

國立交通大學

電子工程學系 電子研究所

碩 士 論 文

應用於無線影像娛樂系統的隨選記憶體系統

On-Demand Memory System for Wireless Video
Entertainment Systems

研 究 生：張 雍

指 導 教 授：黃 威 教 授

中 華 民 國 九 十 九 年 七 月

應用於無線影像娛樂系統的隨選記憶體系統

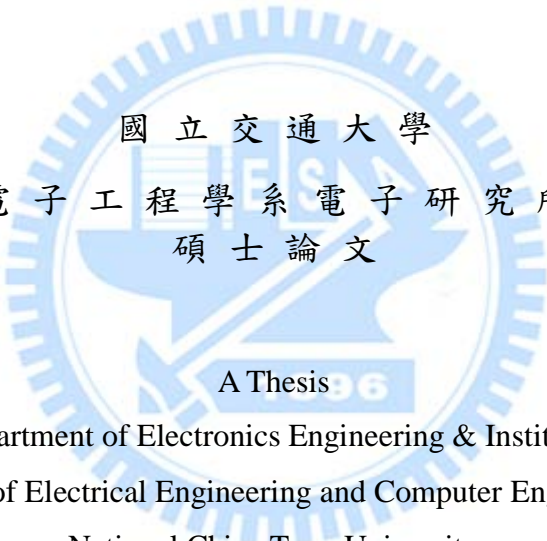
On-Demand Memory System for Wireless Video
Entertainment Systems

研究生：張 雍

Student : Yung Chang

指導教授：黃 威 教授

Advisor : Prof. Wei Hwang



國立交通大學
電子工程學系電子研究所
碩士論文

A Thesis

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

for the Degree of

Master

in

Electronics Engineering

July 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年七月

應用於無線影像娛樂系統的隨選記憶體系統

學生：張 雍

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，邁向多核心、多執行及多系統融合平台才有辦法達到未來的需求。然而，多核心平台需要一個的記憶體系統來提供足夠的資料頻寬以及良好的記憶體管理機制。在本論文中，我們提出了適合於多核心平台的高效能、低功率隨選記憶體系統。並將其應用在無線影像娛樂系統上。

在所提出的隨選記憶體系統中，主要包含了分散式記憶體管理器以及集中式記憶體管理器。在分散式記憶體管理器中，我們提出了一個借取 (borrowing) 機制，此機制可以動態地分配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，所提出的適應性快取控制機制可以根據不同處理單元的記憶體存取特性來分配不一樣記憶體資源。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外記憶體。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)，我們提出了預取(pre-fetch)資料的機制和有效率的動態記憶體(DRAM)資料安排的機制來減少快取記憶體的失誤率以及動態記憶體的能源消耗。並利用在集中式記憶體中的適應性快取控制，可讓系統達到最佳的記憶體使用率。

On-Demand Memory System for Wireless Video Entertainment Systems

Student : Yung Chang

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 processing capability and a merging multi-task system to sustain the growth. Therefore, a well-organized memory system can provide enough bandwidth and optimize memory managements. In this thesis, an on-demand memory system is presented to overcome the challenges in the multi-task and heterogeneous multi-core system design. The proposed on-demand memory system, consisting of distributed and centralized memory management units (MMUs), provides energy-efficient memory-centric on-chip data communication for wireless video entertainment systems.

Distributed MMUs (d-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 requirement of the processor elements in the system, adaptive memory resource allocation is applied via 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).

Acknowledgements

我要感謝我的指導教授黃威教授這兩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系統，多媒體，與系統整合的領域，讓我這兩年的研究雖然辛苦但是充滿了挑戰及樂趣。

另外要特別的感謝就是跟我同一個團隊的老師，學長和同學。特別感謝實驗室的黃柏蒼學長、王湘斐學長在這段研究期間的合作與指導。也要感謝 eHomeII 計畫團隊的黃威教授，黃經堯教授、許騰尹教授、張錫嘉教授、張添烜教授、闕河鳴教授、劉志尉教授、桑梓賢教授的指教，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。在這也要特別感謝同一個計畫團隊的李國龍博班學長以及陳宥宸同學，在研究合作中給了許多支援與協助。

接下來要感謝實驗室的張銘宏學長、謝維致學長、楊皓義學長以及邱議德、謝忠穎、陳璽文、林天鴻同學。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。

最後要感謝我的家人和朋友在研究過程給我的打氣與鼓勵以及關心，讓我的研究過程能順利完成。

Contents

Chapter 1 Introduction	1
1.1 Motivation.....	1
1.2 Contributions.....	2
1.3 Organization.....	3
Chapter 2 Related Researches of Memory Systems	5
2.1 Memory hierarchy.....	5
2.2 Cache.....	6
2.2.1 An overview of Cache Memory	6
2.2.2 Reconfigurable Cache Techniques and Improvements.....	8
2.3 DRAM.....	16
2.3.1 DRAM characteristic	16
2.3.2 DRAM controller techniques and Improvements.....	18
2.3.3 Modern DRAM Development	22
Chapter 3 Memory-Centric On-chip Data Communication Platform for Wireless Video Entertainment Systems	28
3.1 Motivations	28
3.2 Memory-Centric On-chip Data Communication Platform	30
3.2.1 Overall Architecture.....	30
3.2.2 Concepts of On-Demand Memory System	32
3.3 Wireless Video Entertainment Systems	33
3.3.1 Wireless Processing Unit (WPU).....	36
3.3.2 Medium Access Control (MAC).....	38
3.3.3 LT Coding	39
3.3.4 Scalable Video Coding (SVC)	41
3.4 Memory-Centric On-Chip Data Communication Platform for Wireless Video Entertainment Systems.....	42
Chapter 4 Hierarchy Memory Management Units for On-Demand Memory System	45
4.1 Distributed Memory Management Unit Organization	45
4.1.1 Design of d-MMU.....	46
4.2 Centralized Memory Management Unit Organization.....	53
4.2.1 Design of c-MMU	54
4.2.2 External Memory Interface in DRAM controller	60
4.2.3 Simulation Results of the Adaptive Cache.....	67
4.3 Summary	75
Chapter 5 On-Demand Memory System for Wireless Video	

Entertainment Systems	77
5.1 Data Pre-fetch for SVC	77
5.1.1 Introduction.....	77
5.1.2 Inter-layer prediction of the SVC.....	78
5.1.3 Proposed Inter-layer Pre-fetch Scheme.....	80
5.2 Address Translator for SVC	83
5.2.1 Introduction.....	83
5.2.2 Centralized MMU with Address Translator.....	84
5.2.3 Data Arrangement	85
5.3 Analysis & Simulation Results	88
5.3.1 Improvement of adding IPS	89
5.3.2 Improvement of adding Address Translator.....	91
5.3.3 Analysis and Simulation Results of Adaptive Cache Control for Wireless Video Entertainment Systems	96
5.4 Summary	101
Chapter 6 Conclusions and Future Work.....	103
6.1 Conclusions.....	103
6.2 Future Work	104
Bibliography.....	106
References of Chapter 1	106
References of Chapter 2.....	107
References of Chapter 3.....	110
References of Chapter 4.....	112
References of Chapter 5.....	113
References of Chapter 6.....	116
Vita.....	117

List of Figures

Fig.2. 1 Memory hierarchy.....	5
Fig.2. 2 A simple cache memory.	7
Fig.2. 3 A four-block cache configured as direct mapped, two-way set associative, and fully associative.....	7
Fig.2. 4 Associativity-based partitioning organization for reconfigurable caches	8
Fig.2. 5 Overlapped wide-tag partitioning organization for reconfigurable caches	9
Fig.2. 6 A selective-ways organization.	10
Fig.2. 8 Tiles - A physical organization of molecules.	11
Fig.2. 9 Different steps in cache access in the molecular cache	11
Fig.2. 10 An example of typical CMP cache partitioning	12
Fig.2. 11 Basic structure of the reconfigurable Amorphous Cache for processors with large on-chip cache memories.....	13
Fig.2. 12 Simplified architecture of a DRAM.	16
Fig.2. 13 Bank state diagram.....	17
Fig.2. 14 DDR3 Read command [2.12].....	18
Fig.2. 15 DDR3 Write command [2.12]	18
Fig.2. 16 State machine for storing page hit history information.	19
Fig.2. 17 Interlaced method	20
Fig.2. 19 Configurations of different layers of the proposed memory controller	22
Fig.2. 20 DRAM roadmap	23
Fig.2. 21 CPU v.s. DRAM performance.....	24
Fig.2. 22 Accesses addressed to same bank.....	25
Fig.2. 23 Accesses addressed to different bank	25
Fig.3. 1 Wireless Video Entertainment Systemss.....	28
Fig.3. 2 Homogeneous multi-core platform (a) Intel Polaris (b) Tiler TILEPro64™ Processor	29
Fig.3. 3 Trend of the data transmitting bandwidth	29
Fig.3. 4 Comparison between memory bandwidth, memory capacity and communication efficiency in multi-core systems	30
Fig.3. 5 The architecture of memory-centric on-chip data communication platform	31
Fig.3. 6 Illustration of the memory hierarchy in on-demand memory system.....	33

Fig.3. 7 Multi-Task wireless video entertainment system.....	34
Fig.3. 8 Transmitter and receiver block diagram	35
Fig.3. 9 Single-FFT Architecture for MIMO Modem	36
Fig.3. 10 Single-FFT Architecture for MIMO Modem	37
Fig.3. 11 Single-FFT Architecture for MIMO Modem	37
Fig.3. 12 MAC Layer Architecture.....	38
Fig.3. 13 An example of decidable codewords which BP decoding fails to decode.....	40
Fig.3. 14 Architecture of an SVC encoder.....	41
Fig.4. 1 Block diagram of a local node.....	45
Fig.4. 2 Illustration of read operation	46
Fig.4. 3 Illustration of write operation.....	47
Fig.4. 4 Illustration of hiding miss penalty	47
Fig.4. 5 d-MMU and efficient Network Interface.....	48
Fig.4. 6 Buffer borrowing interface between NI and d-MMU	49
Fig.4. 8 Architecture of the empty memory block searching	50
Fig.4. 9 Searching flow chart of the borrowing mechanism in d-MMU .	51
Fig.4. 10 Block diagrams of borrowing mechanism in network interface	52
Fig.4. 11 Borrowing control policy of the buffering control	52
Fig.4. 12 (a) Execution time under various injection loads and queue sizes (b) Transferred packets under various injection loads and queue sizes. .	53
Fig.4. 13 c-MMU block diagram.....	54
Fig.4. 15 Illustration of the memory partition.....	55
Fig.4. 19 Detail architecture of c-MMU.....	60
Fig.4. 20 Connection of EMI	61
Fig.4. 21 Architecture of EMI	62
Fig.4. 22 State diagram of EMI Finite State Machines	64
Fig.4. 23 bank-miss scheduling	65
Fig.4. 24 read / write scheduling	66
Fig.4. 25 row-conflict scheduling.....	66
Table.4. 5 Simulation of the bandwidth utilization	67
Fig.4. 26 DRAM latency estimation for different situations	69
Fig.4. 28 System configuration interface of the System Power Calculator	71
Fig.4. 29 Summary of the power measurement result in the System Power Calculator	72
Fig.4. 30 Organization of simulation.....	72

Fig.4. 32 Total memory energy consumption	75
Fig.5. 1 Illustration of inter-layer motion prediction [5.12]	79
Fig.5. 2 Illustration of inter-layer residual prediction [5.12]	79
Fig.5. 3 Illustration of inter-layer intra prediction [5.12]	80
Fig.5. 5 Illustration of the Inter-layer Pre-fetch Scheme	82
Fig.5. 6 d-MMU architecture with Pre-fetch Command Generator	83
Fig.5. 7 Centralized MMU architecture with Address translator	84
Fig.5. 8 Architecture of the DRAM organization	85
Fig.5. 9 Conventional mapping scheme for the selected DRAM	86
Fig.5. 10 Video frame arrangement of a GOP	87
Fig.5. 11 Frame map to memory	88
Fig.5. 12 Miss rate of the L1 cache versus L1 cache size	89
Fig.5. 13 L1 cache ways v.s. Miss Rate	90
Fig.5. 14 Memory access count of L2 Cache	90
Fig.5. 15 DRAM access count	90
Fig.5. 16 L1 cache energy measurement	91
Fig.5. 17 DRAM row-miss rate	92
Fig.5. 19 DRAM activate power	93
Fig.5. 20 DRAM bandwidth utilization	93
Fig.5. 21 DRAM energy consumption	94
Fig.5. 22 Total Execution cycles	95
Fig.5. 23 On-chip cache energy consumption	95
Fig.5. 24 Total memory energy consumption	95
Fig.5. 25 Video coding performance [5.18]	96
Fig.5. 26 SVC memory requirements of different scalable layers for a GOP	97
Fig.5. 28 Memory energy consumption for different SVC levels	99
Fig.5. 29 Relation between simulation time interval and decoding SVC level	100
Fig.5. 30 Simulation result of total execution cycles	100
Fig.5. 31 Simulation result of memory energy consumption	100
Fig.6. 1 Architecture of femtocell home multimedia center	105

List of Tables

Table.2. 1 Cost-performance for various memory technologies.....	6
Table.2. 2 Related work of adaptive caches.....	15
Table.2. 3 The maximum transfer rate for SDR, DDR, DDR2 and DDR3	24
Table.2. 4 Number of banks for SDR, DDR, DDR2 and DDR3	25
Table.2. 5 Supply voltages for DDR family	26
Table.4. 1 System Specification	36
Table.4. 3 Micron`s DDR3 configurations	61
Table.4. 4 Common timing parameters of Micron DDR3 SDRAM	64
Table.4. 5 Simulation summary.....	67
Table.4. 7 Summary of system and DRAM Configuration	70
Table.4. 8 List of simulation information	74
Table.4. 9 Memory requirement assumption and corresponding bank assignment for c-MMU	74
Table.5. 1 Selected Micron DDR3 size parameters	86
Table.5. 2 Summary of SVC information	88
Table.5. 3 List of simulation information	98
Table.5. 4 c-MMU bank assignment for wireless video entertainment systems	98

Chapter 1

Introduction

1.1 Motivation

For development of system on a chip (SoC) and multimedia technologies, amount of data and computing required to be processed increase quickly. Multi-task processing technique is more and more important for integrating various processor elements into a chip [1.1]-[1.3]. Generally, most of systems require the memories for storing. In multi-task environment, memory is center of storage system, and it is the most serious bottle neck because the performance of processor elements is much faster than the memory. Accordingly, the organization of memory system for a multi-task system will affect the system performance dramatically.

