2.52] A. Y. Weldezion, M. Grange, D. Pamunuwa, Z. Lu, A. Jantsch, R. Weerasekera, and H. Tenhunen, “Scalability of Network-on-Chip Communication Architecture for 3-D Meshes,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 114-123, 2009.

[2.53] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan and M. Kandemir, “Design and Management of 3D Chip Multiprocessors Using Network-in-Memory,” in *Proceeding of International Symposium on Computer Architecture*, pp 130-141, 2006.

[2.54] J. Duato, S. Yalamanchili and L. Ni, *Interconnection Networks - An Engineering Approach*, Morgan Kaufmann, 2003.

[2.55] Y. Qian, Z. Lu and W. Dou, “Analysis of Worst-Case Delay Bounds for On-Chip Packet-Switching Networks,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.29, No. 5, pp.802-815, May 2010.

[2.56] I. Nousias and T. Arslan, “Wormhole Routing with Virtual Channels using Adaptive Rate Control for Network-on-Chip (NoC),” *First NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pp.420-423, 2006.

[2.57] L.I. Tabada and P.U. Tagle, “Shared Buffer Approach in Fault Tolerant Networks,” *International Conference on Computer Technology and Development*, pp.235-239, 2009.

[2.58] C. Xiao, M. Zhang, Y. Dou and Z. Zhao, “Dimensional Bubble Flow Control and Fully Adaptive Routing in the 2-D Mesh Network on Chip,” in *Proceeding of International Conference on Embedded and Ubiquitous Computing*, pp.353-358, 2008.

[2.59] A. Pullini, F. Angiolini, D. Bertozzi, and L. Benini, “Fault Tolerance Overhead in Network-on-Chip Flow Control Schemes,” in *Proceeding of Symposium on Integrated Circuits and Systems Design*, pp.224-229, 2005.

[2.60] T.-P. Lee and J. Siliquini, "Deficit round robin with hop-by-hop credit based flow control," *IEEE Region 10 Conference*, pp.1-4, 2007.

[2.61] N. Concer, L. Bononi, M. Soulie, R. Locatelli, and L. P. Carloni, "CTC: An end-to-end flow control protocol for multi-core systems-on-chip," in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp.193-202, 2009.

[2.62] F. Worm, P. Ienne, P. Thiran, and G. De-Micheli, "A Robust Self-Calibrating Transmission Scheme for On-Chip Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 13, No. 1, pp.126-138, Jan. 2005.

[2.63] E. Costa, S. Bampi, and J. Monteiro, "Power Efficient Arithmetic Operand Encoding," in *Proceeding of Symposium Integrated Circuits and Systems Design*, pp. 201-206, 2001.

[2.64] Y. Aghaghiri, F. Fallah and M. Pedram, "ALBORZ: Address Level Bus Power Optimization," in *Proceeding of International Symposium on Quality Electronics Devices*, pp. 470-475, 2002.

[2.65] P. P. Pande, A. Gangul, B. Feero, and C. Grecu, "Applicability of Energy Efficient Coding Methodology to Address Signal Integrity in 3D NoC Fabrics," in *Proceeding of IEEE International On-Line Testing Symposium*, pp. 161-166, 2007.

[2.66] J. C. S. Palma, L. S. Indrusiak, F. G. Moraes, A. Garcia Ortiz, M. Glesner, and R. A. L. Reis, "Inserting Data Encoding Techniques into NoC-Based Systems," *IEEE Computer Society Annual Symposium on VLSI*, pp. 299-304, 2007.

[2.67] P. Partha Pratim, Z. Haibo, G. Amlan, and C. Grecu, "Crosstalk-aware Energy Reduction in NoC Communication Fabrics," in *Proceeding of IEEE International SOC Conference*, pp. 225-228, 2006.

[2.68] A. Ganguly, P. Partha Pratim, B. Belzer, and C. Grecu, "Addressing Signal Integrity in Networks on Chip Interconnects through Crosstalk-Aware Double Error Correction Coding," in *IEEE Computer Society Annual Symposium on VLSI*, pp. 317-2324, 2007.

[2.69] C. Grecu, A. Ivanov, R. Saleh, E. S. Sogomonyan, and P. Partha Pratim, "On-line fault detection and location for NoC interconnects," *IEEE International On-Line Testing Symposium*, pp. 6-10, 2006.

[2.70] C. D'Alessandro, S. Delong, A. Bystrov, A. Yakovlev, and O. Maeovsky, "Multiple-rail phase-encoding for NoC," *IEEE International Symposium on Asynchronous Circuits and Systems*, pp. 10-16, 2006.

[2.71] S.R. Sridhara and Naresh R. Shanbhag, “Coding for System-on-Chip Networks: A Unified Framework,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 13, No. 6, pp.655-667, Jun. 2005.

[2.72] S. Chen and X. Liu, “A Low-Latency and Low-Power Hybrid Insertion Methodology for Global Interconnects in VDSM Designs,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp.75-82, 2007.

[2.73] P. P. Pande, H. Zhu, A. Ganguly and C. Grecu, “Design of Low power & Reliable Networks on Chip through joint crosstalk avoidance and forward error correction coding,” in *Proceeding of IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems*, pp.466-476, 2006.

[2.74] Y.-C. Lan, S.-H. Lo, Y.-C. Lin, Y.-H. Hu and S.-J. Chen, “BiNoC: A Bidirectional NoC Architecture with Dynamic Self-Reconfigurable Channel,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp.256-265, 2009.

[2.75] S. H. Lo, Y. C. Lan, H. H. Yeh, W. C. Tsai, Y. H. Hu and S. J. Chen, “QoS Aware BiNoC Architecture,” in *Proceeding of IEEE International Parallel & Distributed Processing Symposium*, pp.1-10, Apr. 2010.

[2.76] A. Ejlali and B. M. Al-Hashimi, “SEU-Hardened Energy Recovery Pipelined Interconnects for On-Chip Networks,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 67-76, 2008.

[2.77] B. Fu, D. Wolpert and P. Ampadu, “Lookahead-Based Adaptive Voltage Scheme for Energy-Efficient On-Chip Interconnect Links,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 54-63, 2009.

[2.78] S.-J. Lee, K. Kim, H. Kim, N. Cho, and H.-J. Yoo, “Adaptive network-on-chip with wave-front train serialization scheme,” in *Proceeding of IEEE symposium on VLSI Circuits*, pp. 104-107, 2005.

[2.79] M. Kinsky, M. H. Cho, T. Wen, G. E. Suh, M. van Dijk and S. Devadas, “Application-Aware Deadlock-Free Oblivious Routing,” in *Proceedings of the Int'l Symposium on Computer Architecture* , pp. 208-219, June 2009.

[2.80] M. Lis, M. H. Cho, K. S. Shim, and S. Devadas, “Path-Diverse Inorder Routing,” in *Proceedings of the International Conference on Green Circuits and Systems*, pp. 311-316, June 2010.

[2.81] M. Majer, C. Bobda, A. Ahmadiania, and J. Teich, “Packet Routing in Dynamically Changing Networks on Chip,” in *Proceeding of IEEE International Parallel and Distributed Processing Symposium*, pp. 154b, 2005.

[2.82] G. Michelogiannakis, D. Pnevmatikatos, and M. Katevenis, “Approaching Ideal NoC Latency with Pre-Configured Routes,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp. 153-162, 2007.

[2.83] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, “Routing Table Minimization for Irregular Mesh NoCs,” *Design, Automation & Test in Europe Conference & Exhibition*, pp. 1-6, 2007.

[2.84] V. Catania, R. Holzman, S. Kumar, and M. Palesi, “A methodology for design of application specific deadlock-free routing algorithms for NoC systems,” in *Proceeding of International conference on Hardware/software codesign and system synthesis*, pp. 142-147, 2006.

[2.85] M. Dehyadgari, M. Nickray, A. Afzali-kusha, and Z. Navabi, “Evaluation of pseudo adaptive XY routing using an object oriented model for NOC,” *International Conference on Microelectronics*, pp. 5-8, 2005.

[2.86] A. Mello, L. Moller, N. Calazans, and F. Moraes, “MultiNoC: a multiproCESSing system enabled by a network on chip,” in *Proceeding of Design, Automation & Test in Europe Conference & Exhibition*, pp. 234-239 Vol. 233, 2005.

[2.87] M. Palesi, G. Longo, S. Signorino, R. Holzman, S. Kumar and V. Catania, “Design of Bandwidth Aware and Congestion Avoiding Efficient Routing Algorithm for Network-on-Chip Platforms,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 97-106, 2008.

[2.88] P. Dongkook, C. Nicopoulos, K. Jongman, N. Vijaykrishnan, and C. R. Das, “A Distributed Multi-Point Network Interface for Low-Latency, Deadlock-Free On-Chip Interconnects,” *International Conference on Nano-Networks and Workshops*, pp.1-6, 2006.

[2.89] U. Y. Ogras, R. Marculescu, L. Hyung Gyu, and C. Naehyuck, “Communication architecture optimization: making the shortest path shorter in regular networks-on-chip,” in *Proceeding of Design, Automation & Test in Europe Conference & Exhibition*, pp. 6-9, 2006

[2.90] A. Sobhani, M. Daneshthalab, M. H. Neishaburi, M. D. Mottaghi, A. Afzali-Kusha, O. Fatemi, and Z. Navabi, “Dynamic routing algorithm for avoiding hot spots in on-chip networks,” in *Proceeding of International Conference on Design and Test of Integrated Systems in Nanoscale Technology*, pp. 179-183, 2006.

[2.91] H. Barati, A. Movaghar, A. Barati, A.A. Mazreah, “Routing Algorithms Study and Comparing in Interconnection Networks,” in *Proceeding of International Conference on Information and Communication Technologies*, pp.1-5, 2008.

[2.92] J. Duato, O. Lysne, R. Pang, and T. M. Pinkston, “A theory for deadlock-free dynamic network reconfiguration. Part I,” *IEEE Transactions on Parallel and Distributed Systems* , Vol. 16, pp. 412-427, 2005.

[2.93] M. Palesi, K. Shashi, R. Holsmark, and V. Catania, “Exploiting Communication Concurrency for Efficient Deadlock Free Routing in Reconfigurable NoC Platforms,” *IEEE International Parallel and Distributed Processing Symposium*, pp. 1-8, 2007.

[2.94] A. Giuseppe, C. Vincenzo, P. Maurizio, and P. Davide, “Neighbors-on-Path: A New Selection Strategy for On-Chip Networks,” in *Proceeding of IEEE/ACM/IFIP Workshop Embedded Systems for Real Time Multimedia*, pp. 79-84, 2006.