In addition, multimedia technologies are usually applied in multi-task systems for video processing. These technologies have not only provided existing applications like desktop video/audio but also spawned brand new industries and services like digital video recording, video-on-demand services, high-definition TV, digital home sever, etc. It generally needs huge memory requirement for high quality or multiple scalable level video processing. The memory system needs to provide enough memory space and high data bandwidth for satisfying the video real-time requirement.

In order to provide huge bandwidth requirement for multi-task system, a multilevel memory hierarchy is a well-known design methodology. A well-organized memory hierarchy system can have fast memory access time provided by highest hierarchy level memory and cheap cost per storage bit provided by the lowest hierarchy memory. In addition, the data transfer to off-chip memory is especially important due to the scarce resource of off-chip bandwidth. As many recent studies have shown, the off-chip memory system is one of the primary performance bottlenecks in current systems.

As the number of processor elements in SoC system increases quickly, the data communication and memory access traffic problem are more and more serious for constructing multi-task or multi-core systems. Especially for the system that have video process requirement such as digital TV, digital home sever or mobile devices. With video processing, a large amount of data needs to be processed and finished in a tightly bounded time. Higher resolution of video processing requires more memory bandwidth for real-time requirement. Furthermore, modern video coding schemes such as scalable video coding (SVC) [1.4] or multi-view coding techniques [1.5] require more memory bandwidth than the conventional coding scheme. Additionally, in a multi-task system, different processor elements may have quite different memory behavior. For instance, video processor element requires large memory but the wireless processor element may be not. It will result in bad memory utilization if traditional memory system is applied in multi-task platform.

How to manage and utilize the memory is the most important issue for constructing a multi-task platform. 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 processor elements in a system. Therefore, a memory-centric on-chip data communication platform with on-demand memory system will be proposed in this thesis. The on-demand memory system provides high bandwidth and low power memory accesses for a multi-core platform by powerful memory management units (MMUs). Furthermore, MMUs can support that different memory resources can be assigned for different processor elements according to the memory behavior. Moreover, when decoding the video frames, video decoder generally has have regular memory access characteristics. According to the regular behavior, some techniques can be used for improving the decoding performance.

1.2 Contributions

In this thesis, a memory-centric on-chip data communication platform is presented for merging heterogeneous processor elements into a system, and applied to wireless video entertainment systems. In this platform, on-demand memory system is constructed for dynamically allocating memory resources and efficiently managing

memory accesses. The contributions of on-demand memory system will be introduced as following.

A. Buffer borrowing mechanism for data communication

In order to reduce the stall 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

In multi-task system, different processor elements (PEs) may have different memory requirements at runtime. 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 (EMI) for DDR3 DRAM

Modern DDR3 DRAM device is applied for supporting huge data storage. An efficient external memory interface for DDR3 DRAM is constructed in this work.

D. Inter-layer Pre-fetch (IPS) for SVC

In wireless video entertainment systems, SVC technique is used for video coding. IPS is proposed to reduce the miss rate when decoding frames by SVC.

E. Efficient Address Translator (AT) for SVC

A suitable DRAM data allocation for frame data is presented. It can improve the DRAM access efficiency for processing SVC.

1.3 Organization

The organization of this thesis is depicted as following. The related researches of memory systems will be introduced in Chapter 2. In the chapter, the concept of memory hierarchy, the previous work of the reconfigurable cache, DRAM architecture, basic operation of DRAM, DRAM controller and modern DRAM development will be described.

And then, Chapter 3 presents a memory-centric on-chip data communication

platform with on-demand memory system for wireless video entertainment application. The development of the wireless video entertainment systems and the concept of on-demand memory system will be introduced.

Chapter 4 presents the design of Distributed and Centralized memory management units (MMUs) which are applied in memory-centric on-chip data communication platform. Buffer borrowing mechanism in distributed MMUs and adaptive cache scheme in centralized MMU are proposed for optimizing the memory resources utilization dynamically in on-demand memory system. To communicate with external memory, an efficient external memory interface will be presented. In addition, the memory latency and energy measurement methods will be introduced in Chapter 4.

Subsequently, a pre-fetch and DRAM data allocation schemes are proposed in Chapter 5 to improve the memory energy efficiency of Scalable Video Coding (SVC) functional block in wireless video entertainment systems. Pre-fetch command generator and address translator are applied in Distributed MMU and Centralized MMU, respectively. With these proposed schemes, the memory energy consumptions including on-chip cache and off-chip DRAM can be reduced significantly for decoding the video frames by SVC function. Finally, the conclusion and future work will be discussed in Chapter 6.

Chapter 2

Related Researches of Memory Systems

In this chapter, the related research of memory system including cache and DRAM systems will be introduced. Furthermore, the previous work of reconfigurable cache and DRAM controllers will be introduced, too. Firstly, the concept of memory hierarchy will be described in section 2.1. After that, the overview of cache and DRAM systems will be described in section 2.2 and 2.3, respectively.

2.1 Memory hierarchy

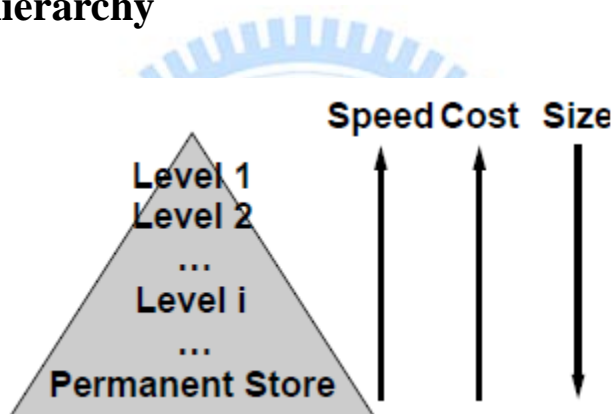


Fig.2. 1 Memory hierarchy

In computer or SoC systems, memory elements are necessary for data storage, and the most important development concept is memory hierarchy because a well-organized hierarchy enables the memory system to have both advantages simultaneously which are the fastest memory access time and the cheapest cost per storage bit. The memory hierarchy is based on a principle of locality including temporal and spatial locality. In general, the memory hierarchy is described as a pyramid which is shown in Fig.2. 1 [2.1]. The higher levels have better performance than the lower levels, but the cost per bit is on the contrary. In ideal, the processor element 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. The list of the performance and energy consumption is shown in Table.2.1. So far, there are no storage element can provide low cost, high bandwidth and low latency simultaneously. The memory hierarchy is built to hide the

negative characteristics and gain the positive characteristics of these memory technologies.

Technology	Bytes per Access (typ.)	Latency per Access	Cost per Megabyte ^a	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

Table.2.1 Cost-performance for various memory technologies

According to different system requirement, the design and configuration of memory hierarchy will differ. In the following sections, the previous work of the adaptive cache design and the external memory controller will be introduced.

2.2 Cache

2.2.1 An overview of Cache Memory

In the memory hierarchy system, cache plays an important role because it is the first level of the memory hierarchy. The basic operation can be illustrated by Fig. 2. 2. Assume the address width of the processor element is 32-bits, the address can be divided into three parts which are offset, Index and Tag. According to the Index value, the address selects a cache line and then check out the Tag. If the Tag of the address is equal to the Tag bits recoded in the cache line and the valid bit is 1, it means the wanted data is in the cache. The data will be delivered if hit. Note that the valid bit is used to indicate whether an entry contains a valid address or not. If the Tag is different or the valid bit is 0, it means that no requested data in the cache. The wanted data may be stored in the lower level memory. When the wanted data is found in the lower level, it would be written back to the cache and update the Tag entries.

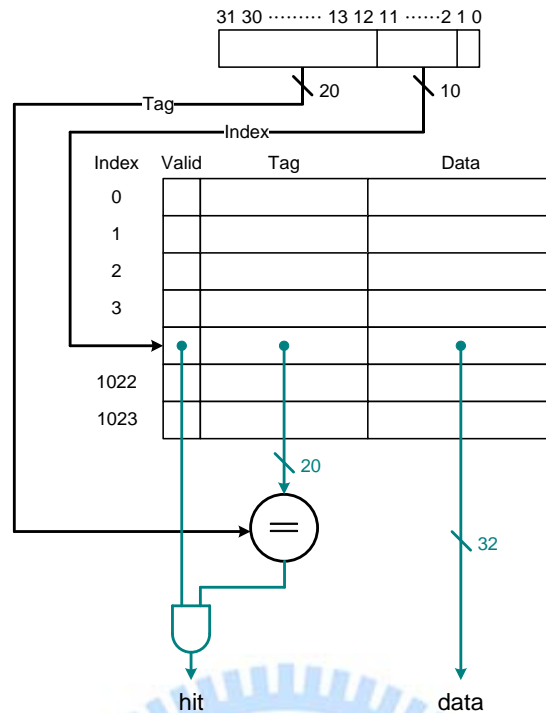


Fig.2. 2 A simple cache memory.

The mapped structure of the above example is called direct mapped because all the memory block address is directly mapped to a single location in the cache. Another extreme mapped method is called fully associative mapped which the memory block can be placed in any location in the cache. To find a wanted block in a fully associative cache, whole entries in the cache must be searched. The hardware cost significantly increases because it needs more number of parallel comparators. The middle mapped scheme between direct mapped and fully associative is called set associative. Fig.2. 3 shows the examples of different associativity structures for a four-block cache.

1-way set associative
(Direct mapped)

	Tag	Data
0		
1		
2		
3		

2-way set associative

	Tag	Data	Tag	Data
0				
1				

4-way set associative
(fully associative)

	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								

Fig.2. 3 A four-block cache configured as direct mapped, two-way set associative, and fully associative.

2.2.2 Reconfigurable Cache Techniques and Improvements

The best configuration of the cache on a system can be distinct from different application characteristics and design constraints [2.2]. Since no cache organization can fulfill the requirements of all applications [2.3], one way to overcome this problem is to create reconfiguration capabilities in the cache. Reconfigurable caches need some additional mechanisms that enable the on-chip SRAM cache to be dynamically partitioned and reused for other processor element. The aspects of the cache organization can be categorized according to different partitioning method, data consistency process, reconfiguration policy and the reconfigurable cache level [2.4]. In the following subsections, the basic concept of these cache organizations and previous works of the adaptive caches will be introduced.

2.2.2.1 Cache Partition methods

In order to resizing the cache size, the SRAM storage partition mechanism is a key challenge in designing a reconfigurable cache. There are several partition methods shown in below.

Associativity-based partitioning

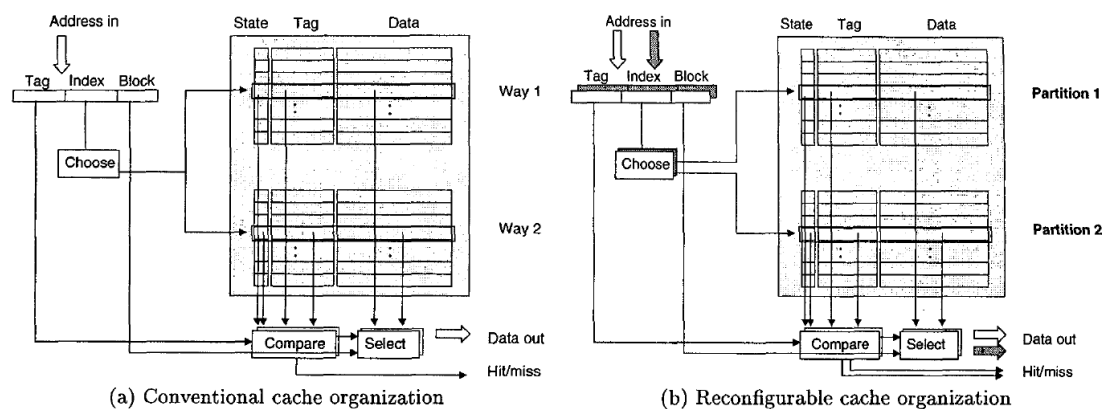


Fig.2. 4 Associativity-based partitioning organization for reconfigurable caches

The associativity-based partitioning divides the reconfigurable cache into partitions at the granularity of ways of the traditional cache [2.4]. Fig.2. 4 shows the example and the comparison with conventional set-associative cache. This partitioning approach has several advantages. First, the organization only requires few changes to the current set-associative cache organization. The second one is that the different

requests which address to 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.

Albonesi [2.5] proposed a selective cache ways method for on-demand cache resource allocation. The technique disables a subset of the ways in the set associative cache to have lower energy consumption. Parthasarathy [2.4] presented the reconfigurable caches for media processing applications, and the associativity-based partitioning mechanism was selected. In contrast to simply turning off some partitions in [2.5], it suggests using the partitions for alternate processor activities to enhance performance. Zhang [2.6] proposed the highly configurable cache architecture for embedded systems. The basic principle is also base on associativity-based partitioning. The cache used a *way concatenation* technique so that it can be configured by software to be direct-mapped, two-way or four-way set associative.

Overlapped wide-tag partitioning

Another partitioning method is called overlapped wide-tag partitioning [2.4]. The different part to the conventional cache is indicated by the dark-shade regions shown in the Fig.2. 5. This partitioning increases the tag array bit size to support the maximum tag bit variation with various partition sizes. According to this organization, the size of partition can potentially be any size, but generally the size would be limited to be powers of two to have simpler implementation. The main drawback of this partitioning is that the data in all blocks requires be flushed when the resizing occur because the mapping of the address has been changed.

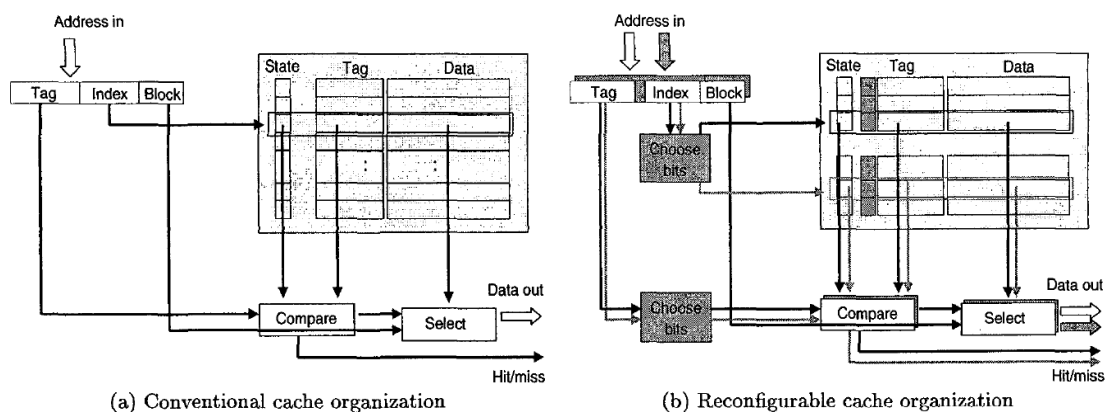


Fig.2. 5 Overlapped wide-tag partitioning organization for reconfigurable caches

Yang [2.7] proposed an i-cache design that the cache size can dynamically be

changed, and the cache partitioning method of resizing is similar to the overlapped wide-tag partitioning. After this work, a hybrid selective-sets-and-ways cache organization was proposed [2.8] to enhance the configuration flexibility. Fig.2. 6 and Fig.2. 7 show the basic structures of selective-ways and selective-sets resizable caches respectively. In addition, Ravi Iyer [2.9] proposed a CQoS : a work on heterogeneous caches regions. In its work, the set partitioning technique is applied in his organization schemes.

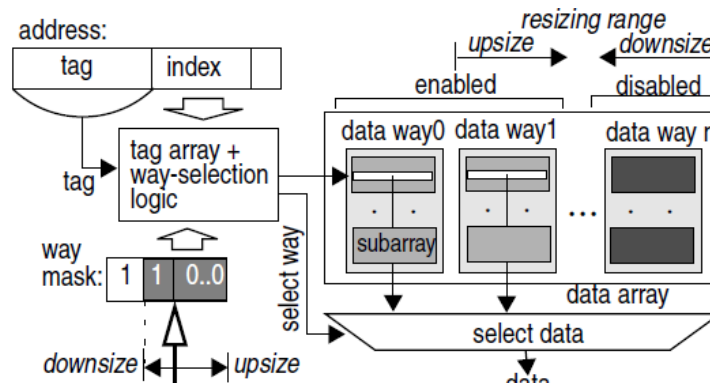


Fig.2. 6 A selective-ways organization.

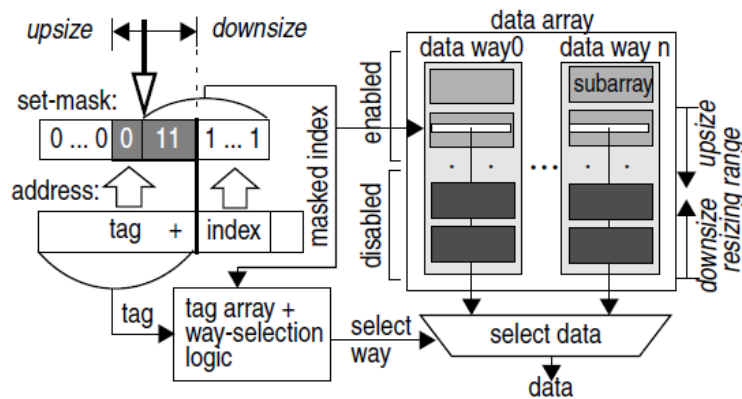


Fig.2. 7 A selective-sets organization.

Molecular-based partitioning [2.10]

In many partitioning works, the cache SRAM is divided into several individual sub-caches. We categorize these partitioning methods as the Molecular-based partitioning. The separated caches could dynamically be reorganized according to different application requirements. Vardarajan presented the Molecular Caches which are composed of many small and reconfigurable building blocks called Molecules [2.10]. The design can dynamically adjust the configuration of the cache capacity, set-associativity, and line size. In their design, 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. The physical organization of molecules is shown in Fig.2. 8. The ‘M’ is the symbol of a molecule. 4-8 tiles are grouped into a tile cluster, and every cluster is associated with a tile controller named Ulmo. It processes the coherence traffic and tile-misses between clusters. Fig.2. 9 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.

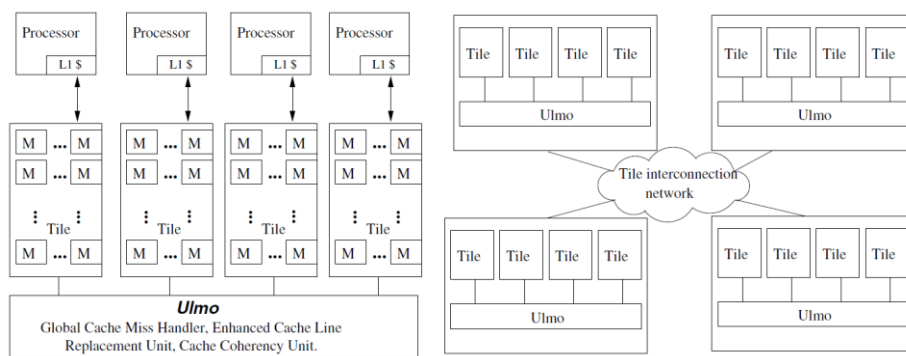


Fig.2. 8 Tiles - A physical organization of molecules.

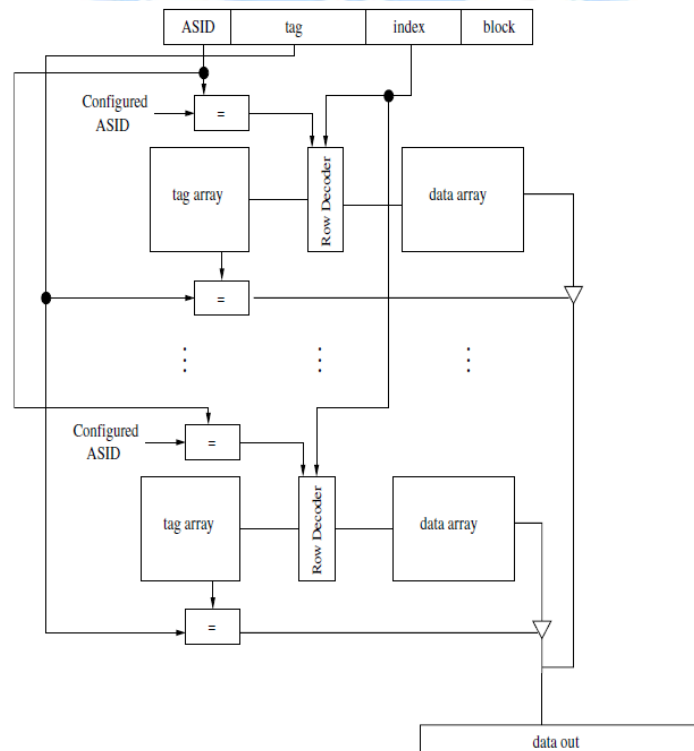


Fig.2. 9 Different steps in cache access in the molecular cache

Kaseridis [2.11] proposed a Bank-aware dynamic cache partitioning for multicore

architectures. A typical allocation in their design is shown in Fig.2. 10. According to different memory resource requirement, the L2 cache banks are separated into eight parts for eight cores.

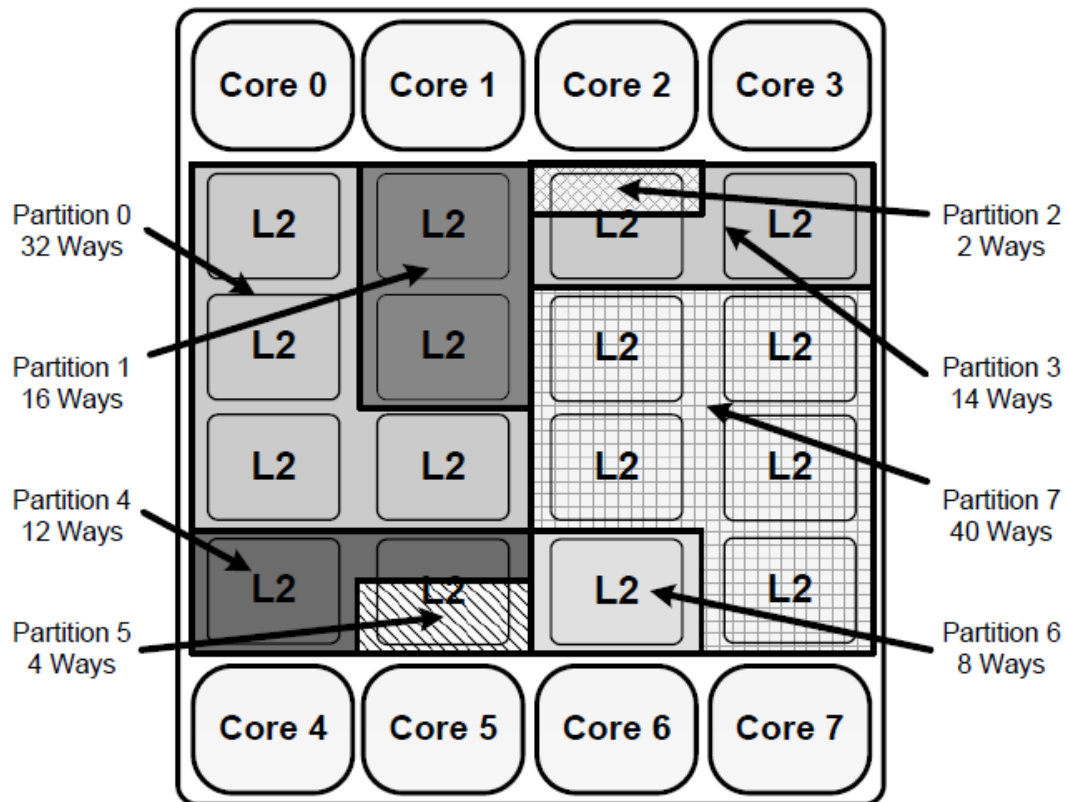


Fig.2. 10 An example of typical CMP cache partitioning

The sub-caches can be heterogeneous caches. In the CQoS work presented by Ravi Iyer [2.9], the heterogeneous caches technique has been used in its platforms. In addition, Benitez [2.2] presents the Amorphous Cache (AC) which is a reconfigurable L2 on-chip cache, and it is organized by heterogeneous sub-caches. Fig.2. 11 shows the AC structure and maximum cache size is 2MB. There are six sub-caches which the sizes are ranging from 64KB to 1MB. The AC uses configuration registers to organize the cache into different cache size and number of way set-associative. It has eighteen configurations because the cache size can be range from 64KB to 2MB and the set-associative can be 4, 8, and 16-ways.

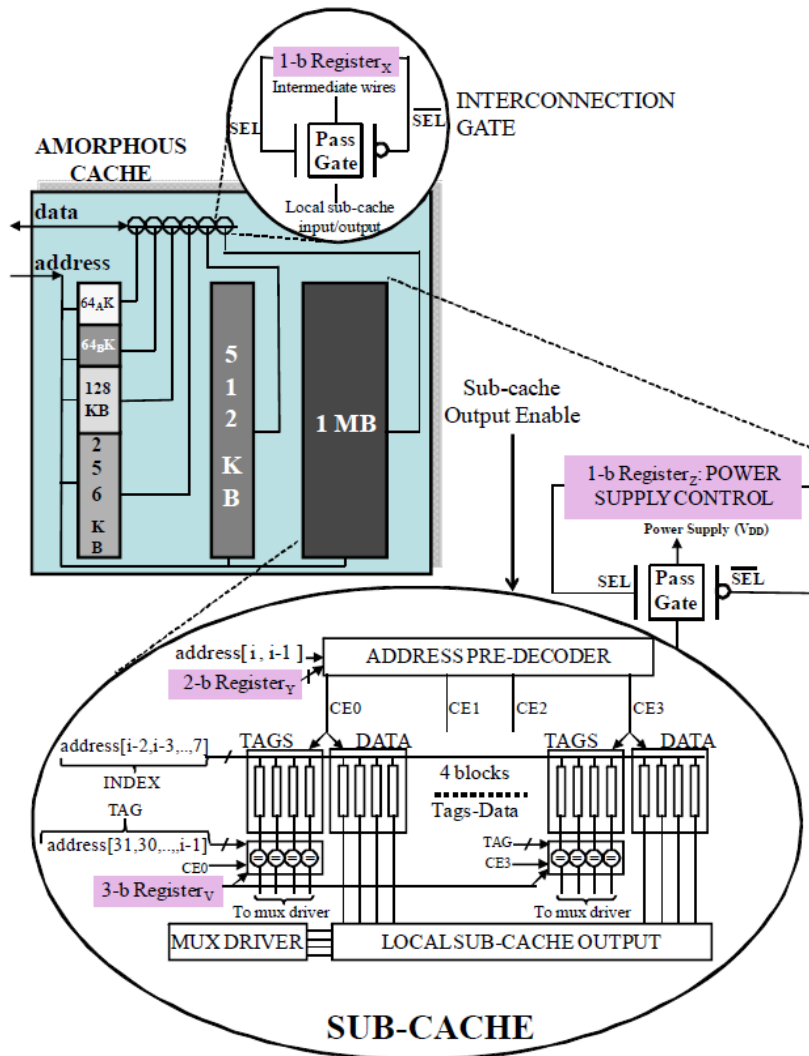


Fig.2. 11 Basic structure of the reconfigurable Amorphous Cache for processors with large on-chip cache memories

2.2.2.2 Data Consistency

Another problems need to conquer is data consistency after resizing the cache. Reconfigurable caches need a mechanism to ensure that the data which belongs to a particular processor element resides only in the partition associated with that particular activity [2.4]. Generally there are two approaches for the data consistency which are cache scrubbing and Lazy transitioning. The concept will briefly be introduced as follows.

Cache scrubbing

Cache scrubbing scheme moves all valid data to the new partition parts or lower levels of memory when the reconfiguration happened. At the time of reconfiguration,

this approach requires examining all the locations of the cache to check for their validity and performing suitable actions on valid data [2.4]. Cache-scrubbing would induce big overhead because of the huge data access. But it can be acceptable when the reconfiguration is infrequent.

Lazy transitioning

When the reconfiguration happened frequently, the other suitable scheme is that the data is lazily moved into its correct partition parts only when it is accessed. In order to achieve the scheme, it needs additional cache line information to indicate the user of the corresponding cache line. According to this information, the access which address to this cache line can be checked. Note that if a miss occur in the appropriate partition, other partitions must need to be checked because the data may laze in other partitions. This method can avoid high overhead with moving large amounts of data when the reconfiguration happened, but it need more state storage and may increase the contention for the other SRAM partition parts.

2.2.2.3 Reconfiguration Policy and Detection

A reconfigurable cache needs a detection mechanism and reconfiguration policy to determine when to reconfigure. The cache reconfiguring strategy can be *static* or *dynamic* strategy. The cache resizing is done prior to the application execution when using static strategy. Instead of the static strategy, dynamic strategy reconfigure the cache organization when the application runtime. It needs a detection mechanism to dynamically monitor the performance and energy dissipation to determine when to reconfigure and what organization to be chosen. The mechanism can be software or hardware controlled.

According to different organization of the configuration caches, the reconfiguration policy and detection mechanism may be different. Albonesi [2.5] used a software-visible register, called *Cache Way Select Register(SWSR)*, to enable/disable the particular ways. 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. According to the measurement, it can select a suitable way organization for the cache. Kaseridis [2.11]

used the Mattson's stack distance algorithm and the concept of *Marginal Utility*, which originated from economic theory, to be the assignment policy in bank-aware cache partitioning. Benitez [2.2] proposed a *Basic Block Vectors*(BBV)-based tuning technique to trace the loop characteristics of the program in the runtime, and it dynamically learned the configuration type by holding the previous CPI value.