[2.95] M. Daneshtalab, A. Sobhani, A. Afzali-Kusha, O. Fatemi and Z. Navabi, “NoC Hot Spot minimization Using AntNet Dynamic Routing Algorithm,” in *Proceeding of International Conference on Application-specific Systems, Architectures and Processors*, pp. 33-38, 2006.

[2.96] J. Flich, S. Rodrigo and J. Fuato, “An Efficient Implementation of Distributed Routing Algorithm for NoCs,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 87-96, 2008.

[2.97] X. Duan, D. Zhang and X. Sun, “Routing Scheme of an Irregular Mesh-Based NoC,” in *Proceeding of International Conference on Networks Security, Wireless Communications and Trusted Computing*, pp. 572-575, 2009.

[2.98] R. Holsmark, S. Kumar, M. Palesi and A. Mejia, “HiRA: A Methodology for Deadlock Free Routing in Hierarchical Networks on Chip,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 2-11, 2009.

[2.99] Z. Song, G. Ma and D. Song, “Hierarchical Star: An Optimal NoC Topology for High-Performance SoC Design,” in *Proceeding of International Multi-symposiums on Computer and Computational Sciences*, pp 2-11, 2009.

[2.100] A. Kohler and M. Radetzki, “Fault-Tolerant Architecture and Deflection Routing for Degradable NoC Switches,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 22-31, 2009.

[2.101] Y.-C. Lan, M.-C. Chen, A.-P. Su, Y.-H. Hu and S.-J. Chen, “Fluidity Concept for NoC: A Congestion Avoidance and Relief Routing Scheme,” in *Proceeding of IEEE International SOC Conference*, pp 65-70, Sept. 2008.

[2.102] W. Song, D. Edwards, J.L. Nunez-Yanez and S. Dasgupta, “Adaptive Stochastic Routing in Fault-tolerant On-chip Networks,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 32-37, 2009.

[2.103] C.-H. Chao, K.-Y. Jheng, H.-Y. Wang, J.-C. Wu, and A.-Y. Wu, “Traffic- and Thermal-Aware Run-Time Thermal Management Scheme for 3D NoC Systems,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 223-230, 2010.

[2.104] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, N. Vijaykrishnan, M. S. Yousif and C. R. Das, "A Novel Dimensionally-Decomposed Router for On-Chip Communication in 3D Architecture," in *Proceedings of the International Symposium on Computer Architecture*, pp. 138-149, 2009.

[2.105] H. Matsutani, M. Koibuchi and H. Amano, "Tightly-Coupled Multi-Layer Topologies for 3-D NoCs," in *Proceeding of International Conference on Parallel Processing*, pp. 75-84, 2007.

[2.106] J. Kim, "Low-Cost Router Microarchitecture for On-Chip Networks," in *Proceeding of IEEE/ACM International Symposium on Microarchitecture (MICRO-42)*, pp. 255-266, 2009.

[2.107] A. Ruadulescu, K. Goossens, G. De-Micheli, S. Murali, and M. Coenen, "A buffer-sizing algorithm for networks on chip using TDMA and credit-based end-to-end flow control," in *Proceeding of International Conference on Hardware/software codesign and system synthesis*, pp. 130-135, 2006.

[2.108] T.D. Richardson, C. Nicopoulos, D. Park, V. Narayanan, X. Yuan, C. Das and V. Degalahal, "A hybrid SoC interconnect with dynamic TDMA-based transaction-less buses and on-chip networks," in *Proceeding of International Conference on VLSI Design*, pp. 8-pp.15, 2006.

[2.109] T. Marescaux, B. Bricke, P. Debacker, V. Nollet, and H. Corporaal, "Dynamic time-slot allocation for QoS enabled networks on chip," *Workshop on Embedded Systems for Real-Time Multimedia*, pp. 47-52, 2005.

[2.110] X. Gao, Z. Zhang and X. Long, "Round Robin Arbiters for Virtual Channel Router," *International Multi-conference on Computational Engineering in Systems Applications*, pp. 1610-1614, 2006.

[2.111] Y.-L. Lee, J.-M. Jou and Y.-Y. Chen, "A High-Speed and Decentralized Arbiter Design for NoC," in *Proceeding of International Conference on Computer Systems and Applications*, pp. 350-353, 2009.

[2.112] L.-Y. Lin, C.-Y. Wang, P.-J. Huang, C.-C. Chou, J.-Y. Jou, "Communication-driven task binding for multiprocessor with latency insensitive network-on-chip," in *Proceeding of IEEE Asia and South Pacific Design Automation Conference*, pp. 39-44, 2005.

[2.113] M. Lai, Z. Wang, L. Gao, H. Lu and K. Dai "A Dynamically-Allocated Virtual Channel Architecture with Congestion Awareness for On-Chip Routers," in *Proceeding of Design Automation Conference*, pp. 630-633, Jun. 2008.

[2.114] M. Lai, L. Gao, W. Shi and Z. Wang, "Escaping from Blocking A Dynamic Virtual Channel for Pipelined Routers," in *Proceeding of International Conference Complex, Intelligent and Software Intensive System*, pp.795-800, 2008.

[2.115] H. Zimmer, S. Zink, T. Hollstein and M. Glesner, “Buffer-Architecture Exploration for Routers in a Hierarchical Network-on-Chip,” in *Proceeding of IEEE International Parallel and Distributed Processing Symposium*, pp.171a, 2005.

[2.116] C.A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M.S. Yousif and C.R. Das, “ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers,” in *Proceeding of IEEE/ACM International Proceeding Microarchitecture*, pp. 333-346, 2006.

[2.117] M.A.J. Jamali and A. Khademzadeh, “A New Method for Improving the Performance of Network on Chip using DAMQ Buffer Schemes,” in *Proceeding of International Conference Application Information Communication Technology*, pp. 1-6, 2009.

[2.118] J. Liu and J. G. Delgado-Frias, “A Shared Self-Compacting Buffer for Network-On-Chip Systems,” in *Proceeding of IEEE International Midwest Symposium on Circuits and System*, pp. 26-30, 2006.

[2.119] J. Hu, U.Y. Ogras and R. Marculescu, “Application-specific buffer space allocation for networks-on-chip router design”, in *Proceeding of IEEE/ACM International Conference of Computer Aided Design*, pp. 354-361, 2004.

[2.120] T.-C. Huang, U.Y. Ogras and R. Marculescu, “Virtual Channels Planning for Network-on-Chip,” in *Proceeding of International Symposium on Quality Electronic Design*, pp. 879-884, 2007.

[2.121] N. Concer, L. Bononi, M. Soulie, R. Locatelli and L. P. Carloni, “The Connection-Then-Credit Flow Control Protocol for Heterogeneous Multicore System-on-Chip,” *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 29, No. 6, pp. 869-882, Jun. 2010.

[2.122] A. Radulescu, J. Dielissen, K. Goossens, E. Rijpkema and P. Wielage, “An Efficient On-Chip Network Interface Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Configuration,” in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 1-6, Mar. 2004.

[2.123] F. Clermidy, R. Lemaire, Y. Thonnart and P. Vivet, “A Communication and Configuration Controller for NoC based Reconfigurable Data Flow Architecture,” in *Proceeding of ACM/IEEE International Symposium Networks-on-Chip*, pp. 153-162, May. 2009.

[2.124] Y.-L. Lai, S.-W. Yang, M.-H. Sheu, H.-Y. Tang and P.-Z. Huang, “A High-Speed Network Interface Design for Packet-Based NoC,” in *Proceeding of IEEE International Conference on Communication, Circuits and Systems*, pp. 2667-2671, 2006.

[2.125] P. S. Bhojwani, R. Mahapatra, E. J. Kim and T. Chen, “A Heuristic for Peak Power Constrained Design of Network on Chip (NoC) based Multimode Systems,” in *Proceeding of International Conference on VLSI Design*, pp. 124-129, 2005.

[2.126] T.-T. Ye, L. Benini and G. De-Michelis, “Analysis of power consumption on switch fabrics in network routers,” in *Proceeding of IEEE Design and Automation Conference*, pp.524-529, 2002.

[2.127] J. Kim, D. Park, C. Nicopoulos, N. Vijaykrishnan, C. R. Das, “Design and Analysis of an NoC Architecture from Performance, Reliability and Energy Perspective,” in *Proceeding of IEEE Symposium on Architecture for Networking and Communication Systems*, pp.173-182, 2005.

[2.128] S.-J. Lee, K. Lee, H.-J. Yoo, “Analysis and Implementation of Practical Cost-Effective Network-on-Chips,” *IEEE Design & Test of Computers Magazine*, pp. 422-433, 2005.

[2.129] A. Banerjee, R. Mullins and S. Moore, “A Power and Energy Exploration of Network-on-Chip Architectures,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp 163-172, 2007.

[2.130] S. Penolazzi and A. Jantsch, “A High Level Power Model for the Nostrum NoC,” in *Proceeding of EUROMICRO Conference on Digital System Design*, pp. 673-676, 2006.

[2.131] H.-S. Wang, L.-S. Peh and S. Malik, “A Power Model for Routers: Modeling Alpha 21364 and InfiniBand Routers,” *IEEE Micro*, Vol. 23, No. 1, pp. 26-35, 2003.

[2.132] A. B. Kahng., B. Li, L.-S. Peh and K. Samadi, “ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage,” in *Proceeding of IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp. 423-429, Mar. 2009.

[2.133] N. Banerjee, P. Vellanki and K. S. Chatha, “A Power and Performance Model for Network-on-Chip Architectures,” in *Proceeding of IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp. 1250-1255, Mar. 2004.

[2.134] S.-E. Lee and N. Bagherzadeh, “A high level power model for Network-on-Chip (NoC) router,” *Elsevier Journal of Computer and Electrical Engineering*, pp. 837-845, Jan. 2009.

[2.135] Y. Hu, Y. Zhu, H. Chen, R. Graham and C.-K. Cheng, “Communication Latency Aware Low Power NoC Synthesis,” in *Proceeding of IEEE Design and Automation Conference*, pp.574-579, 2006.

[2.136] T. Simunic, S.P. Boyd and P. Glynn, “Managing power consumption in networks on chips,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 1, pp.96-107, Jan. 2004.

[2.137] C. Ciordas, A. Hansson, K. Goossens, and T. Basten, “A monitoring-aware network-on-chip design flow,” in *Proceeding of EUROMICRO Conference on Digital System Design*, pp. 97-106, 2006.

[2.138] J. M. Rabaey, “Scaling the power wall: Revisiting the low-power design rules,” *Keynote speech at Symposium on SoC*, Nov. 2007.

[2.139] P. Bhojwani, J.-D. Lee and Rabi Mahapatra, "SAPP: Scalable and Adaptable Peak Power Management in NoCs," in *Proceeding of International Symposium on Low Power Electronic Devices*, pp.340-345, 2007.