The related works of the reconfigurable caches are shown in the Table.2. 2.

Work	Partitioning mechanism	Data consistency	Detection mechanism	Reconfigurable cache level	Application
[2.2]	Molecular-based	Cache scrubbing	Hardware controlled; Dynamic strategy	L2	General purpose
[2.4]	Associativity-based	Cache scrubbing	Software controlled	L1	Media processing
[2.5]	Associativity-based	Lazy transitioning	Software controlled; Dynamic strategy	L1	General purpose
[2.6]	Associativity-based	N/A	Software controlled; Static strategy	L1	Embedded System
[2.7]	Overlapped wide-tag	Cache scrubbing	Software controlled, Static/Dynamic strategy	L1 I-cache	General purpose
[2.8]	Hybrid	Cache scrubbing	Software controlled Static/Dynamic strategy	L1	General purpose
[2.9]	Overlapped wide-tag Molecular-based	Cache scrubbing	Software controlled dynamic strategy	Shared cache	Multi-core Network-intensive applications.
[2.10]	Molecular-based	Cache scrubbing	Software controlled; Dynamic strategy	L2	General purpose multi-core
[2.11]	Molecular-based	N/A	Software controlled Dynamic strategy	L2	General purpose multi-core

Table.2. 2 Related work of adaptive caches

2.3 DRAM

2.3.1 DRAM characteristic

Dynamic random-access memory(DRAM) 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 [2.1]. 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).

2.3.1.1 Basic DRAM architecture

DRAM architecture is usually composed of the data memories, address decoders, row buffer, mode register, data buffer. Fig.2. 12 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.

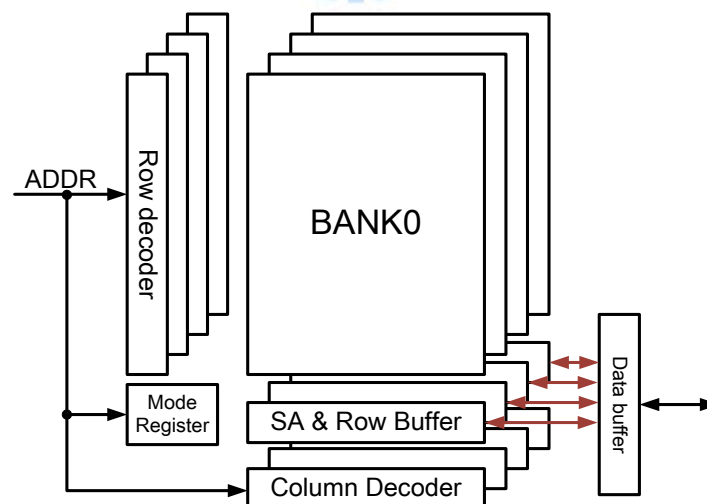


Fig.2. 12 Simplified architecture of a DRAM.

2.3.1.2 DRAM command and operation

The normal commands and its operation used in DRAM will be introduced as follows.

NO OPERATION (NOP):

The NOP command can prevent unwanted commands from being registered during idle or wait states. Operations already in progress are not affected.

ACTIVE:

This command is used to open a row in a particular bank. The row remains open for accesses until a PRECHARGE command is issued to that bank.

READ/WRITE:

The read/write command is used to initiate a read/write access to an active row, if auto precharge is selected, the row being accessed will be closed at the end of read.

PRECHARGE:

The precharge command is used to deactivate the open row in a particular bank. The bank will be available for a subsequent row access a specified time (tRP).

REFRESH:

The refresh command can be used to retain data in the DRAM.

A memory access operation, which simplified state diagram is depicted in Fig.2. 13, contains three operation including row activation (ACTIVE), column access (read/write), and precharge.

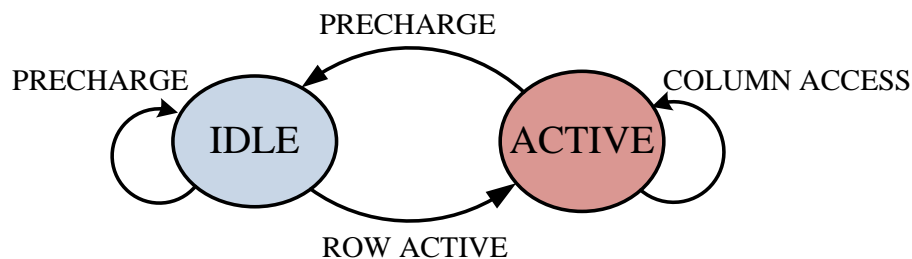


Fig.2. 13 Bank state diagram.

The active command opens a particular row in one of the bank, and copies the row data into the row buffer. The active command needs a latency period called t_{RCD} to accomplish this operation. Then, after t_{RCD} delay a column access command (read / write) can be issued to sequential access data or single data according to the burst length and burst type set in the mode register. During the t_{RCD} time, no other commands can be issued to the bank. However, commands to other banks are permissible due to the parallel processing capability of each bank. For read operation, the valid data-out from the starting column address will be available following the CAS latency after the read command, as shown in Fig.2. 14. For write command in DDR3 SDRAM, the write data must wait a write latency and then sent to the DRAM. The timing diagram is shown in Fig.2. 15. Finally a precharge command must be issued before opening a different row in the same bank.

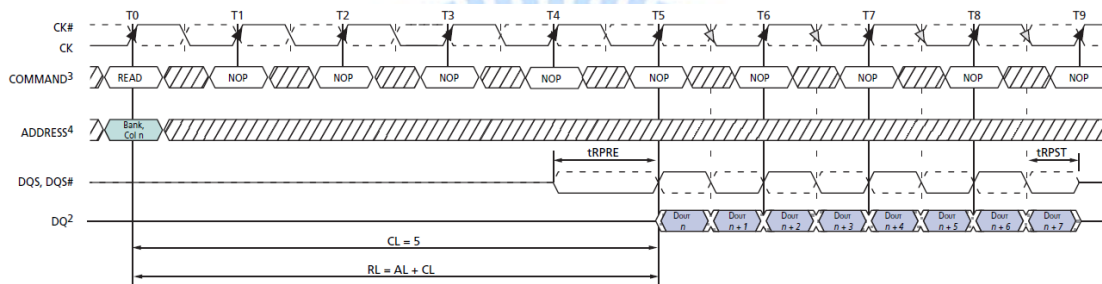


Fig.2. 14 DDR3 Read command [2.12].

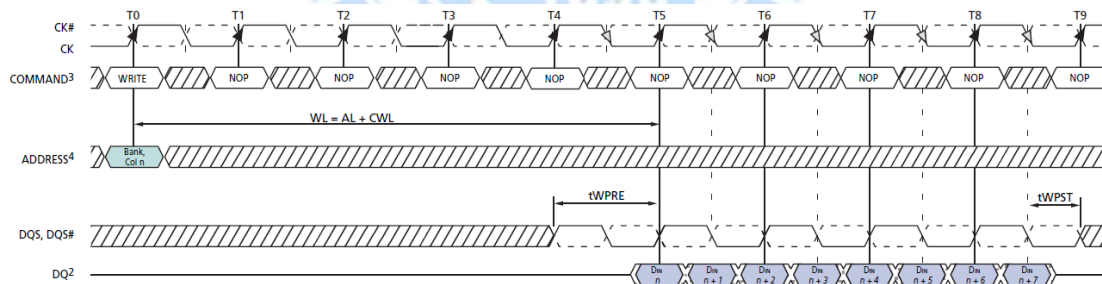


Fig.2. 15 DDR3 Write command [2.12]

2.3.2 DRAM controller techniques and Improvements

According to different applications or systems, the memory controllers can be categorized into two classes which are particular-purpose and general purpose memory controller. The particular-purpose memory controller serves one kind of specific application to reduce the memory access latency. In many multimedia applications, the advanced video processes need huge data storage space. In order to

support the real-time video environment, the system needs external memory storage to store the image frame data or motion information. But the memory access speed is much slower than the processor unit execution speed. Many researchers have shown the well memory management method according to the regular memory access behavior in video process can significantly improve the overall system performance.

Base on the different specific applications, there have several approaches been proposed to increase the efficiency of memory access for video coding applications. Kim memory interface architecture [2.13] reorganizes data arrangement in synchronous DRAM to increase the row-hit rate. Park proposed a memory node control approach [2.14] for HDTV video decoder. It uses history-based prediction to predict the next command is row-hit or row-miss. If it predicts the next command is row-miss, it will pre-charge the current bank. If row-hit, the current row will stay in the active state. The prediction is implemented by a finite state machine which shown in Fig.2. 16.

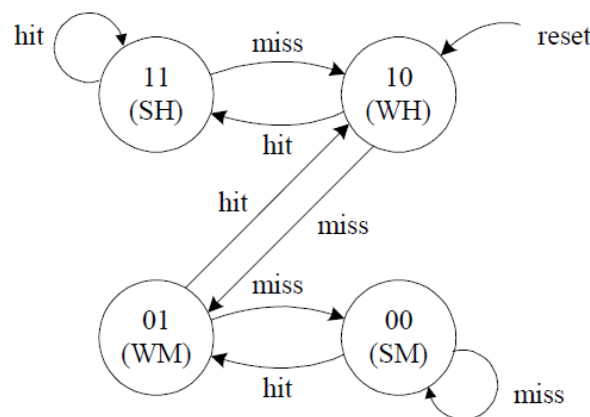


Fig.2. 16 State machine for storing page hit history information.

Chang proposed a two-layer external memory management unit [2.15] for H.264/AVC decoder. The memory management unit consists of two layers. The first layer is the address translation which provides an efficient pixel data arrangement to reduce the row-miss occurrence. The second layer is the external memory interface (EMI). In the address translation layer, the address translation machine uses a novel data arrangement which is suitable for H.264/AVC decoder to increase the memory bandwidth and reduce the power consumption. In order to minimize the number of active and pre-charge, chessboard-based arrangement memory mapping is presented as shown in Fig.2. 17. It is further compounded with the fact that Luma and Chroma are placed interleaved. The interlaced memory mapping method put the luminance

block and chrominance block in the same row of the bank. Because the decoder accesses a chrominance block after each luminance block, it doesn't need to re-active the row when accessing the chrominance block. Thus, it leads to the latency and power consumption reduced. To decrease the latency of row-miss and bank-miss status, the physical addresses produced by AT are stored in specific command FIFO. Then the command FIFO can auto-detect whether the row-miss or bank miss would happen. The architecture of command FIFO is shown in Fig.2. 18. The incoming address is compared with PAR. If bank address and column address are the same as PAR, we set hit bit of the previous command to one. It leads to auto-precharge capability turned off. Otherwise, the hit bit remains zero such that auto-pre-charge capability turns on to reduce the latency of row-miss.

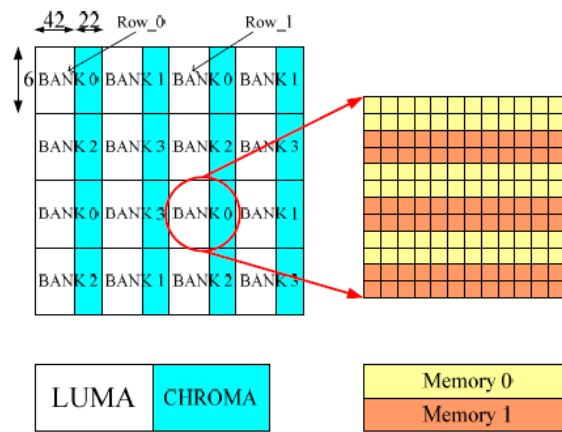


Fig.2. 17 Interlaced method

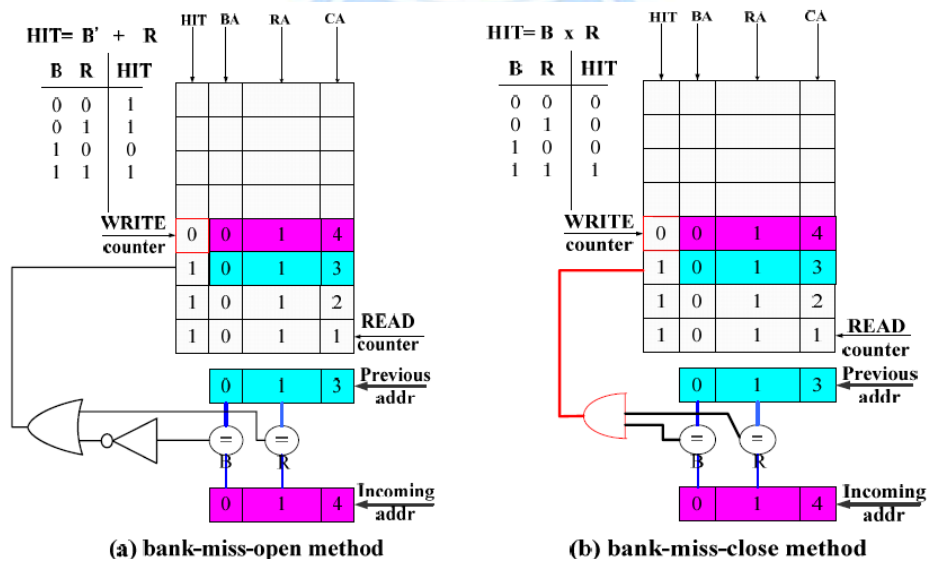


Fig.2. 18 Two architectures of command FIFO. B equals to one means bank hit. R equals to one means row hit.

Kim [2.13] and Chang [2.15] reorganize the data arrangement, Park [2.14]

proposed a history-based memory mode controller, Zhu [2.16] and Hongqi [2.17] adjust the page size. These designers are trying to reduce the total row-miss and minimize the DRAM access latency. In the advanced memory controller, rearrange data is necessary to reduce the access latency. In addition, the advance video coding standard, H.264/AVC, provides several new coding tools including sub-pixel inter-prediction, variable block size motion compensation. Although these techniques can reduce bit-rate and improve the video quality, they require huge memory bandwidth to fetch additional reference pixel for motion compensation(MC) and interpolation. Fortunately, designers can use data reuse scheme to reduce the sub-pixel MC data loading bandwidth from DRAM. Interpolation window reuse(IWR) scheme was [2.18] proposed to reduce data access for the overlapped data. Li [2.19] proposed a cache-based architecture to reuse intra-MB overlapped data, and Chuang [2.20] also proposed an IWR-liked with N-way associative cache architecture to reuse inter-MB and inter-MB overlapped data.

In order to improve the bandwidth, Kang [2.21] and Heithecker [2.22] proposed multi-channel memory controller. The concept of the multi-channel can be applied to the general purpose memory controller. In the SoC system design, a variety of processor elements integrate into a chip. Different applications have different memory needs, finding a single topology that fits well with all applications is difficult, in order to adopt a variety of the functions, flexible and adaptable memory control is more and more important in SoC systems. Furthermore, in the multi core systems, the multi-channel memory controller will be needed to support high bandwidth and provide different application memory requirement. There are many researches develop many kind of efficiency memory systems. Lee [2.23] presents a multilayer, quality-aware memory controller to satisfy different memory access requirement. Fig.2. 19 shows the configurations of different layers of the proposed memory controller. Layer 0 is called memory interface socket (MIS), it is a configurable, programmable, and high-efficient SDRAM controller for designers to rapidly integrate SDRAM subsystem into their designs. Layer 1 is quality-aware scheduler (QAS), it 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.

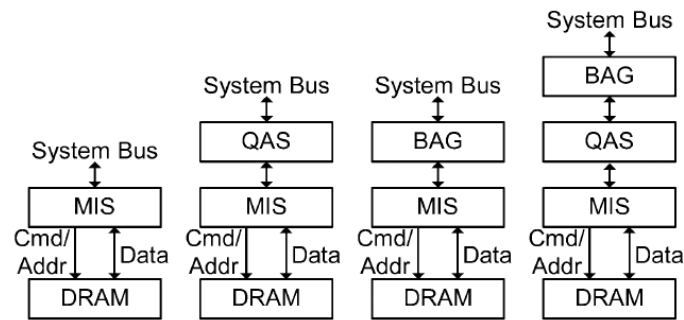


Fig.2. 19 Configurations of different layers of the proposed memory controller

Nikolov [2.24] present an efficient multiprocessor platform which separated the data communication path and memory data access path. Soinininen [2.25] presents the smart memory tile architecture to improve the memory bandwidth and performance. Ipek [2.26] proposed a self-optimizing memory controller which base on reinforcement learning concept. And in order to adjust the memory access scheduling dynamically, Zheng [2.27] proposed a ME-LREQ(Memory Efficiency-Least Request) policy.

Besides, many SoC and computer systems require DRAM devices to store data. Due to the 3-D(bank, row, column) structure, modern DRAM devices have non-uniform access latencies [2.28]. Continuous memory accesses directed to the same row of the same bank have less access latency than directed to the different row of the same bank because row conflict would not occur. Many researchers have demonstrated that rearrange and execute the memory requests out of order can significantly reduce the low conflict rate and improve the memory bandwidth efficiency. Shao [2.28] proposed a burst scheduling mechanism to maximize bus utilization of the SDRAM device. With this scheduling, memory accesses to the same rows of the same banks are clustered into bursts. Subsequently, Hu [2.29] proposed new memory access schedule algorithms overcame the starvation problem in burst scheduling.

2.3.3 Modern DRAM Development

From the day DRAM has been invented, the requests of performance accelerate

very fast. Fig.2. 20 shows the roadmap of DDR SDRAM family from 2001 to 2008. The bandwidth significantly increased in these years. For discussing the DRAM, the important issues are bandwidth, latency, and power. This section will introduce the development of DRAM that improve the performance and the future trend of DRAM.

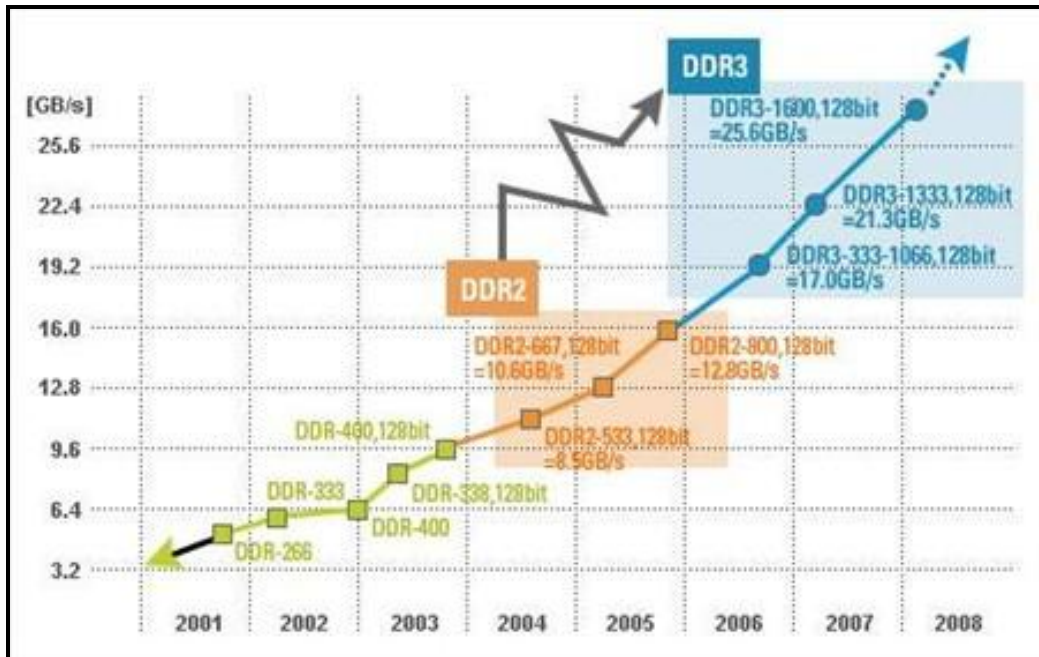


Fig.2. 20 DRAM roadmap

2.3.3.1 Bandwidth

The improvement of DRAM bandwidth has never satisfied the increasingly complicated application such as multimedia and 3D processing. To fulfill the demand for high bandwidth, various new DRAM specifications have been announced by DRAM manufacturers. The SDRAM standards supported by JEDEC [2.30] have become the mainstream of DRAM market. Several techniques have been applied on the latest standards announced by JEDED to provide users higher bandwidth.

Component	I/O bus clock (MHz)	Data transfer rate (MT/s)	Peak transfer rate(MB/s)
SDR	133	133	532
DDR	200	400	3200
DDR2	533	1066	8533

DDR3	800	1600	12800
------	-----	------	-------

Table.2. 3 The maximum transfer rate for SDR, DDR, DDR2 and DDR3

Table.2. 3 shows the maximum data transfer rate of SDR, DDR, DDR2 and DDR3 components. In SDR, the data transfer rate is equal to the I/O bus frequency, and the data is transferred at the positive edge of clock. In the DDRx standards, the data is transferred at positive and negative edge of clock. The data rate of these standards is twice as the I/O bus clock frequency. In addition, the PREFETCH technique makes DRAM be able to provide quadruple bandwidth than SDR with core frequency remains unchanged.

2.3.3.2 Latency

The DRAM response latency can directly influence the speed of the whole system. The speed of the system for the multimedia process is very essential to achieve the real-time request. So if the DRAM latency is shorter, the whole system can boost its performance. However, the situation is not as we expected. Fig.2. 21 compares the performance trend of CPU and DRAM. While CPU clock speed increases 7.65 times, DRAM latency also has a 4.6 times increase. The improvement of CPU is much faster than the improvement of DRAM. Long response latency waste its processing power on waiting and the performance is limited.

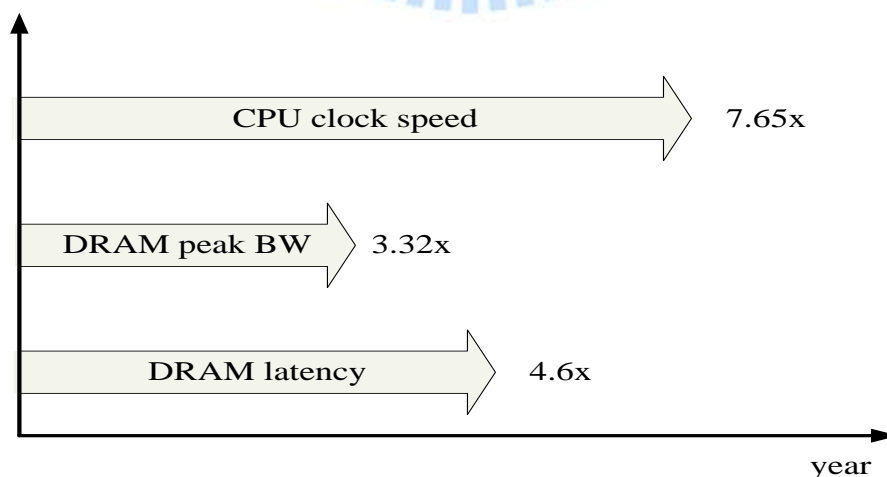


Fig.2. 21 CPU v.s. DRAM performance

One way to reduce the access latency is that parallel execute the accesses which address to different banks as much as possible. The successive accesses to the same

bank cost more latency than the successive accesses to the different banks. The timing diagrams of successive accesses with same and different bank are shown in Fig.2. 22 and Fig.2. 23 respectively. If the number of banks increases, the rate of accessing different banks can be increased. Table.2. 4 shows the number of banks of the DDR family.

	SDR	DDR	DDR2	DDR3
Number of banks	4	4	4,8	8

Table.2. 4 Number of banks for SDR, DDR, DDR2 and DDR3

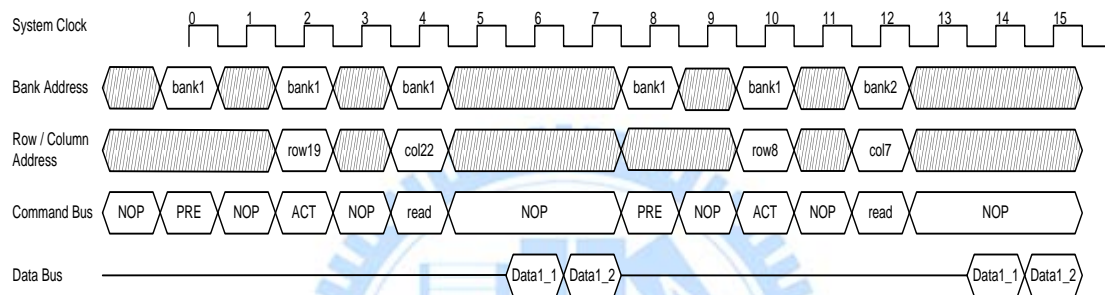


Fig.2. 22 Accesses addressed to same bank

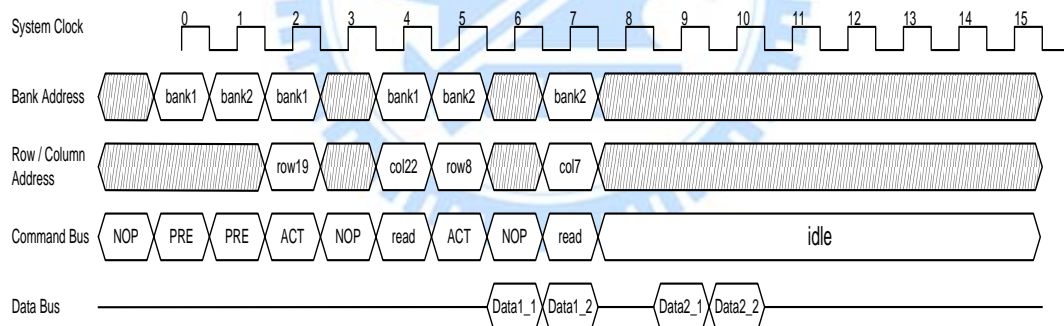


Fig.2. 23 Accesses addressed to different bank

2.3.3.3 Power

In many application of portable wireless devices such as mobile and PDA, power consumption is the significant issue because of battery life is limited. With the application of multimedia becomes popular, the request of memory size is larger. Accordingly, the designers often select DRAM to be the body memory component.

In order to reduce the power of DRAM, many products have been invented for low power such as BAT-RAM from micron [2.31] and Mobile-RAM from Infineon [2.32].

The low-power DRAM has some special features inside.

Low Operating voltage

Compare with SDR SDRAM, the operating voltage of low-power DRAM is lowered from 3.3v to 1.8v. Thus, the power consumption can significantly be decreased. For the DDR family, the supply voltage is shown in Table.2. 5.

	SDR	DDR	DDR2	DDR3
Supply voltage	3.3V	2.6V	1.8V	1.5V

Table.2. 5 Supply voltages for DDR family

Output Driver Strength

Because the low-power DRAM is designed for use in smaller systems that are typically point-to-point connection, an option to control the drive strength of the output buffers is provided. Drive strength should be selected based on expected loading of the memory bus. There are four allowable setting for the output drivers, including full strength driver, half strength driver, quarter strength driver, and one-eighth strength driver.

Temperature Compensated Self Refresh (TCSR)

Most of the time mobile devices stay in standby mode and DRAM can enter self refresh mode to save unnecessary power consumption. In the self-refresh mode, DRAM will refresh the data stored in the DRAM cell. The refresh period is inversely proportional to temperature, traditional DRAM can only support single refresh period which is the worst condition. In the low-power DRAM, a temperature sensor is implemented for auto control of the self refresh oscillator on the device. Therefore, the refresh current is decreasing while the temperature is low.

Partial Array Self Refresh

For further power savings during SELF REFRESH, the PASR feature enables the control to select the amount of memory that will be refreshed during SELF REFRESH.

Stopping the external clock

One method of controlling the power efficiency in applications is to throttle the clock that controls the SDRAM. There are two basic ways to control the clock:

1. Change the clock frequency, when the data transfers require a different rate of speed.
2. Stopping the clock altogether.

Both of these are specific to the application and its requirements and both allow power savings due to possible fewer transitions on the clock path.

The clock can also be stopped altogether if there are no data accesses in progress, either WRITE or READ, that would be affected by this change; i.e., if a WRITE or a READ is in progress, the entire data burst must be through the pipeline prior to stopping the clock.

For the full duration of the clock stop mode. One clock cycle and at least one NOP is required after the clock is restarted before a valid command can be issued.

It is recommended that the DRAM be in a pre-charged state if any changes to the clock frequency are expected. This will eliminate timing violations that may otherwise occur during normal operations.

Power-Down

Power down can occur when all banks are idle, this mode is referred to as precharge power-down. If power down occurs when there is a row active in the bank, this mode is referred to as active power-down. Entering power-down mode deactivates all input and output buffers, therefore the power is saved.

Deep Power-Down

Deep power down is an operating mode used to achieve maximum power reduction by eliminating the power of the memory array. Data will not be retained when the device enters power-down mode. Since DRAM is often used as temporary data buffers, enter DPD mode while the device is in standby mode won't cause any loss.