[2.140] G. Liang, P. Liljeberg, E. Nigussie and H. Tenhunen, “A Review of Dynamic Power Management Methods in NoC under Emerging Design Considerations,” in *Proceeding of IEEE NORACHIP*, pp. 1-6, Nov. 2009.

[2.141] G. Liang and A. Jantsch, “Adaptive power management for the on-chip communication network,” in *Proceeding of EUROMICRO Conference on Digital System Design*, pp. 649-656, 2006.

[2.142] S. Moore, G. Taylor, R. Mullins and P. Robinson, “Point to Point GALS Interconnect,” in *Proceeding of International Symposium on Asynchronous Circuits and Systems*, pp. 69-75, April 2002.

[2.143] J. Muttersbach, T. Villiger, and W. Fichtner, “Practical Design of Globally-Asynchronous Locally-Synchronous Systems,” in *Proceeding of International Symposium on Asynchronous Circuits and Systems*, pp.52-59, 2000.

[2.144] E. Amini, M. Najibi, and H. Pedram, “Globally Asynchronous Locally Synchronous Wrapper Circuit based on Clock Gating,” in *Proceeding of Emerging VLSI Technologies and Architectures*, 2006.

[2.145] D. Rostislav, R. Ginosar, and Christos P. Sotiriou, “High Rate Data Synchronization in GALS SoCs,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 14, No. 10, pp. 1063-1074, Oct. 2006.

[2.146] J. Mekie, S. Chakraborty, and D.K. Sharma, “Evaluation of Pausible Clocking for Interfacing High Speed IP Cores in GALS Framework,” in *Proceeding of International Conference on VLSI Design*, pp. 559-564, 2004.

[2.147] D. Kim, K. Kim, J.-H. Kim, S.-J. Lee and H.-J. Yoo, “Solutions for Real Chip Implementation Issues of NoC and Their Application to Memory-Centric NoC,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp. 30-39, 2007.

[2.148] M. N. Bojnordi, N. M. Madani, M. Semsarzade, and A. Afzali-Kusha, “An Efficient Clocking Scheme for On-Chip Communications,” in *Proceeding of IEEE Asia Pacific Conference on Circuits and Systems*, pp. 119-122, 2006.

[2.149] R. Dobkin, R. Ginosar, and I. Cidon, “QNoC Asynchronous Router with Dynamic Virtual Channel Allocation,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp. 218-218, 2007.

[2.150] I. M. Panades, A. Greiner, and A. Sheibanyrad, “A Low Cost Network-on-Chip with Guaranteed Service Well Suited to the GALS Approach,” in *Proceeding of International Conference on Nano-Networks and Workshops*, pp. 1-5, 2006.

[2.151] T. Chelcea and S.M. Nowick, “Robust Interfaces for Mixed-Timing Systems,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 8, pp. 857-873, Aug. 2004.

[2.152] I. M. Panades and A. Greiner, “Bi-Synchronous FIFO for Synchronous Circuit Communication Well Suited for Network-on-Chip in GALS Architectures,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp.83-94, 2007.

[2.153] T. Ono and M. Greenstreet, “A Modular Synchronizing FIFO for NoCs,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp.224-233, 2009.

[2.154] T. Villiger, H. Kaslin and W. Fichtner, “Self-Timed Ring for Globally Asynchronous Locally Synchronous Systems”, in *Proceeding of International Symposium on Asynchronous Circuits and Systems*, pp. 141-150, 2003.

[2.155] U.Y. Ogras, R. Marculescu, P. Choudhary and D. Marculescu, “Voltage-Frequency Island Partitioning for GALS-based Networks-on-Chip,” in *Proceeding of IEEE/ACM Design Automation Conference*, pp. 110-115, June 2007.

[2.156] U.Y. Ogras, R. Marculescu, and D. Marculescu, “Variation-Adaptive Feedback Control for Network-on-Chip with Multiple Clock Domain,” in *Proceeding of IEEE/ACM Design Automation Conference*, p.p. 614-619, June 2008.

[2.157] E. Beignre, F. Clermidy, H. Lhermet, S. Miermont, Y. Thonnart, X.-T. Tran, A. Valentian, D. Varreau, P. Vivet, X. Popon and H. Lebreton, “An Asynchronous Power Aware and Adaptive NoC Based Circuit,” *IEEE Journal of Solid-State Circuits*, Vol. 44, No. 4, pp. 1167–1177, Apr. 2009.

[2.158] E. Beignre, F. Clermidy, S. Miermont, and P. Vivet, “ Dynamic Voltage and Frequency Scaling Architecture for Unit Integration within a GALS NoC,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp.129-138, 2008.

[2.159] I. Miro-Panades, F. Clermidy, P. Vivet and A. Greiner, “Physical Implementation of the DSPIN Network-in-Chip in the FAUST Architecture,” *in Proceeding of IEEE International Symposium on Networks-on-Chip*, pp.139-148, 2008.

[2.160] N. Dutt, “Memory-Aware NoC Exploration and Design,” *in Proceeding of IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp. 1128-1129, 2008.

[2.161] Krste Asanovic, “Memory is Network,” *in Panel Discussion of IEEE International Symposium on Networks-on-Chip*, 2009.

[2.162] D. Kim, K. Kim, J.-K. Kim, S. Lee and H.-J. Yoo, “Memory-Centric Networks-on-Chip for Power Efficient Execution of task-level pipeline on a multi-core processor,” *IET Computers & Digital Techniques*, pp. 513-524. 2009.

[2.163] D. Benitez, J. C. Moure, D. Rexachs and E. Luque, “A reconfigurable cache memory with heterogeneous banks,” *in Proceeding of IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp.825-830, 2010.

[2.164] N. Hardavellas, I. Pandis, R. Johnson, N. Mancheril, A. Ailamaki and B. Falsafi, “Database Servers on Chip Multiprocessors: Limitations and Opportunities,” *in Proceeding of Conference on Innovative Data System Research*, pp.79-87, 2007.

[2.165] P. Ranganathan, S. Adve, and N.P. Jouppi, “Reconfigurable Caches and their Application to Media Processing,” *in Proceeding of International Symposium on Computer Architecture*, pp.214-224, 2000.

[2.166] D.H. Albonesi, “Selective cache ways: on-demand cache resource allocation,” *in Proceeding of International Symposium on Microarchitecture*, pp.248-259, 1999.

[2.167] C. Zhang, F. Vahid, W. Najjar, “A Highly Configurable Cache Architecture for Embedded Systems”, *in Proceeding of International Symposium on Computer Architecture*, 136-146, 2003.

[2.168] S.-H. Yang, M. D. Powell, B. Falsafi, K. Roy, and T. N. Vijaykumar, “An integrated circuit/architecture approach to reducing leakage in deep-submicron high-performance i-caches,” *in Proceeding of IEEE International Symposium on High-Performance Computer Architecture*, pp. 147-157, 2002.

[2.169] S. Yang, M. Powell, B. Falsafi and T.N. Vijaykumar, “Exploiting Choice in Resizable Cache Design to Optimize Deep-Submicron Processor Energy-Delay”, *in Proceeding of IEEE International Symposium on High-Performance Computer Architecture*, pp.151-161, 2002.

- [2.170] R. Iyer, “CQoS: a framework for enabling QoS in shared caches of CMP platforms,” in *Proceeding of Annual International Conference on Supercomputing*, pp. 257-266, 2004.
- [2.171] K. Varadarajan, S.K. Nandy, V. Sharda, A. Bharadwaj, R. Iyer, S. Makineni and D. Newell, “Molecular Caches: A caching structure for dynamic creation of application-specific heterogeneous cache regions,” in *Proceeding of International Symposium on Microarchitecture*, pp.433-442, 2006.
- [2.172] D. Kaseridis, J. Stuecheli, L.K. John, “Bank-aware Dynamic Cache Partitioning for Multicore Architectures,” in *Proceeding of International Conference on Parallel Processing*, pp.18-25, 2009.

## References of Chapter 3

- [3.1] L. Benini and G. De Micheli, *Network on Chips: Technology and Tools*, Morgan Kaufmann, 2006.
- [3.2] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufmann, 2004.
- [3.3] V. Raghunathan, M.B. Srivastava, and R.K. Gupta, “A Survey of Techniques for Energy Efficient On-Chip Communication,” in *Proceeding of IEEE/ACM Design and Automation Conference*, pp. 900–905, Jun. 2003.
- [3.4] R. Marculescu, U.Y. Ogras, L.-S. Peh, N.E. Jerger, and Y. Hoskote, “Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 28, No. 1, pp. 3–21, Jan. 2009.
- [3.5] H. Lekatsas and J. Henkel, “ETAM++: Extended Transition Activity Measure for Low Power Address Bus Designs,” in *Proceeding of VLSI Design Conference*, pp. 113–120, 2002.
- [3.6] K.-H. Baek, K.-W. Kim, and S.M. Kang, “A Low Energy Encoding Technique for Reduction of Coupling Effects in SoC Interconnects,” in *Proceeding of IEEE International Midwest Symposium on Circuits and System*, Vol. 1, pp. 80–83, 2000.
- [3.7] C.-G. Lyuh and T.-W. Kim, “Low Power Bus Encoding with Crosstalk Delay Elimination,” in *Proceeding of International ASIC/SoC Conference*, pp. 389–393, 2002.
- [3.8] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf, “An Adaptive Dictionary Encoding Scheme for SOC Data Buses,” in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 1059–1064, 2002.

[3.9] K.-M. Lee, S.-J. Lee, and H.-J. Yoo, “Low-Power Network-on-Chip for High Performance SoC Design,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 14, No. 2, pp. 148–160, Feb. 2006.

[3.10] J. Yang and R. Gupta, “FV Encoding for Low-Power Data I/O,” in *Proceeding of IEEE International Symposium on Low Power Electronic and Design*, pp. 84–87, 2001.

[3.11] R.-B. Lin, “Inter-Wire Coupling Reduction Analysis of Bus-Invert Coding,” *IEEE Transactions on Circuits and Systems—I*, Vol. 55, No.7, pp.1911–1920, Aug. 2008.

[3.12] C.S. D’Alessandro, D. Shang, A. Bystrov, A.V. Yakovlev, and O. Maevsky, “Phase-Encoding for On-Chip Signalling,” *IEEE Transactions on Circuits and Systems—I*, Vol. 55, No.2, pp.535–545, Mar. 2008.

[3.13] G. Chen, S. Duvall, and S. Nooshabadi, “Analysis and design of memoryless interconnect encoding scheme,” in *Proceeding of IEEE International Symposium on Circuits and Systems*, pp. 2990–2993, May. 2009.

[3.14] B. Fu and P. Ampadu, “On Hamming Product Codes with Type-II Hybrid ARQ for On-Chip Interconnects,” *IEEE Transactions on Circuits and Systems—I*, Vol. 56, No.9, pp.2042–2054, Sept. 2009.