Chapter 3

Memory-Centric On-chip Data Communication Platform for Wireless Video Entertainment Systems

In this chapter, a memory-centric on-chip data communication platform is developed for wireless video entertainment systems. First of all, the introduction and motivation of the wireless video entertainment systems will be depicted in the section 3.1. Subsequently, section 3.2 will describe the concept of the memory-centric on-chip data communication platform. And then the development of the wireless video entertainment systems will be introduced in the section 3.3. Finally, wireless video entertainment systems will be constructed in memory-centric on-chip data communication platform, and it will be described in section 3.4.

3.1 Motivations

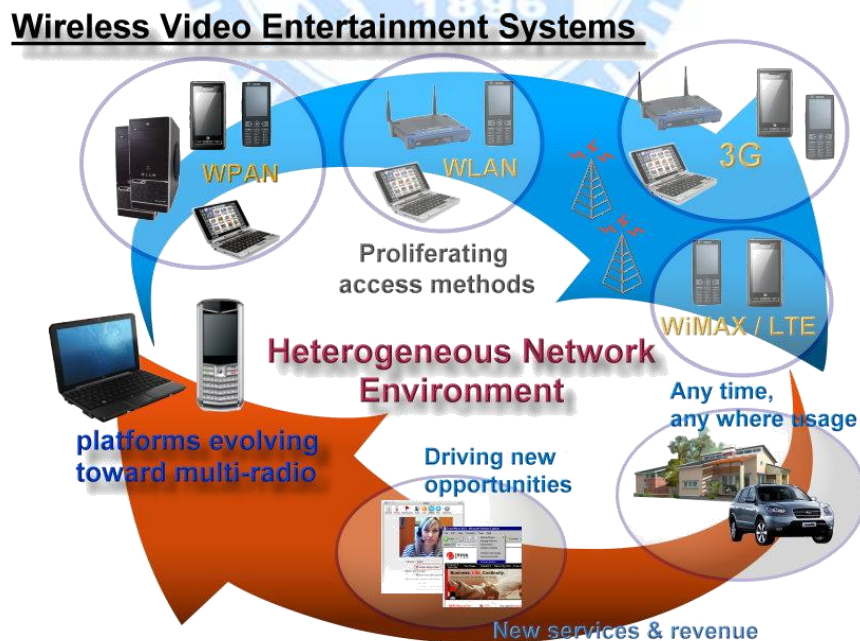


Fig.3. 1 Wireless Video Entertainment Systems

With the advancements of the wireless communication and multimedia techniques, various digital communication products are developed in our life. These modern

electronic products provide more convenient communication environment and media enjoyment for humans than those before. However, with different applications or standards, a variety of devices would be needed. Fig.3. 1 illustrates a heterogeneous network environment in our life. In recent years, merging different networks, electronic appliances and media devices into a heterogeneous integrated platform becomes an important issue that enables people enjoy their life in an more friendly and energy-efficient digital environment.

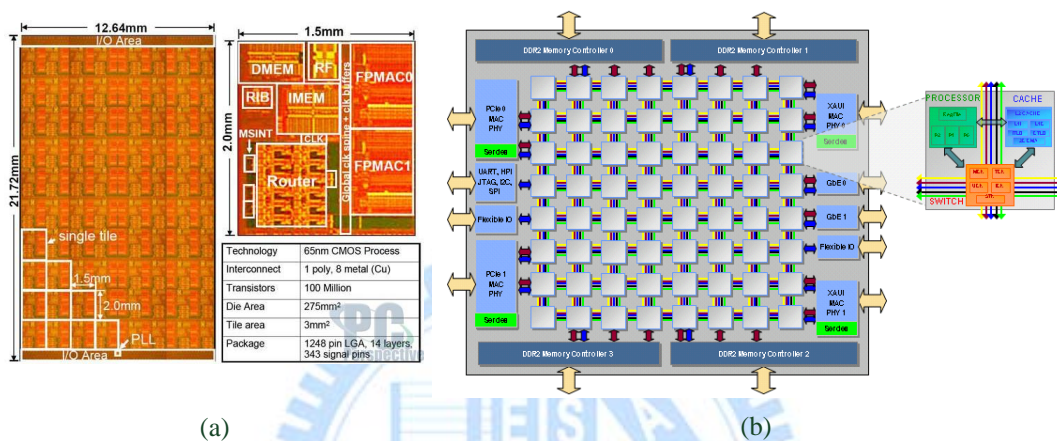


Fig.3. 2 Homogeneous multi-core platform (a) Intel Polaris (b) Tiler TILEPro64™ Processor

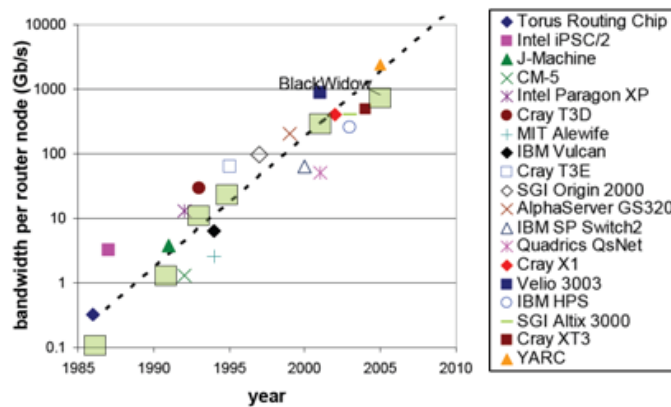


Fig.3. 3 Trend of the data transmitting bandwidth

To integrate various applications into a system, a multi-task/multi-core concept provide a typical solution to build the system. The design of multi-core platform is a popular research area recently [3.1]-[3.7]. Fig.3. 2 shows two homogeneous multi-core platforms. Intel proposed an 80-core platform as shown in Fig.3. 2(a) [3.1] and Tiler [3.2] proposed a 64-core platform as shown in Fig.3. 2(b). These multi-core platforms can execute billions of operation per second. Furthermore, the data transmitting bandwidth for the multi-core platform is increasing year by year as shown in the Fig.3. 3. However, the overall system performance could be limited by

the task partitioning, task mapping, memory resource allocation, and memory data accessing. Fig.3. 4 indicates the bottlenecks of multi-core platforms with insufficient memory bandwidth and memory capacity for supporting high communication efficiency in the multi-core systems. With ongoing development of multi-core or multi-task system, both the memory capacity and memory access bandwidth are required. Enabling multiple memory data access is necessary for improving the memory bandwidth. However, increasing the memory read/write ports not only increases the hardware complexity but also reduces the memory performance and noise immunity. Conventional memory access method cannot provide enough memory bandwidth for multi-core platform. Hence, the memory management in multi-core or multi-task platform will become more and more important. It is an essential issue that reducing additional memory access and increasing the memory bandwidth effectively. For these reasons, a memory-centric on-chip data communication platform will be proposed and introduced in the following section.

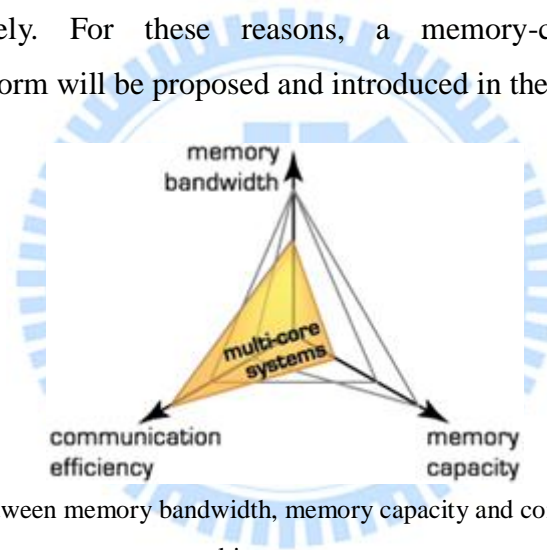


Fig.3. 4 Comparison between memory bandwidth, memory capacity and communication efficiency in multi-core systems

3.2 Memory-Centric On-chip Data Communication Platform

3.2.1 Overall Architecture

To solve the problems as mentioned above, a hierarchy memory-centric on-chip data communication platform is proposed and the architecture is shown in Fig.3. 5. Heterogeneous processing elements such as microprocessors and application-specific stream processors can be integrated in the platform. In this platform, each processor

element owns distributed memory management unit (d-MMU). The d-MMU includes local cache (D-cache and I-cache) and cache controller which can efficiently handle all memory requests generated by the processor elements. It can dynamically allocate unused space in cache for buffering the transmitting data. If processor elements need additional memory resource requirements, the centralized memory resources including centralized cache and off-chip DRAM can be used. It is controlled by a centralized memory management unit (c-MMU). It can dynamically allocate and manage the memory resources according to different memory requirements.

For the data communication between processor elements, message-passing technique is applied for this platform. The processor elements transmit/receive the data to/from others through an on-chip interconnection network. Network interface is applied to packetize the transmitted data to interconnection and de-packetizes the received data from interconnection. Furthermore, in order to have better energy utilization for green computing, the power management unit can be applied to dynamically control the supply voltage and operating frequency of each processor element for saving energy consumptions.

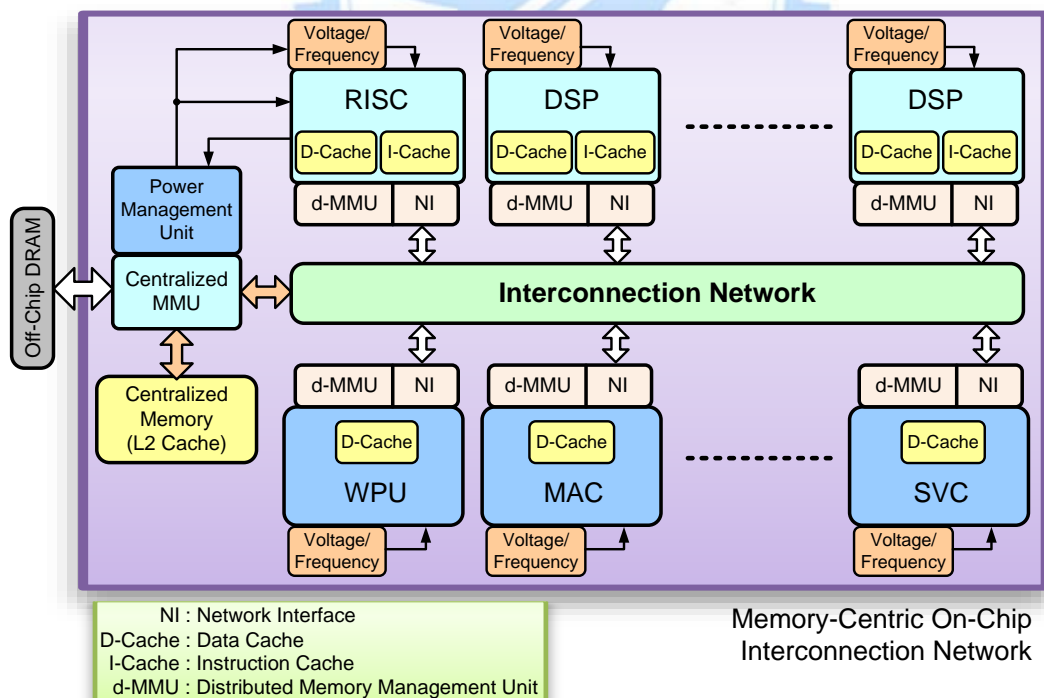


Fig.3. 5 The architecture of memory-centric on-chip data communication platform

In the heterogeneous multi-task platform, different processor elements would have quite different memory requirements with different specific functions in a platform.

For instance, the memory requirement of the video decoding is larger than that of the wireless processing unit. Moreover, different system environment factors may affect memory utilizations for the applications in platform during runtime. Different qualities of wireless channels may have different memory behavior in a wireless video integrated system. Thus, a multilevel memory hierarchy on-demand memory system is applied for this platform. The memory system enables the processing elements to own different memory resources dynamically. In the following section, the concept of on-demand memory system will be introduced.

3.2.2 Concepts of On-Demand Memory System

In on-demand memory system, a three-level memory hierarchy is constructed, and the illustration is shown in Fig.3. 6. For the first hierarchy level, distributed memory management unit (d-MMU) is applied to control the memory accesses. It includes distributed cache and cache controller for processor elements. Furthermore, in order to improve the transmitting efficiency for data communication, d-MMU can dynamically allocate unused space in distributed cache to store packet data so that the stall caused by data blocking can be prevented. The detail design of d-MMU will be described in chapter 4.

For the second level hierarchy of the on-demand memory system, centralized memory management unit (c-MMU) is constructed to provide more memory resources for processor elements. In c-MMU, a cache controller and centralized cache is included. In addition, the configuration of centralized cache can be dynamically adjusted according to the different memory requirement from processor elements. For example, if a processor element need larger memory requirement than others, it can own more centralized memory resources than other processor elements. Adaptive cache control in c-MMU controls the adaptive allocation and cache operation. In addition, unused memories can be power down to save memory power consumptions for green computing.

For supporting enough memory space, off-chip DRAM is applied, and it is the third memory hierarchy level in the system. DRAM controller is needed to access the off-chip DRAM devices. It includes an external memory interface and address translator to improve the memory access efficiency.

In the on-demand memory system, all processor elements own a private address space and can dynamically be allocated. For data switching between processor elements, message-passing mechanism is used. On-chip interconnection network in the platform is designed for data communication. Note that the thesis is focus on on-demand memory system. The design of interconnection network is not included in this thesis.

In conclusion, adaptive memory resource allocation can be achieved and the memory utilization can be improved by the memory management units. The detail organizations and the design of these memory management units are described in chapter 4.

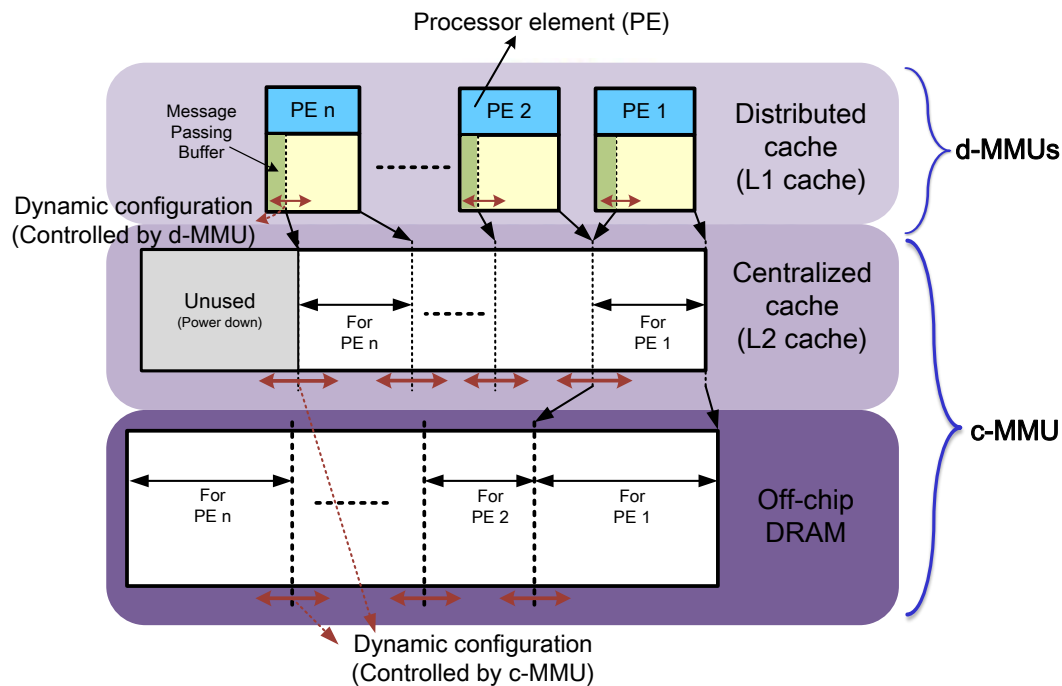


Fig.3. 6 Illustration of the memory hierarchy in on-demand memory system

3.3 Wireless Video Entertainment Systems

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. It provides an adaptable processor element to process various communications.