[3.15] S.R. Sridhara, and N.R. Shanbhag, “Coding for System-on-Chip Networks: A Unified Framework,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 13, No. 8, pp. 655–667, Jun. 2005.

[3.16] S.R. Sridhara, and N.R. Shanbhag, “Coding for Reliable On-Chip Buses: A Class of Fundamental Bounds and Practical Codes,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.26, No. 5, pp. 977–582, May. 2007.

[3.17] K.N. Patel and I.L. Markov, “ Error-Correction and Crosstalk Avoidance in DSM Buses,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 10, pp. 1076–1080, Oct. 2004.

[3.18] P.P. Panda, A. Ganguly, B. Feero, B. Belzer, and C. Grecu, “Design of Low power & Reliable Networks on Chip through Joint Crosstalk Avoidance and Forward Error Correction Coding,” in *Proceeding of IEEE International Symposium on Defect and Fault-Tolerance VLSI Systems*, pp. 466–476, Oct. 2006.

[3.19] A. Ganguly, P.P. Panda, and B. Belzer, “Crosstalk-Aware Channel Coding Schemes for Energy Efficient and Reliable NOC Interconnects,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 17, No. 11, pp. 1626–1639, Oct. 2004.

[3.20] S.R. Sridhara, and N.R. Shanbhag, “Coding for Reliable On-Chip Buses: Fundamental Limits and Practical Codes”, in *Proceeding of IEEE International Conference on VLSI Design*, pp. 417–422, Jan. 2005.

[3.21] F. Worm, P. Ienne, P. Thiran, and G. De Micheli, “A Robust Self-Calibrating Transmission Scheme for On-Chip Networks,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 13, pp. 126–139, Jan. 2005.

[3.22] F. Worm, P. Thiran, G. De Micheli, and P. Ienne, “Self-Calibrating Networks-On-Chip,” in *Proceeding of IEEE International Symposium on Circuits and Systems*, Vol. 3, pp. 2361–2364, May 2005.

[3.23] M. Simone, M. Lajolo, and D. Bertozzi, “Variation Tolerant NoC Design by Means of Self-Calibrating Links,” in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 1402–1407, 2008.

[3.24] R. Ho, “On-Chip Wires: Scaling and Efficiency,” Ph.D. dissertation, Stanford University, Aug. 2003.

[3.25] R. Ho, K. Mai, and M. Horowitz, “Efficient On-Chip Global Interconnects,” in *Proceedings of IEEE Symposium on VLSI Circuits*, pp. 271–274, 2003.

[3.26] P.P. Sotiriadis and A.P. Chandrakasan, “A Bus Energy Model for Deep Submicron Technology,” *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, Vol. 10, No. 3, pp.341–350, Jun. 2002.

[3.27] K.-W. Kim, K.-H. Baek, N. Shanbhag, C.-L. Liu, and S.-M. Kang, “Coupling-Driven Signal Encoding Scheme for Low-Power Interface Design”, in *Proceeding of International Conference on Computer-Aided Designs*, pp. 318–321, 2000.

[3.28] P.P. Sotiriadis, “Interconnect Modeling and Optimization in Deep Submicron Technologies,” Ph.D. dissertation, Massachusetts Institute Technology, Cambridge, May 2002.

[3.29] R. Pendurkar, A. Chatterjee, and Y. Zorian, “Switching Activity Generation with Automated BIST Synthesis for Performance Testing of Interconnects,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 20, No. 9, pp. 1143–1158, Sept. 2001.

[3.30] K. Sekar and S. Dey, “LI-BIST: A Low-Cost Self-Test Scheme for SoC Logic Cores and Interconnects,” in *Proceeding of IEEE VLSI Test Symposium*, pp.417–422, 2002.

[3.31] B. Xiaoliang, et al., “Self-Test Methodology for At-Speed Test of Crosstalk in Chip Interconnects,” in *Proceeding of IEEE Design and Automation Conference*, pp.619–624, Jun. 2000.

[3.32] R. Tamhankar, S. Murali, S. Stergiou, A. Pullini, F. Angiolini, L. Benini, and G. De Micheli, “Timing-Error-Tolerant Network-on-Chip Design Methodology,”

*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 26, No. 7, pp. 1297–1310, July 2007.

- [3.33] Yi Zhao, S. Dey, and Li Chen, “Double Sampling Data Checking Technique: An Online Testing Solution for Multi-Source Noise-Induced Errors on On-Chip Interconnects and Buses,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No.7, pp. 746–755, July 2004.
- [3.34] D.N. Truong, W.H. Cheng, T. Mohsenin, Yu Zhiyi, A.T. Jacobson, G. Landge, M.J. Meeuwsen, C. Watnik, A.T. Tran, X. Zhibin, E.W. Work, J.W. Webb, P.V. Mejia, and B.M. Baas, “A 167-Processor Computational Platform in 65nm CMOS,” *IEEE Journal of Solid-State Circuits*, Vol. 44, No.4, pp. 1–15, Apr. 2009.
- [3.35] S.R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, “An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS,” *IEEE Journal of Solid-State Circuits*, Vol. 43, No.1, pp. 29–41, Jan. 2008.
- [3.36] M.A. Anders, H. Kaul, S.K. Hsu, A. Agarwal, S.K. Mathew, F. Sheikh, R.K. Krishnamurthy, and S. Borkar, “A 4.1Tb/s Bisection-Bandwidth 560Gb/s/W Streaming Circuit-Switched 8×8 Mesh Network-on-Chip in 45nm CMOS,” in *Proceeding of IEEE International Solid-State Circuits Conference*, pp. 110–112, Feb. 2010.

## References of Chapter 4

- [4.1] L. Benini and G. De Micheli, *Network on Chips: Technology and Tools*, Morgan Kaufmann, 2006.
- [4.2] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufmann, 2004.
- [4.3] Y. Qian, Z. Lu, W. Dou and Q. Dou, “Analyzing Credit-Based Router-to-Router Flow Control for On-Chip Networks,” *IEICE Transaction on Electronics*, Vol. E92-C, No. 10, Oct. 2009.
- [4.4] R. Beidas and Z. Jianwen, “A Queuing-Theoretic Performance Model for Context-Flow System-on-Chip Platforms,” In *Proceeding of Workshop of Embedded System for Real-Time Multimedia*, pp.21–26, 2004.
- [4.5] Y. Qian, Z. Lu, and W. Dou, “Worst-Case Flit and Packet Delay Bounds in Workhole Networks on Chip,” *IEICE Transactions Fundamentals*, Vol. E92-A, No. 12, Dec. 2009.

[4.6] J. Kim, “Low-Cost Router Microarchitecture for On-Chip Networks,” in *Proceeding of IEEE/ACM International Symposium on Microarchitecture (MICRO-42)*, pp. 255–266, 2009.

[4.7] G. Varatkar and R. Marculescu, “Traffic Analysis for On-Chip Networks Design of Multimedia Applications,” in *Proceeding of Design Automation Conference*, pp. 795–800, Jun. 2002.

[4.8] J. Hu, U.Y. Ogras and R. Marculescu, “Application-Specific Buffer Space Allocation for Networks-on-Chip Router Design”, in *Proceeding of IEEE/ACM International Conference of Computer-Aided Design*, pp. 354–361, 2004.

[4.9] J. Hu, U.Y. Ogras and R. Marculescu, “System-Level Buffer Allocation for Application-Specific Network-on-Chip Router Design,” *IEEE Transactions Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, No. 12, pp. 2919–2933, Dec. 2007.

[4.10] M. Coenen, S. Murali, A. Radulescu and K. Goossens, “A Buffer-Sizing Algorithm for Networks on Chip using TDMA and Credit-Based End-to-End Flow Control,” in *Proceeding of IEEE International Conference on Hardware/Software Codesign and Syst. Synthesis*, pp. 130–135, 2006.

[4.11] W.J. Dally and B. Towles, “Route Packets, not Wires: On-Chip Interconnection Network,” in *Proceeding of Design Automation Conference*, pp. 684–689, Jun. 2001.

[4.12] A. K. Kodi, A. Sarathy and A. Louri, “iDEAL: Inter-Router Dual-Function Energy and Area-Efficient Links for Network-on-Chip (NoC),” in *Proceeding of International Symposium on Computer Architecture*, pp. 241–250, 2008.

[4.13] D. Kim, K. Kim, J.-Y. Kim, S. Lee and H.-J. Yoo, “Memory-Centric Network-on-Chip for Power Efficient Execution of Task-Level Pipeline on a Multi-Core Processor,” *IET Computer & Digital Technology*, Vol. 3, iss. 5, pp. 513–524, 2009.

[4.14] T.-C. Huang, U.Y. Ogras and R. Marculescu, “Virtual Channels Planning for Network-on-Chip,” in *Proceeding of International Symposium on Quality Electronic Design*, pp. 879–884, 2007.

[4.15] J. Park, B.W. Okrafka, S. Vassiliadis and J. Delgado-Frias, “Deign and Evaluation of a DAMQ Multiprocessor Network with Self-Compacting Buffers,” in *Proceedings of Supercomputing*, pp. 713–722, 1994.

[4.16] N. Ni, M. Pirvu and L. Bhuyan, “Circular Buffered Switch Design with Wormhole Routing and Virtual Channels,” in *Proceeding of IEEE International Conference on Computer-Aided Design*, pp. 466–473, 1998.

[4.17] J. Liu and J. G. Delgado-Frias, “A Shared Self-Compacting Buffer for Network-On-Chip Systems,” in *Proceeding of IEEE International Midwest Symposium on Circuits and Systems*, pp. 26–30, 2006.

[4.18] M.A.J. Jamali and A. Khademzadeh, “A New Method for Improving the Performance of Network on Chip using DAMQ Buffer Schemes,” in *Proceeding of International Conference on Application of Information and Communication Technologies*, pp. 1–6, 2009.

[4.19] C.A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M.S. Yousif and C.R. Das, “ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers,” in *Proceeding of IEEE/ACM International Symposium on Microarchitecture (MICRO-39)*, pp. 333-346, 2006.

[4.20] M. Lai, Z. Wang, L. Gao, H. Lu and K. Dai “A Dynamically-Allocated Virtual Channel Architecture with Congestion Awareness for On-Chip Routers,” in *Proceeding of Design Automation Conference*, pp. 630–633, Jun. 2008.

[4.21] M. Lai, L. Gao, W. Shi and Z. Wang, “Escaping from Blocking A Dynamic Virtual Channel for Pipelined Routers,” in *Proceeding of International Conference on Complex, Intelligent and Software Intensive Systems*, pp.795–800, 2008.

[4.22] M.H. Neishaburi and Z. Zilic, “Reliability Aware NoC Router Architecture Using Input Channel Buffer Sharing,” in *Proceeding of 19th ACM Great Lakes symposium on VLSI*, pp. 511–516, 2009.

[4.23] A.K. Kodi, A. Sarathy, A. Louri and J. Wang, “Adaptive Inter-router Links for Low-Power, Area-Efficient and Reliable Network-on-Chip (NoC) Architectures,” in *Proceeding of Asia and South Pacific Design Automation Conference*, pp. 1–6, 2009.

[4.24] R.S. Ramanujam, V. Soteriou, B. Lin and L.-S. Peh, “Design of a High-Throughput Distributed Shared-Buffer NoC Router,” in *Proceeding of IEEE International Symposium on Networks-on-Chip*, pp.69-78, 2010.

[4.25] L. Wang, J. Zhang, X. Yang and D. Wen, “Router with Centralized Buffer for Network-on-Chip,” in *Proceeding of 19th ACM Great Lakes symposium on VLSI*, pp. 469–474, 2009.

[4.26] V. Soteriou, R.S. Ramanujam, B. Lin and L.-S. Peh, “A High-Throughput Distributed Shared-Buffer NoC Router,” *IEEE Computer Architecture Letters*, Vol. 8, No. 1, Jan.-Jun. 2009.

[4.27] W. J. Dally, and C. L. Seitz, “The Torus Routing Chip,” *Journal of Distributed Computing*, Vol. 3(4), pp. 267–286, 1979.

[4.28] S.R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and

S. Borkar, “An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS,” *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 1, pp. 29–41, Jan. 2008.

[4.29] K.-M. Lee, S.-J. Lee, and H.-J. Yoo, “Low-Power Network-on-Chip for High Performance SoC Design,” *IEEE Transactions on Very Large Scale Integration (VLSI) System*, Vol. 14, No. 2, pp. 148–160, Feb. 2006.

[4.30] J. Howard, S. Dighe, Y. Hoskote, S. Vangal, D. Finan, G. Ruhl, D. Jenkins, H. Wilson, N. Borkar, G. Schrom, F. Paillet, S. Jain, T. Jacob, S. Yada, S. Marella, P. Salihundam, V. Erraguntla, M. Konow, M. Riepen, G. Droege, J. Lindemann, M. Gries, T. Apel, K. Henriss, T. Lund-Larsen, S. Steibl, S. Borkar, V. De1, R.V.D. Wijngaart and T. Mattson,, “A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS,” in *Proceeding of IEEE International Solid-State Circuits Conference*, pp. 108–110, Feb. 2010.

[4.31] M. Made, F. Felicijan, A. Efthymiou, D. Edwards and L. Lavagno, “Asynchronous On-Chip Networks,” *IET Proceeding of Computer & Digital Technology*, Vol. 152, No. 2, pp. 273–283, 2005.

[4.32] A. Lines, “Asynchronous interconnect for synchronous soc design,” *IEEE Micro*, Vol. 24, No. 1, pp.32–41, 2004.

[4.33] R. Dobkin, R. Ginosar and C. P. Sotiriou, “High Rate Data Synchronization in GALS SoCs,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 14, No. 10, pp. 1063–1074, Oct. 2006.

[4.34] T. Chelcea and S. M. Nowick, “Robust Interfaces for Mixed-Timing systems,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 8, pp. 857–873, Aug. 2004.

[4.35] I. Miro-Panades, F. Clermidy, P. Vivet and A. Greiner, “Physical Implementation of the DSPIN Network-on-Chip in the FAUST Architecture,” in *Proceeding of IEEE International Symposium on Netwrok-on-Chip*, pp. 139–148, Apr. 2008.

[4.36] P. Vivet, D. Lattard, F. Clermidy, E. Beigne, C. Bernard, Y. Durand, J. Durupt and D. Varreau, “FAUST, an Asynchronous Network-on-Chip based Architecture for Telecom Applications,” in *Proceeding of IEEE International Symposium on Asynchronous Circuits and Systems*, pp. 172–181, Mar. 2006.

[4.37] E. Beignre, F. Clermidy, H. Lhermet, S. Miermont, Y. Thonnart, X.-T. Tran, A. Valentian, D. Varreau, P. Vivet, X. Popon and H. Lebreton, “An Asynchronous Power Aware and Adaptive NoC Based Circuit,” *IEEE Journal of Solid-State Circuits*, Vol. 44, No. 4, pp. 1167–1177, Apr. 2009.

[4.38] E. Beignre and P. Vivet, “Design of On-chip and Off-chip Interfaces for a GALS NoC Architecture,” in *Proceeding of IEEE International Symposium on Asynchronous Circuits and Systems*, pp. 172–181, Mar. 2006.

- [4.39] M. Li, Q.-A. Zeng and W.-B. Jone, “DyXY – A Proximity Congestion-Aware Deadlock-Free Dynamic Routing Method for Network on Chip,” in *Proceeding of Design Automation Conference*, pp. 849-852, June, 2006.
- [4.40] P.-T. Huang and W. Hwang, “An Adaptive Congestion-Aware Routing Algorithm for Mesh Network-on-Chip Platform,” in *Proceeding of IEEE System-on-Chip Conference*, pp. 375-378, Sept. 2009.

## References of Chapter 5

- [5.1] L. Benini and G. De Micheli, *Network on Chips: Technology and Tools*, Morgan Kaufmann, 2006.
- [5.2] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufmann, 2004.
- [5.3] P.P. Pande, C. Grecu, M. Jones, A. Ivanov and R. Saleh, “Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures,” *IEEE Transactions on Computers*, Vol. 54, No. 8, Aug. 2005.
- [5.4] Eric Fleury and Pierre Fraigniaud, “A General Theory for Deadlock Avoidance in Wormhole-Routed Networks,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 9, No. 7, 1998.
- [5.5] C.J. Glass and L.-M. Ni, “Adaptive Routing in Mesh-Connected Networks,” *Technical Report MSU-CPS-ACS-45*, Oct. 15, 1991.
- [5.6] H. Barati, A. Movaghar, A. Barati and A.A. Mazreah, “Routing Algorithms Study and Comparing in Interconnection Networks,” *International Conference on Information and Communication Technologies*, pp.1-5, 2008.
- [5.7] K.-S. Ding, C.-T. Ho and J.-J. Tsay, “Matrix transpose on meshes with wormhole and XY routing,” in *Proceeding of EEE Symposium on Parallel and Distributed Processing*, pp. 656-663, Oct. 1994.
- [5.8] A. Sobhani1, M. Daneshtalab, M.H. Neishaburi, M.D. Mottaghi, A. Afzali-Kusha, O. Fatemi and Z. Navabi, “Dynamic Routing Algorithm for Avoiding Hot Spots in On-chip Networks,” *International Conference on Design and Test of Integrated Systems in Nanoscale Technology*, pp.179-183, Sept. 2006.
- [5.9] G. Ascia, V. Catania, M. Palesi and D. Patti, “Neighbor-on-Path: A New Selection Strategy for On-Chip Networks,” in *Proceeding of IEEE/ACM/IFIP Workshop on Embedded Systems for Real Time Multimedia*, pp.79-84, 2006.
- [5.10] M. Daneshtalab, A. Sobhani, A. Afzali-Kusha, O. Fatemi and Z. Navabi, “NoC Hot Spot minimization Using AntNet Dynamic Routing Algorithm,”

*International Conference on Application-specific Systems, Architectures and Processors*, pp. 33-38, 2006.

[5.11] J. Hu and R. Marculescu, “DyAD – Smart Routing for Network-on-Chip,” in *Proceeding of IEEE/ACM Design and Automation Conference*, pp. 260-263, 2004.

[5.12] M. Li, Q.A. Zeng and W.B. Jone, “DyXY – A Proximity Congestion-Aware Deadlock-Free Dynamic Routing Method for Network on Chip,” in *Proceeding of IEEE/ACM Design and Automation Conference*, pp. 849-852, June, 2006.

[5.13] M. Palesi, G. Longo, S. Signorino, R. Holsmark, S. Kumar and V. Catania, “Design of Bandwidth Aware and Congestion Avoiding Efficient Routing Algorithm for Network-on-Chip Platforms,” in *Proceeding of IEEE/ACM International Symposium on Network-on-Chip*, pp. 97-106, 2008.

[5.14] J. Flich, A. Mejia, P. Lopez and J. Duato, “Region-Based Routing: An Efficient Routing Mechanism to Tackle Unreliable Hardware in Network on Chips,” in *Proceeding of IEEE/ACM International Symposium on Network-on-Chip*, pp. 183-194, 2007.

[5.15] M. Koibuchi, H. Matsutani, H. Amano and T. M. Pinkston, “A Lightweight Fault-Tolerant Mechanism for Network-on-Chip,” in *Proceeding of IEEE/ACM International Symposium on Network-on-Chip*, pp. 13-22, 2008.

[5.16] M.R. Aliabadi, A. Khademzadeh and A.M. Raiyat, “A Novel Reliable Routing Algorithm for Network on Chips,” in *Proceeding of International Conference on Industrial Engineering and Engineering Management*, pp.1375-1379, 2008.

[5.17] G.M. Chiu, “The odd-even turn model for adaptive routing,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, No. 7, pp. 729-738, July 2000.

[5.18] C. J. Glass and L.-M. Ni, “The turn model for adaptive routing,” *Journal of the Association for Computing Machinery*, pp. 874-902, Sept. 1994.

## References of Chapter 6

[6.1] M. Meribout, T. Ogura, and M. Nakanishi, “On using the CAM concept for parametric curve extraction,” *IEEE Transactions on Image Processing*, Vol. 9, No. 12, pp. 2126–2130, Dec. 2000.

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

[6.3] D.J. Craft, “A fast hardware data compression algorithm and some algorithmic extensions,” *IBM Journal of Research Development*, Vol. 42, No. 6, pp. 733–745, Nov. 1998.

[6.4] L.-Y. Liu, J.-F. Wang, R.-J. Wang, and J.-Y. Lee, “CAM-based VLSI architectures for dynamic Huffman coding,” *IEEE Transactions on Consumer Electronics*, Vol. 40, No. 3, pp. 282–289, Aug. 1994.

[6.5] S. Choi, S.-J. Song, K. Sohn, H. Kim, J. Kim, N. Cho, J.-H. Woo, J. yoo and H.-J. Yoo, “A 24.2- $\mu$ W Dual-Mode Human Body Communication Controller for Body Sensor Network,” in *Proceeding of IEEE European Solid-State Circuits Conference*, pp. 227–230, 2006.

[6.6] S. Choi, K. Sohn, J. Kim, J. Yoo and H.-J. Yoo, “A TCAM-based Periodic Event Generator for Multi-Node Management in the Body Sensor Network,” in *Proceeding IEEE Asian Solid-State Circuits Conference*, pp. 307–310, 2006.