Many researches try to integrate the communication device and entertainment platform into a system. However, the current technologies and systems cannot effectively meet the requirements of these digital homes for some reasons [3.8]. 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. So the system bandwidth must be improved. Furthermore, the scalability, security and power are also the problems.

To solve these problems, wireless severs and multimedia processor elements can be integrated. By integrating heterogeneous elements into a platform, a variety of services can be achieved in a system. In order to serve various transmit channels, the multi-task wireless video entertainment system is shown in Fig.3. 7. Analog front-end system receives and digitizes the wireless signals. Then the data is processed by an integrated, high performance digital system.

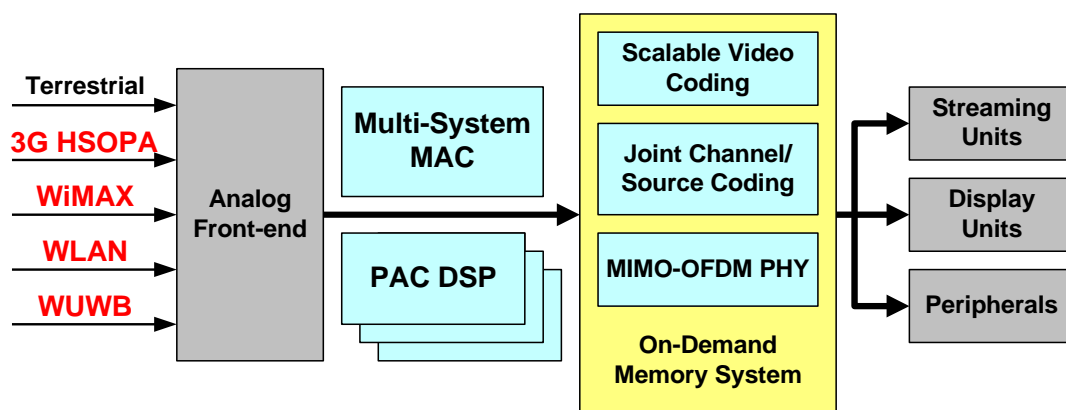


Fig.3. 7 Multi-Task wireless video entertainment system

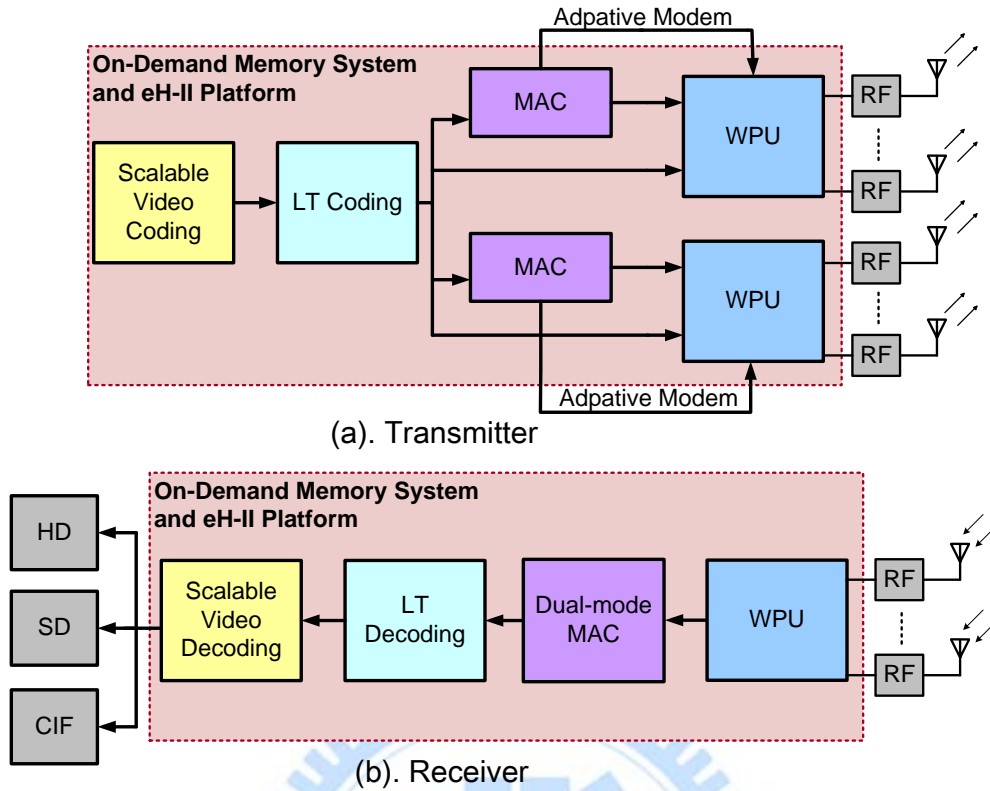


Fig.3. 8 Transmitter and receiver block diagram

In order to support various communication standards and have video entertainment for digital home service, a wireless video entertainment systems is developed. It includes four functional blocks. The block diagram of wireless video entertainment system transceiver is shown in Fig.3. 8. In this system, Scalable Video Coding (SVC), the extension of the H.264/AVC standard technique, is applied to provide spatial, temporal and quality scalability of the video sequences [3.9]. For the channel coding, Luby Transform (LT) coding, one kind of error correcting method, 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. These functional blocks are grouped into a SoC system. At current development stage, receiver system is developed in an integrated on-demand memory system as which the red block in the Fig.3. 8(b) represents. The system specification is listed in Table.4. 1. Additionally, the details of WPU, MAC, LT coding and SVC coding will be described in the following sections.

	WPU (4x4)	MAC	LT Coding	SVC
Input data rate	160MBps (4Gbx12/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)

Table.4. 1 System Specification (Receiver)

3.3.1 Wireless Processing Unit (WPU)

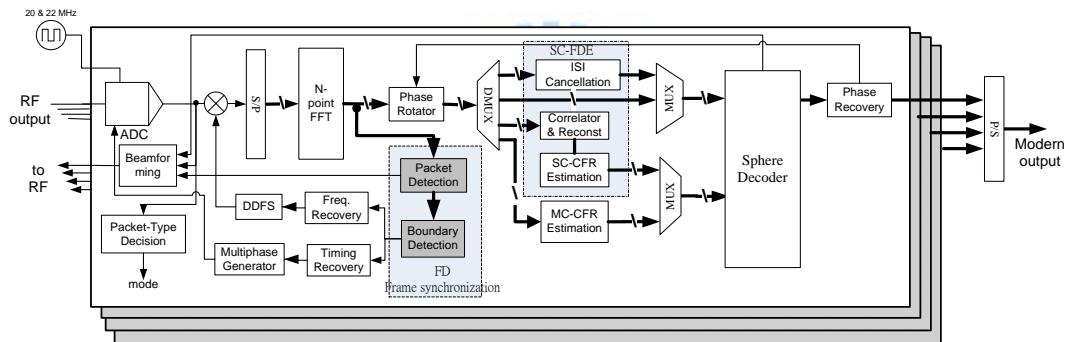
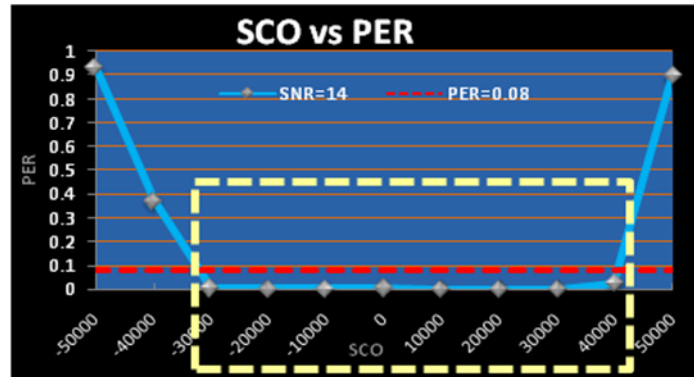


Fig.3. 9 Single-FFT Architecture for MIMO Modem

The WPU is a 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.3. 9. 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.

- 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 for DSSS non-CP SCBT
 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 as shown in Fig.3. 10.



Performance: -30000~40000ppm

Fig.3. 10 Single-FFT Architecture for MIMO Modem

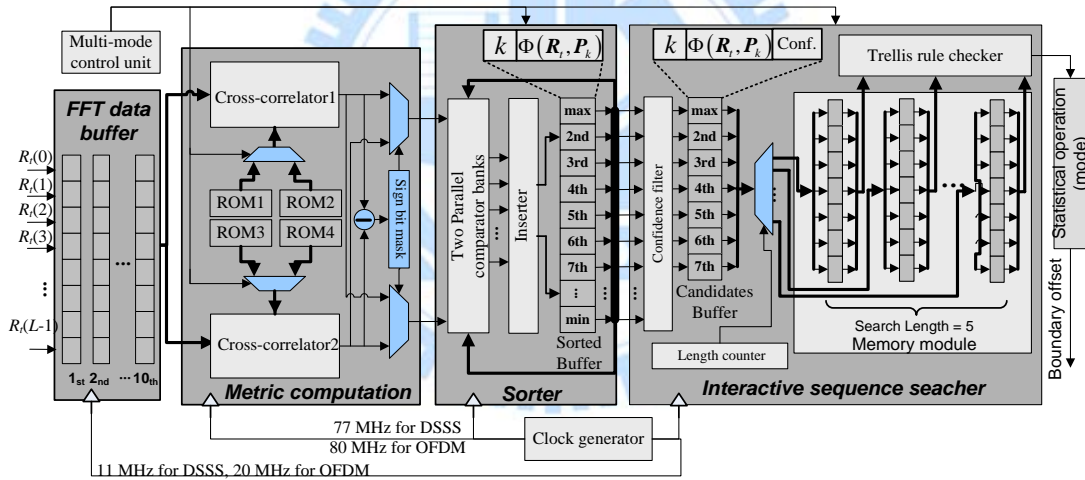


Fig.3. 11 Single-FFT Architecture for MIMO Modem

Moreover, For the FD boundary decision, it contains the following features, only 1% detection error with low SNR (<5 dB) and gigh CFO tolerance. It is a trellis-based detector, and can be used both for DSSS and OFDM different systems. Fig.3. 11 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

3.3.2 Medium Access Control (MAC)

Medium Access Control (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, we apply Object-Oriented Analysis and Design (OOA&D) principle on both protocols, identifying the common and different components between both systems. By using divide-and-conquer and bottom-up design approaches, we are able to integrate WiMAX and WiFi MAC, and facilitate reuse and performance optimization of common components between the two systems.

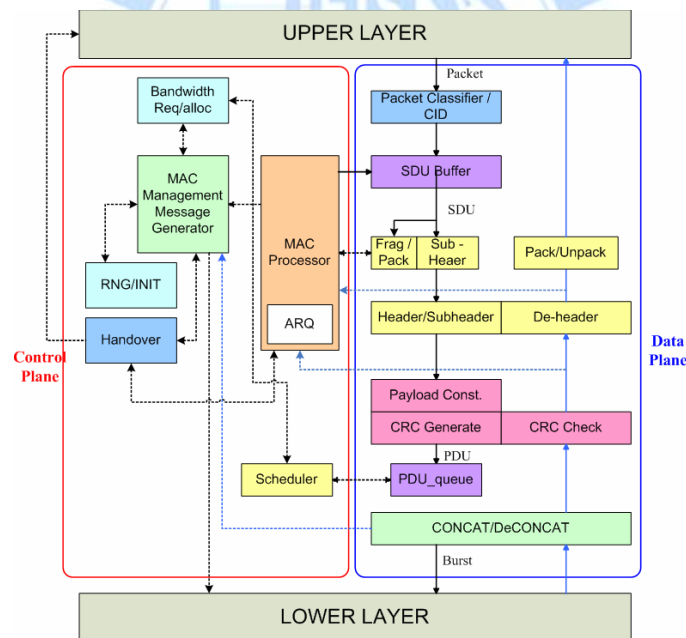


Fig.3. 12 MAC Layer Architecture

As shown in Fig.3. 12, 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 or BS-to-MS signaling.

Besides data processing performance that directly relates to software/hardware co-design, there are other factors that have great impact on 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.

3.3.3 LT Coding

LT code is a class of rateless codes. Its performance 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.

LT codes conduct BP 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.

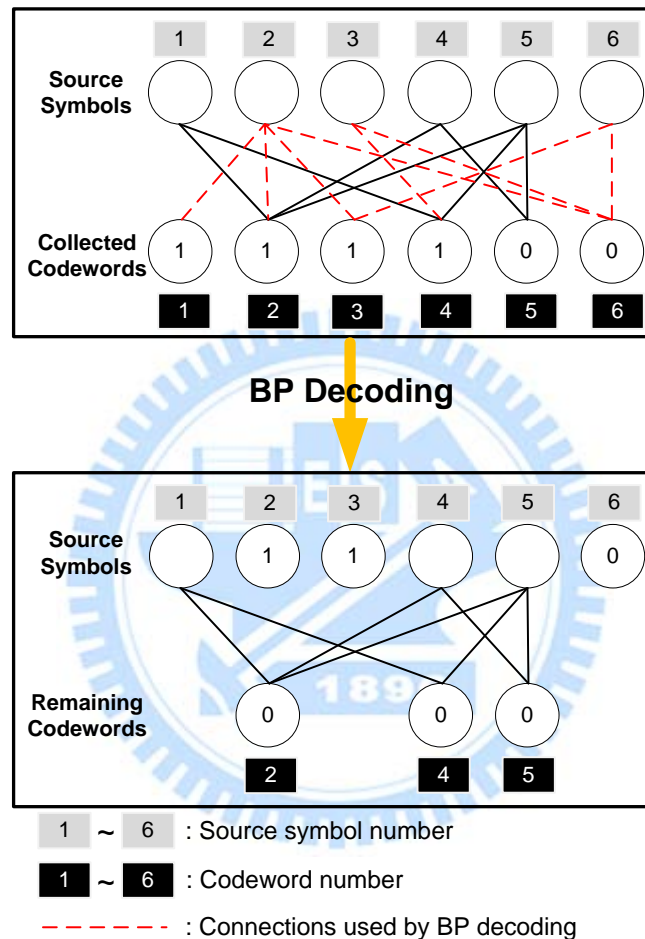


Fig.3. 13 An example of decidable codewords which BP decoding fails to decode

Fig.3. 13 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.3. 13. 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.