[6.7] C.-C. Wang, C.-J. Cheng, T.-F. Chen, and J.-S. Wang, “An Adaptively Dividable Dual-Port BiTCAM for Virus-Detection Processors in Mobile Devices,” *IEEE Journal of Solid-State Circuits*, Vol. 44, No. 5, pp. 1571–1581, Jan. 2009.

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

[6.9] N.-F. Huang, K.-B. Chen and W.-E. Chen, “Fast and Scalable Multi-TCAM Classification Engine for Wide Policy Table Lookup,” in *Proceeding of IEEE International Conference on Advanced Information Networking and Applications*, Vol. 1, pp. 792–797, Mar. 2005.

[6.10] M. Kobayashi, T. Murase and A. Kuriyama, “A Longest prefix match search engine for multi-gigabit IP processing,” in *Proceeding of IEEE International Conference on Communication*, Vol.3, pp. 1360–1364, Jun. 2000.

[6.11] Y. Tang, W. Lin and B. Liu, “A TCAM Index Scheme for IP Address Lookup,” *First International Conference on Communications and Networking in China*, pp. 1–5, Oct, 2006.

[6.12] N.-F. Huang, W.-E. Chen, J.-Y. Luo and J.-M. Chen, “Design of Multi-field IPv6 Packet Classifiers Using Ternary CAMs,” *IEEE GLOBECOM*, Vol. 3, pp. 1878–1881, Nov. 2001.

[6.13] L. Benini and G. De Micheli, *Network on Chips: Technology and Tools*, Morgan Kaufmann, 2006.

[6.14] L. Wang, H. Sing, Y. Jiang and L. Zhang, “A routing-table-based adaptive and minimal routing scheme on network-on-chip architectures,” *Elsevier Journal of Computer and Electrical Engineering*, Vol. 35, No. 6, pp.846-855, Nov. 2009.

[6.15] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufmann, 2004.

[6.16] M. Daneshbalab, A. Sobhani, A. Afzali-Kusha, O. Fatemi and Z. Navabi, “NoC Hot Spot minimization Using AntNet Dynamic Routing Algorithm,” *International Conference on Application-specific Systems, Architectures and Processors*, pp. 33-38, 2006.

[6.17] E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny, “Routing table minimization for irregular mesh NoCs” in *Proceeding of Design, automation and test in Europe*, pp. 942-947, 2007.

[6.18] H. Che, Z. Wang, K. Zheng and B. Liu, “DRES: Dynamic Range Encoding Scheme for TCAM Coprocessors,” *IEEE Transactions on Computer*, Vol. 57, No. 7, pp. 902–915, Jul. 2008.

[6.19] H. Noda, K. Inoue, M. Kuroiwa, A. Amo, A. Hachisuka, H.J. Mattausch, T. Koide, S. Soeda, K. Dosaka and K. Arimoto, “A 143MHz 1.1W 4.5Mb dynamic TCAM with hierarchical searching and shift redundancy architecture,” in *Proceeding of IEEE International Solid-State Circuits Conference*, pp. 208–523, Feb. 2004.

[6.20] 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 and K. Arimoto, K. Fujishuma, K. Anami and 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. 2002.

[6.21] J.-S. Wang, H.-Y. Li, C.-C. Chen and C. Yeh, “An AND-type Match-line Scheme for High-Performance Energy -Efficient Content Addressable Memories,” *IEEE Journal of Solid-State Circuits*, Vol. 41, No. 5, pp. 1108–1119, May 2006.

[6.22] Y.-D. Kim, H.-S. Ahn, J.-Y. Park, S. Kim and D.-K. Jeong, “A Storage- and Power-Efficient Range-Matching TCAM for Packet Classification,” in *Proceeding of IEEE International Solid-State Circuits Conference*, pp. 587–596, Feb. 2006.

[6.23] J.-S. Wang, C.-C. Wang, and C. Yeh, “TCAM for IP-Address Lookup Using Tree-style AND-type Match Lines and Segmented Search Lines,” in *Proceeding of IEEE International Solid-State Circuits Conference*, pp. 166–167, Feb. 2006.

[6.24] K. Pagiamtzis, and A. Sheikholeslami, “A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme,” *IEEE Journal of Solid-State Circuits*, Vol. 39, No. 9, pp.1512–1519, Sept. 2004.

[6.25] S.-D. Choi, K. Sohn and H.-J. Yoo, “A 0.7-fJ/bitsearch 2.2-ns search time hybrid-type TCAM architecture,” *IEEE Journal of Solid-State Circuits*, Vol. 40, No. 1, pp.254–260, Jan. 2005.

[6.26] N. Mohan and M. Sachdev, “Low-Capacitance and Charge-Shared Match Lines for Low-Energy High-Performance TCAMs,” *IEEE Journal of Solid-State Circuits*, Vol. 42, No. 9, pp.2054–2060, Sept. 2007.

[6.27] C.-C. Wang, J.-S. Wang and C. Yeh, “High-Speed and Low-Power Design Techniques for TCAM Macros,” *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 2, pp.530–540, Feb. 2008.

[6.28] T. Kusumoto, D. Ogawa, K. Dosaka, M. Miyama and Y. Mastsuda, “A charge recycling TCAM with Checkerboard Array arrangement for low power applications,” in *Proceeding of IEEE Asian Solid-State Circuits Conference*, pp. 253–256, 2008.

[6.29] K. L. Shepard, V. Narayanan, and R. Rose, “Harmony: A methodology for noise analysis in deep submicron digital integrated circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 6, pp. 1132-1150, Aug. 1999.

[6.30] C.-H. Hua, T.-S. Cheng and W. Hwang, “Distributed Data-Retention Power Gating Techniques for Column and Row Co-controlled Embedded SRAM,” *IEEE International Workshop on Memory Technology, Design, and Testing*, pp. 129–134, Aug. 2005.

[6.31] K.S. Min, H.D. Choi and T. Sakurai, “Leakage-suppressed clock-gating circuit with Zigzag Super Cut-off CMOS (ZSCCMOS) for leakage-dominant sub-70-nm and sub-1-V-VDD LSIs,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 14, No. 4, pp. 430–435, April 2006.

[6.32] Y. Tsukikawa, T. Kajimoto, Y. Okasaka, Y. Morooka, K. Furutani, H. Miyamoto and H. Ozaki, “An Efficient Back-Bias Generator with Hybrid Pumping Circuit for 1.5-V DRAMs,” *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 4, pp. 534–538, April 1994.

[6.33] P. Favrat, P. Deval and M. Declercq, “A High-Efficiency CMOS Voltage Doubler,” *IEEE Journal of Solid-State Circuits*, Vol. 33, No. 3, pp. 410–416, March 1998.

[6.34] K. Roy, S. Mukhopadhyay and H. Mahmoodi-Meimand, “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits,” *Proceedings of the IEEE*, Vol.91, No.2, pp. 305–327, Feb. 2003.

[6.35] I. Arsovski and R. Wistort, “Self-referenced sense amplifier for across-chip-variation immune sensing in high-performance Content

-Addressable Memories," in *Proceeding of IEEE Custom Integrated Circuits Conference*, pp.453–456, 2006.

[6.36] [Online]. Available: <http://go6.net/ipv6-6bone>

[6.37] T.-C. Chen and R.-B. Sheen, "A Power-Efficient Wide-Range Phase-Locker Loop," *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 1, pp. 51–62, Jan. 2002.

## References of Chapter 7

[7.1] Y. Yuyama, M. Ito, Y. Kiyoshige, Y. Nitta, S. Matsui, O. Nishii, A. Hasegawa, M. Ishikawa, T. Yamada, J. Miyakoshi, K. Terada, T. Nojiri, M. Satoh, H. Mizuno, K. Uchiyama, Y. Wada, K. Kimura, H. Kasahara, H. Maejima, "A 45nm 37.3GOPS/W heterogeneous multi-core SoC," *Digest of Technical IEEE International Solid-State Circuits Conference Papers*, pp.100-101, 7-11 Feb. 2010.

[7.2] B. Jacob, S. W. Ng and D.T. Wang, *Memory Systems: Cache, DRAM, Disk*, Morgan Kaufmann, 2007.

[7.3] H. Kim, and I. C. Park, "High-performance and low-power memory-interface architecture for video processing applications," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No 11, pp.1160-1170, 2001.

[7.4] S. I. Park, Y. Yi, and I. C Park, "High Performance Memory Mode Control for HDTV Decoders," *IEEE Transactions on Consumer Electronics*, Vol. 49, No. 4, pp. 1348-1353, Nov. 2003.

[7.5] C.-H. Chang, M.-H. Chang, and W. Hwang, "A Flexible Two-Layer External Memory Management for H.264/AVC Decoder," in *Proceeding of IEEE International SoC Conference*, pp. 219-222, Sept. 2007.

[7.6] J. Zhu, L. Hou, R. Wang, C. Huang, and J. Li, "High Performance Synchronous DRAMs Controller in H.264 HDTV decoder" in *Proceeding of IEEE International Conference Solid-state and Integrated Circuits Technology*, Vol. 3, pp. 1621-1624, 2004.

[7.7] H. Hu, J. Sun and J. Xu, "High Efficiency Synchronous DRAM Controller for H.264 HDTV Encoder," in *Proceeding of IEEE Conference on Industrial Electronics and Applications*, pp. 2132-2136, May 2009.

[7.8] C.-Y. Tsai, T.-C. Chen, T-W. Chen, and L.-G. Chen, "Bandwidth optimized motion compensation hardware design for H.264/AVC HDTV decoder," in *Proceeding of International Symposium on Circuits and Systems*, pp. 273-276, August 2005.

[7.9] Y. Li, Y. Qu, and Y. He, “Memory Cache Based Motion Compensation Architecture for HDTV H.264/AVC Decoder,” in *Proceeding of International Symposium on Circuits and Systems*, pp. 2906-2909, May 2007.

[7.10] T.-D. Chuang, L.-M. Chang, T.-W. Chiu, Y.-H. Chen and L.-G. Chen, “Bandwidth-Efficient Cache-Based Motion Compensation Architecture with DRAM-Friendly Data Access Control,” in *Proceeding International Conference on Acoustics, Speech and Signal Processing*, pp. 2009-2012, 2009.

[7.11] K.-B. Lee, T.-C. Lin and C.-W Jen, “An Efficient Quality-Aware Memory Controller for Multimedia Platform SoC,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, No. 5, pp. 620-633, May 2005.

[7.12] H. Nikolov, T. Stefanov and E. Deprettere, “Efficient External Memory Interface for Multi-Processor Platforms Realized On FPGA Chips,” in *Proceeding of International Conference on Field Programmable Logic and Applications*, pp. 580-584, Aug. 2007.

[7.13] E. Ipek, O. Mutlu, , J.F. Martinez and R. Caruana, “Self-Optimizing Memory Controllers: A Reinforcement Learning Approach,” in *Proceeding of International Symposium on Computer Architecture*, pp. 39-50, 2008.

[7.14] H. Zheng, J. Lin, Z. Zhang and Z. Zhu, “Memory Access Scheduling Schemes for Systems with Multi-Core Processors,” in *Proceeding of IEEE International Conference on Parallel Processing*, pp. 406-413, Sept. 2008.

[7.15] Y. Li, “Cognitive and Integrated Digital Home via Dynamic Media Access,” *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pp. 1-6, May 2009.

[7.16] D.H. Albonesi, “Selective cache ways: on-demand cache resource allocation,” in *Proceeding of International Symposium on Microarchitecture*, pp.248-259, 1999.

[7.17] P. Ranganathan, S. Adve, and N.P. Jouppi, “Reconfigurable Caches and their Application to Media Processing,” in *Proceeding of International Symposium on Computer Architecture*, pp.214-224, 2000.

[7.18] Micron Technology, Inc., [Online]. Available: <http://www.micron.com/>

[7.19] Micron System Power Calculators, [Online]. Available: [http://www.micron.com/support/dram/power\\_calc.html](http://www.micron.com/support/dram/power_calc.html).

[7.20] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, and L.-G. Chen, “Level C+ Data Reuse Scheme for Motion Estimation with Corresponding Coding Orders,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 4, pp.553-558, April 2006.

[7.21] T.-C. Chen, C.-Y. Tsai, Y.-W. Huang, and L.-G. Chen, “Single Reference Frame Multiple Current Macroblocks Scheme for Multiple Reference Frame

Motion Estimation in H.264/AVC,” *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 17, No. 2, pp.242-247, Feb. 2007.

[7.22] H. Shim, K. Kang, and C.-M. Kyung, “Search area selective reuse algorithm in motion estimation,” in *Proceeding of IEEE International Conference on Multimedia and Expo*, pp.1611-1614, July 2007.

[7.23] M.-C. Lin and L.-R. Dung, “Two-step Windowing Technique for Wide Range Motion Estimation,” in *Proceeding of IEEE Asia Pacific Conference on Circuits and Systems*, pp.1478-1481, November 2008.

[7.24] C.-Y. Tsai, T.-C. Chen, T.-W. Chen and L.-G. Chen, “Bandwidth Optimized Motion Compensation Hardware Design for H.264/AVC HDTV Decoder,” in *Proceeding of IEEE International Midwest Symposium on Circuit and Systems*, pp. 1199-1202, Aug. 2005.

[7.25] R.-G. Wang, J.-T. Li and C. Huang, “Motion Compensation Memory Access Optimization Strategies for H.264/AVC Decoder,” in *Proceeding of IEEE International Conference Acoustics, Speech, and Signal Processing*, pp. 97-100, Mar. 2005.

[7.26] Y. Li and Y. He, “Bandwidth Optimized and High Performance Interpolation Architecture in Motion Compensation for H.264/AVC HDTV Decoder,” *Journal of Signal Processing Systems*, Vol. 52, No.2, pp. 111-126, Aug. 2008.

[7.27] T.-M. Liu and C.-Y. Lee, “Memory-Hierarchy-Based Power Reduction for H.264/AVC Video Decoder,” in *Proceeding of International Symposium on VLSI Design, Automation and Test*, pp. 247- 250, April 2006.

[7.28] P. Chao and Y.-L. Lin, “A Motion Compensation System with a High Efficiency Reference Frame Pre-Fetch Scheme for QFHD H.264/AVC Decoding,” in *Proceeding of IEEE International Conference on Circuits and Systems*, pp. 256-259, May 2008.

[7.29] P.-C. Wang, “Layer-Adaptive Mode Decision based on Rate Distortion Cost Correlation Coefficients for Scalable Video Coding,” *master thesis*, Department of Electrical Engineering, National Dong Hwa University, 2009.

[7.30] H. Schwarz, D. Marpe, T. Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, No. 9, pp.1103-1120, Sept. 2007.

[7.31] C.-H. Chang, M.-H. Chang, and W. Hwang, “A Flexible Two-Layer External Memory Management for H.264/AVC Decoder,” in *Proceeding of IEEE International SOC Conference*, pp. 219-222, Sept. 2007.

[7.32] J. Zhu, L. Hou, R. Wang, C. Huang, and J. Li, “High Performance Synchronous DRAMs Controller in H.264 HDTV decoder,” in *Proceeding of*

*IEEE International Conference on Solid-state and Integrated Circuits Technology*, pp. 1621-1624, 2004.

- [7.33] HP Labs : CACTI model, available on <http://www.hpl.hp.com/research/cacti>
- [7.34] M. Mrak, M. Grgic, S. Grgic, “Scalable video coding in network applications,” *International Symposium on Video/Image Processing and Multimedia Communications*, pp. 205-211, 2002.



# Vita

---

## PERSONAL INFORMATION

Birth Date: 08 January, 1979

Birth Place: Taichung, Taiwan

E-Mail Address: [bug.ee91g@nctu.edu.tw](mailto:bug.ee91g@nctu.edu.tw)

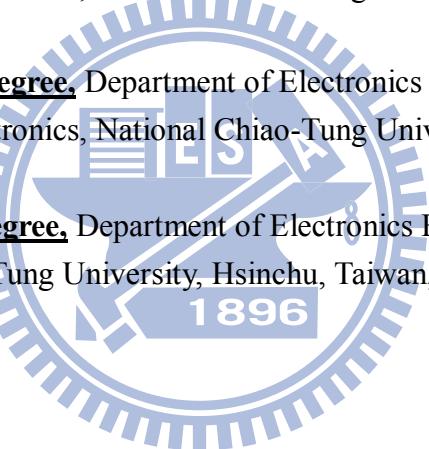
## EDUCATION

02/2004 – 10/2010 **Ph.D. Degree**, Department of Electronics Engineering & Institute of Electronics, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C.

09/2002 – 10/2010 **M.S. Degree**, Department of Electronics Engineering & Institute of Electronics, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C.

09/1997 – 06/2002 **B.S. Degree**, Department of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C.

## AWARD



- 年度優良志工, 新竹家庭扶助中心, 2003.
- Student Outstanding Research Award, MOEA/DoIT (Advanced System Designs for High Performance and Low Power Dual-Core Processors), 2005.
- Outstanding Oral Paper Award, "A Capacitive Boosted Buffer for Energy-Efficient and Variation-Tolerant Sub-Threshold Interconnect," Electronics Technology Symposium (ETS), 2009.
- Ph.D. Teaching Assistant Scholarship, National Chiao-Tung University, 2009.

## PROJECT

- Low Power Embedded Memory Design in Nano-Scale CMOS Technology, TSMC Research Contract, 2003– 2007.
- SoC for MPEG-7/21 Applications and Next Generation Mobile Communications Research (Main project), NSC, 2003-2006.

3. Low Voltage High Bandwidth Embedded Memory for Multimedia Communication SoC Application, (Sub-Project 1), NSC, 2003-2006.
4. Core Technology for e-Home Environment (main project), NSC 2004-2007.
5. System Memory Design for Next-Generation e-Home Server (sub-project 1), NSC 2004-2007.
6. Advanced System Designs for High Performance and Low Power Dual-Core Processors (sub-project 2), MOEA/DoIT, 2005-2007.
7. Multi-System Merging and Green Computing Techniques for Wireless Video Entertainment (main project), NSC, 2006-2010.
8. Low Power On-Demand Memory System for Multi-Core Design (sub-project 1), NSC, 2006-2010.

## **EXPERIENCE**

- ◆ Low Power SoC Lab, IEE, NCTU, Hsinchu, Taiwan
  - Position: Research Assistant (Sep 2002–now)
  - Nature of Position: Develop low power circuits, embedded memory and memory systems under projects supported by Ministry of Economic Affairs (MOEA) and National Science Council (NSC), Taiwan.
- ◆ Department of Electronics Engineering, NCTU, Hsinchu, Taiwan
  - Position: Teaching Assistant (Sep 2003–now)
  - Nature of Position:
    - IP Core Design, 2003. (Lecturer: Prof. Wei Hwang)
    - Computer Architecture, 2004. (Lecturer: Prof. Juinn-Dar Hwang)
    - Embedded Memory Design, 2005–2008. (Lecturer: Prof. Wei Hwang)
    - Low Power and High Performance Digital IC Design, 2007–2008. (Lecturer: Prof. Wei Hwang)
    - Digital Integrated Circuits, 2008–2009. (Lecturer: Prof. Wei Hwang)
    - Multi-Core Architecture and System, 2009. (Lecturer: Prof. Bo-Cheng Lai)
    - Digital Lab., 2006, 2008, 2010. (Department of Electronics Engineering)
    - Electronics Lab. (I), 2006, 2008, 2009. (Department of Electronics Engineering)
    - Electronics Lab. (II), 2005, 2007, 2009. (Department of Electronics Engineering)
- ◆ Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu, Taiwan
  - Position: Summer Intern (Apr. 2006–Jun. 2006)

- Nature of Position: Worked in the memory design division and helped to build an I/O placement interface.
- Mentor: Hong-Ren Liao

◆ Department of Electronics Engineering, NCTU, Hsinchu, Taiwan

- Position: Pre-Ph.D. Teaching Assistant (Feb. 2009–Jun. 2009)
- Nature of Position: Repeated the course of “Signals and Systems” for the students, and helped the lecturer teach Matlab. (Lecturer: Prof. Kuei-Ann Wen)

## **EXTRA-CURRICULAR ACTIVITIES**

參與組織	主要活動記錄
交大幼幼社	<p><u>1997/09-1998/08</u> 1th 交大幼幼營(設計組組員), 執行組組員, 19<sup>th</sup> 兒童向上營(設計組組員)</p> <p><u>1998/09-1999/08</u> 2th 交大幼幼營(設計組組長), 執行組組員, 20<sup>th</sup> 兒童向上營(機動組組長)</p> <p><u>1999/09-2000/08</u> 幼圖組組長, 3th 交大幼幼營(營長), 21<sup>th</sup> 兒童向上營(指導員)</p> <p><u>2000/09-2006/08</u> 4<sup>th</sup> 交大幼幼營(設計組指導員), 5<sup>th</sup>, 6<sup>th</sup>, 7<sup>th</sup>, 8<sup>th</sup>, 9<sup>th</sup> 交大幼幼營(指導員) 22<sup>th</sup>, 23<sup>th</sup>, 24<sup>th</sup>, 25<sup>th</sup> 兒童向上營(指導員)</p>
新竹家扶中心	兒童保護營共三屆, 家扶中心原住民鄉多采多姿營共32屆 家扶中心課業輔導志工(1997-2006)
新竹文化中心	新竹文化中心寶兔館義工(1998-2000)
新竹仁愛之家	新竹仁愛之家志工(1998-2008)
交大電子系學會	<p><u>1997-1999</u> 器材組組員. 系圖組組員. 迎新露營活動組組長 曾擔任系上許多晚會主持人及主辦人</p> <p><u>11<sup>th</sup>, 12<sup>th</sup> 微電子營活動組組員</u></p> <p><u>1999-2000</u> 系學會會長 13<sup>th</sup> 微電子營活動組組長 系隊 籃球隊(1997-2000), 桌球隊(2001-2010)</p>
交大田徑隊	1997-1999 田徑隊隊員(專攻400M, 800M)
交大建中校友會	1997-1998 活動部組員

## **PUBLICATIONS**

### **Journal Paper**

- [1] **P.-T. Huang**, S.-W. Chang, W.-Y. Liu and W. Hwang, “Energy-Efficient Design for Ternary Content Addressable Memory,” *International Journal of Electrical Engineering*, Vol. 15, No.2, pp.97-108, 2008.
- [2] **P.-T. Huang** and W. Hwang, “A 0.047fJ/Bit/Search 65nm 256x144 Energy-Efficient TCAM Macro for Network Routers,” *IEEE Journal of Solid-State Circuits. (in press, Feb. 2011)*
- [3] **P.-T. Huang**, X.-R. Lee, H.-C. Chang, C.-Y. Lee and W. Hwang, “A Low Power DCVSPG Pulsed Latch for Viterbi Decoder,” *Journal of Low Power Electronics. (accepted for publication, 2010)*
- [4] **P.-T. Huang** and W. Hwang, “Self-Calibrated Energy-Efficient and Reliable Channels for On-Chip Interconnection Network,” *IEEE Transactions on Circuits and Systems-I. (revised for publication, 2010)*.
- [5] **P.-T. Huang** and W. Hwang, “Two-Level FIFO Buffer Design for Routers in On-Chip Interconnection Network,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. (submitted for Publication, 2010)*.

### **Conference Paper**

- [1] **P.-T. Huang** and W. Hwang, “Low Power Encoding Schemes for Run-Time On-Chip Bus,” *Proceeding of IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS)*, pp. 1025-1028, Dec. 2004.
- [2] S.-H. Lin, **P.-T. Huang** and W. Hwang, “A Power-Speed Optimization Technique of High-Speed Multiply-Accumulate Design,” *Proceeding of 16<sup>th</sup> VLSI/CAD Symposium*, Aug. 2005.
- [3] C.-K. Tsai, **P.-T. Huang** and W. Hwang, “Low power Pulsed Edge-Triggered Latches Design,” *Proceeding of 16<sup>th</sup> VLSI/CAD Symposium*, Aug. 2005.
- [4] **P.-T. Huang** and W. Hwang, “2-Level FIFO Architecture Design for Switch Fabrics in Network-on-Chip,” *Proceeding of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 4863-4866, May 2006.
- [5] S.-W. Chang, **P.-T. Huang** and Wei Hwang, “A Novel Butterfly Match-line Scheme with Don’t Care Based Hierarchical Search-Line for TCAM,” *17<sup>th</sup> VLSI/CAD Symposium*, Aug. 2006.
- [6] **P.-T. Huang**, W.-K. Chang and W. Hwang, “Low Power Content Addressable

Memory with Pre-Comparison Scheme and Dual-Vdd Technique," *17<sup>th</sup> VLSI/CAD Symposium*, Aug. 2006.

[7] J.-W. Yang, **P.-T. Huang** and W. Hwang, "On-chip DC-DC Converter with Frequency Detector for Reconfigurable Multiplier-Accumulator Unit," *17<sup>th</sup> VLSI/CAD Symposium*, Aug. 2006.

[8] W.-L. Su, **P.-T. Huang**, H.-M. Chiueh and W. Hwang, "A Low Power Pulsed Edge-Triggered Latch for Survivor Memory Unit of Viterbi Decoder," *17<sup>th</sup> VLSI Design/CAD Symposium*, Aug. 2006.

[9] J.-W. Yang, **P.-T. Huang** and W. Hwang, "On-chip DC-DD Converter with Frequency Detector for Dynamic Voltage Scaling Technology," *Proceeding of IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 667-671, Dec. 2006.

[10] **P.-T. Huang**, W.-K. Chang and W. Hwang, "Low Power Pre-Comparison Scheme for NOR-Type 10T Content Addressable Memory," *Proceeding of IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 1303-1307, Dec. 2006.

[11] W.-L. Su, H.-M. Chiueh, **P.-T. Huang** and W. Hwang, "A Low Power Pulsed Edge-Triggered Latch for Survivor Memory Unit of Viterbi Decoder," *Proceeding of IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 553-556, Dec. 2006.

[12] **P.-T. Huang**, S.-W. Chang, W.-Y. Liu, and Wei Hwang, "A 256x128 Energy-Efficient TCAM with Novel Low Power Schemes," *Proceeding of IEEE International Symposium on VLSI Design, Automation and Test*, pp.32-35, Apr. 2007.

[13] W.-Y. Liu, **P.-T. Huang** and W. Hwang, "An Energy-Efficient 256x144 TCAM Design," *18<sup>th</sup> VLSI/CAD Symposium*, Aug. 2007.

[14] M.-T. Chang, **P.-T. Huang**, and W. Hwang, "A 65nm Low Power 2T1D Embedded DRAM with Leakage Current Reduction," *Proceeding of IEEE International SOC Conference (SOCC)*, pp.207-210, Sept. 2007.

[15] **P.-T. Huang**, W.-L. Fang, Y.-L. Wang and W. Hwang, "Low Power and Reliable Interconnection with Self-Corrected Green Coding Scheme for Network-on-Chip," *Proceeding of ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, pp. 77-83, Apr. 2008.

[16] **P.-T. Huang**, S.-W. Chang, W.-Y. Liu, and W. Hwang, "Green Micro-architecture and Circuit Co-Design for Ternary Content Addressable Memory," *Proceeding of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3322-3325, May 2008.

[17] L.-P. Chuang, M.-H. Chang, **P.-T. Huang**, C.-H. Kan, and W. Hwang, "A 5.2

mW All-Digital Fast-Lock Self-Calibrated Multiphase Delay-locked Loop,” *Proceeding of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3342-3345, May 2008.

[18] **P.-T. Huang**, W.-L. Fang, and W. Hwang, “A Self-Calibrated Voltage Scaling Technique for Reliable Interconnections in Network-on-Chip,” *19<sup>th</sup> VLSI Design /CAD Symposium*, Aug. 2008.

[19] M.-T. Chang, **P.-T. Huang**, and W. Hwang, “A Robust Ultra-low Power Asynchronous FIFO Memory With Self-Adaptive Power Control,” *Proceeding of IEEE International SoC Conference, (SOCC)*, pp.175-178, Sept. 2008.

[20] Y. Chang, **P.-T. Huang** and W. Hwang, “A Capacitive Boosted Buffer for Energy-Efficient and Variation-Tolerant Sub-threshold Interconnect,” *Electronic Technology Symposium (ETS)*, Jun. 2009.

[21] **P.-T. Huang** and W. Hwang, “An adaptive Congestion-Aware Routing Algorithm for Mesh Network-on-Chip Platform,” *Proceeding of IEEE International SoC Conference, (SOCC)*, pp. 375-378, Sept. 2009.

[22] **P.-T. Huang** and W. Hwang, “Energy-Efficient Techniques for Circuit Design in Netowork-on-Chip Platforms,” *Proceeding of IEEE International Symposium on Green Circuits and Systems (ICGCS)*, pp. 305-310, Jun. 2010. (Invited paper)

[23] T.-H. Lin, **P.-T. Huang**, and W. Hwang, “Power Noise Suppression Technique using Active Decoupling Capacitor for TSV 3D Integration,” *Proceeding of IEEE International SoC Conference, (SOCC)*, pp. 209-212, Sept. 2010.

## **Patent**

### **(i) United States Patents Granted**

[1] S.-W. Chang, **P.-T. Huang**, W. Hwang and M.-H. Chang, “Stored Don’t-care Based Hierarchical Search-line,” U. S. Patent No. 7525827, Apr. 28, 2009.

[2] **P.-T. Huang**, W.-Y. Liu and W. Hwang, “Super Leakage Current Cut-off Technique for Ternary Content Addressable Memory,” U. S. Patent No. 7616469, Nov. 11, 2009.

[3] **P.-T. Huang**, W.-Y. Liu and W. Hwang, “Leakage Current Cut-off Device for Ternary Content Addressable Memory,” U.S. Patent, 2010.

### **(ii) US Patent Application and Pending**

[1] **P.-T. Huang**, S.-W. Chang and W. Hwang, “Butterfly Match-line Scheme”, US 11/675,440, Feb. 2007.

[2] M.-T. Chang, P.-T. Huang and W. Hwang, “Dual-Threshold-Voltage Two-Port Sub-threshold SRAM Cell Apparatus,” US 12/654,730, Dec. 2009.

**(iii) TW (Taiwan) Patents Granted**

[1] 張書瑋、黃柏蒼、張銘宏及黃威,“內儲存無關項之階層式搜尋線”,台灣發明專利取得 (I321793), 99年3月11日

[2] 黃柏蒼、張書瑋及黃威,“蝴蝶式比較線結構及其搜循方法”,台灣發明專利申請(I324346) , 99年5月1日

**(v) TW (Taiwan) Patent Application and Pending**

[1] 黃柏蒼、劉文彥及黃威,“三元內容可定址記憶體漏電流截斷裝置”,台灣發明專利申請,申請號:096149397, 96年12月21日

[2] 黃柏蒼、劉文彥及黃威,“三元內容可定址記憶體漏電流超截斷裝置”,台灣發明專利申請,申請號:096149999, 96年12月25日

[3] 張牧天、黃柏蒼及黃威,“具備自我補償機制之全差分單阜次臨界靜態隨機存取記憶體單元”,台灣發明專利申請,申請號:097149911, 97年12月19日

[4] 張牧天、黃柏蒼及黃威,“雙門檻電壓雙阜次臨界靜態隨機存取記憶體單元”,台灣發明專利申請,申請號:098100288, 98年11月10日

[5] 林天鴻、黃柏蒼及黃威,“用於矽穿孔三維積體電路之使用主動耦合電容的電源雜訊抑制機制”,台灣發明專利申請中

[6] 張雍、黃柏蒼及黃威,“應用於多任務/多核心系統單晶片之適應性配置快取記憶體”,台灣發明專利申請中

[7] 張雍、黃柏蒼、陳宥辰、李國龍、張添烜及黃威,“應用於可階式視訊編碼的跨層預取機制”,台灣發明專利申請中