3.3.4 Scalable Video Coding (SVC)

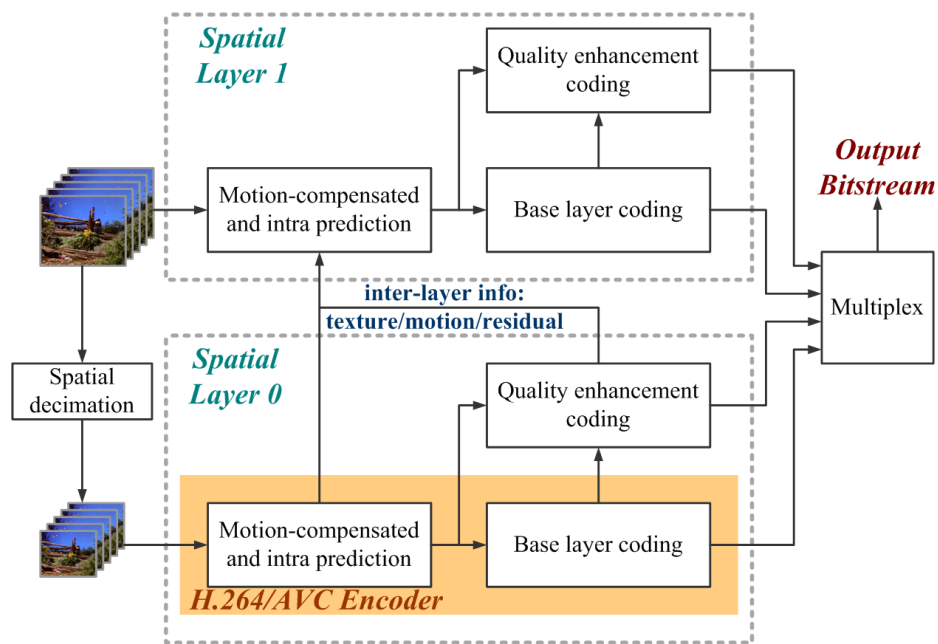


Fig.3. 14 Architecture of an SVC encoder

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.3. 14 shows an 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.

3.4 Memory-Centric On-Chip Data Communication Platform for Wireless Video Entertainment Systems

The designers try to meet efficient processing capability, merge multi-task system and use green computing concept in a system. However, when they try to integrate the heterogeneous functional blocks into a system, multiprocessing technique and multimedia process unit must be used. Furthermore, as the resolution of video processing applications becomes high, video signal processors should deal with a large amount of data within a tightly bounded time. Due to the huge data accesses, the system performance strongly depends on the memory bandwidth between processors and external memories. The system needs real-time and huge memory access requirement, but the speed gap of the memory and processor unit is large in the SoC system. Many researches are trying to minimize the speed gap. A well-organized memory management can significantly reduce the memory access latency. According to the data features of these applications, designer can find a well memory allocation method to reduce the number of memory access time and average access latency. Accordingly, for wireless video entertainment systems, memory-centric on-chip data communication platform is applied to provide a high bandwidth and satisfy enough memory requirements.

According to the receiver system as mentioned in section 3.3, the processing sequence of these multiple tasks is generally step by step. the data stream of wireless

video entertainment systems is shown in Fig.3. 15. In memory-centric on-chip data communication platform, on-demand memory system can support heterogeneous and real-time memory requirement for wireless video entertainment systems. MMUs in on-demand memory system enable the processor elements to have adaptive memory resources. Base on different memory requirement of these processor elements, centralized MMU can dynamically allocate memory resources for processor elements. With suitable memory resource arrangement for different processor elements, the execution efficiency of the streaming processing in wireless video entertainment systems can be improved.

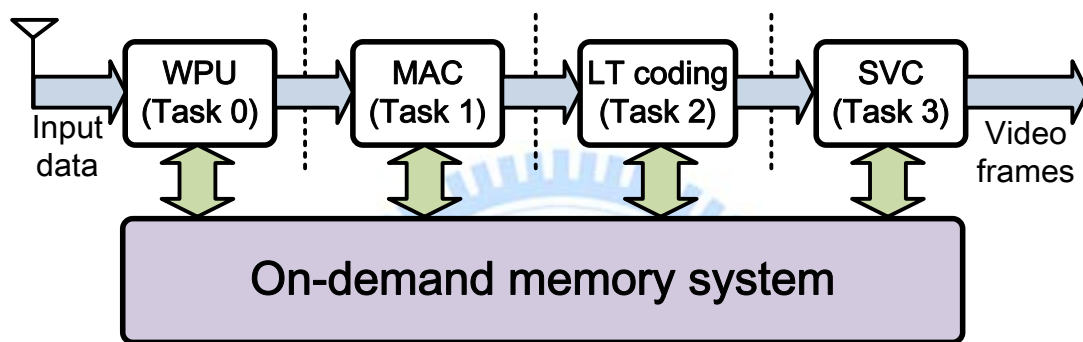


Fig.3. 15 Data stream of wireless video entertainment systems

Overall architecture of the system is shown in Fig.3. 16. The system components can be categorized into data computation, data communication and data storage. For data computation, it includes WPU, MAC, LT coding and SVC processor elements. Wrappers are applied to satisfy the specification of the pre-defined protocol. Subsequently, the other components will be introduced as follows.

For data communication, it includes network interface (NI) and interconnection network. In this system, message-passing mechanism is applied. With this mechanism, the transmitting data are packed into packets by network interface, and through the interconnection network using a pre-defined message-passing protocol. NI packetizes the transmitting data with a header indicating the data source, destination and some data information, and then transmits to the other node. It also de-packetizes the receiving data from the other processor elements. In addition, a packet queue is included in NI to store the blocking packet.

For data storage, each distributed processor element own a d-MMU, it includes a distributed cache (L1 cache) and cache controller for memory access. It also manages the cache usage. When packet queue size in NI is insufficient, d-MMU can borrow

some unused cache block for NI. In addition, c-MMU is constructed for providing more memory resources. It includes centralized cache (L2 cache) and cache controller for processor elements. The cache controller can support dynamical cache re-organization for allocating different cache resources for different processor elements. In c-MMU, a DRAM controller is constructed to efficiently access off-chip DRAM. In DRAM controller, Address translator rearranges and translates address to have an efficient memory allocation, and the memory requests enter the memory interface with command scheduling to reduce memory access latency. The detail description of c-MMU will be described in chapter 4.

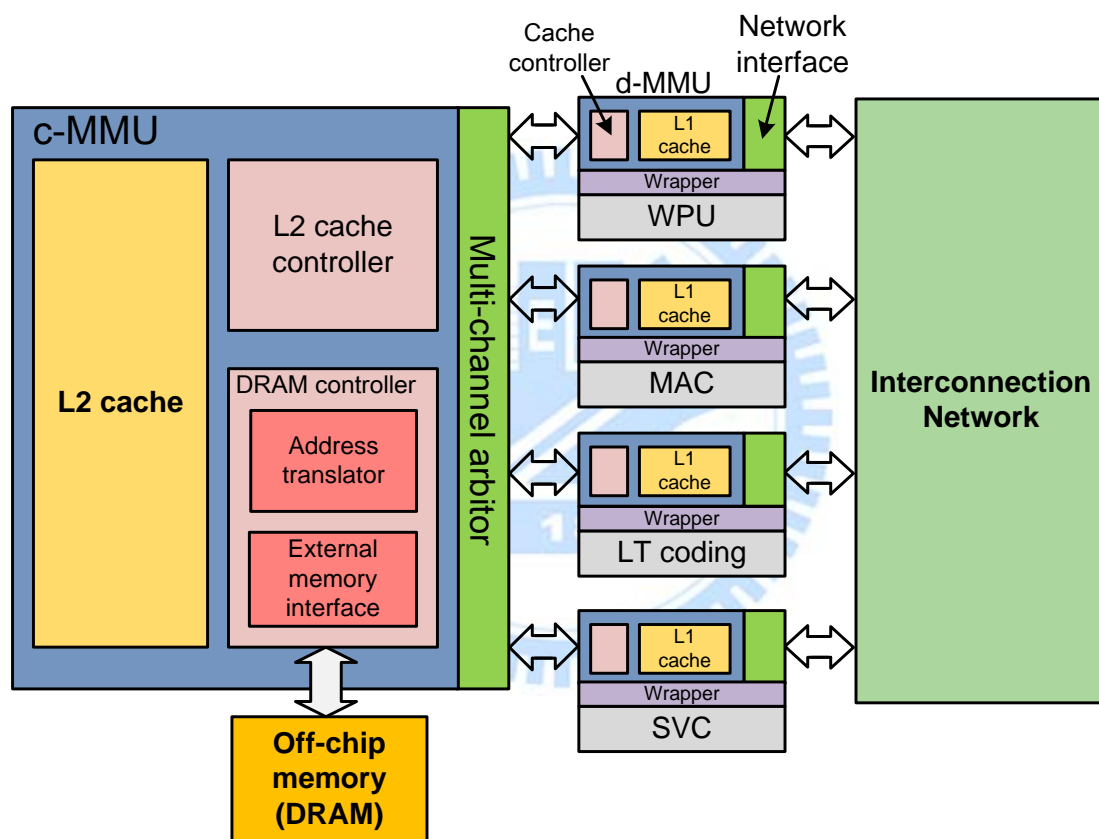


Fig.3. 16 On-Demand Memory System architecture

Chapter 4

Hierarchy Memory Management Units for On-Demand Memory System

In this chapter, the design of distributed memory management unit (d-MMU) and centralized memory management unit (c-MMU) in on-demand memory system will be depicted in section 4.1 and section 4.2, respectively.

4.1 Distributed Memory Management Unit Organization

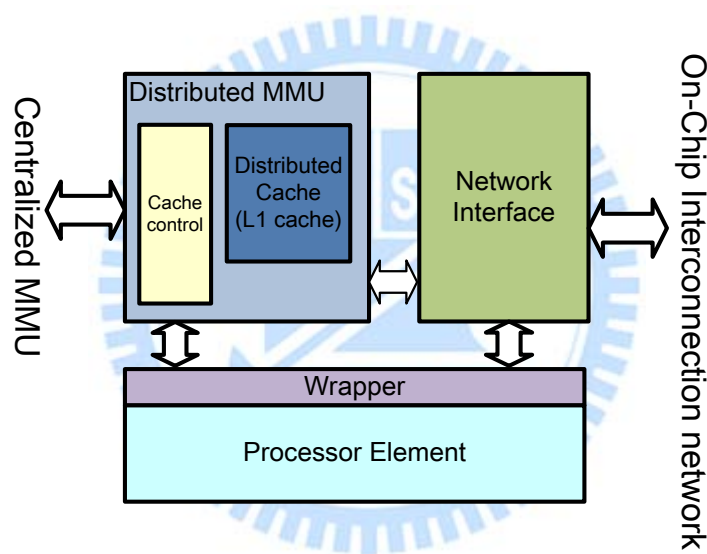


Fig.4. 1 Block diagram of a local node

A local node in memory-centric on-chip data communication platform is organized by distributed memory management unit (d-MMU), Network Interface (NI), wrapper and processor element (PE). The block diagram is shown in Fig.4. 1. To provide local memory resources for each processor element, efficient d-MMU is applied to process the memory requests. Distributed cache (L1 cache) and cache controller are included in d-MMU. Additionally, NI is designed as a bridge between processor element and on-chip interconnection network (OCIN). When the packet buffer in NI is crowded, unused cache blocks can be borrowed for buffering the blocking packets from PEs. In this section, the design of d-MMU with buffer borrowing mechanism will be described.

4.1.1 Design of d-MMU

For the memory-centric on-chip data communication platform, d-MMUs are designed for PEs to store the temporal data of their tasks. Distributed cache performs as a high level cache for the dedicated PE in the on-demand memory system. In addition, a Wrapper is applied to be an interface between processor element and d-MMU. In on-demand memory system, PE uses the burst-based memory access protocol to access memory. By this protocol, read/write operation uses burst transmission mechanism so that it can access continuous data easily. The detail memory access operation will be introduced as follows.

4.1.1.2 Memory access operation

By applied burst-based memory access protocol, the read and write operations are shown in Fig.4. 2 and Fig.4. 3, respectively. With providing start address and burst length(BL) information, processor elements can efficiently access the burst data in memory. Note that the data width is 32-bit (1word) and the addressing unit is in word by definition. Accordingly, the cache miss penalties can be hidden by burst-based memory access protocol. The cache miss would be discovered immediately when a memory burst request has been served. Fig.4. 4 provides the explanation of hiding miss penalties. In Fig.4. 4(a), a read request with miss follows by a read request with hit. The miss penalty can be hidden because the data transmit of the first read haven't been finished. For the memory write request as shown in Fig.4. 4(b), all the miss in the burst can be found immediately whenever write request comes, so it also can hide the miss processing latency of memory write.

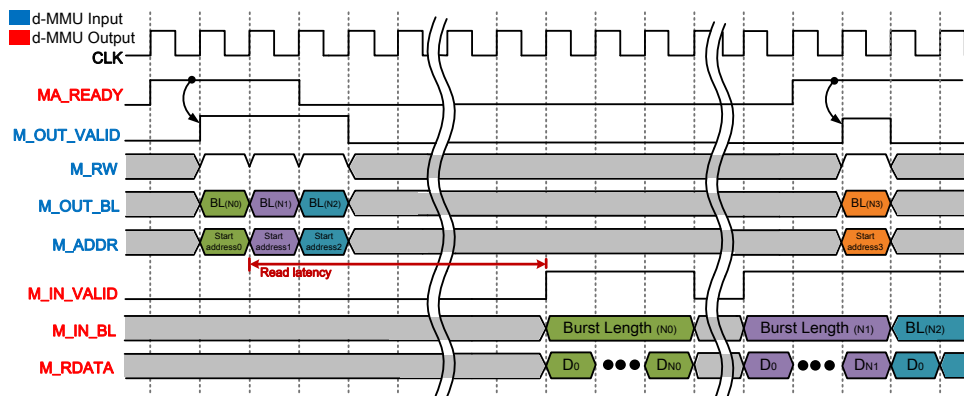


Fig.4. 2 Illustration of read operation

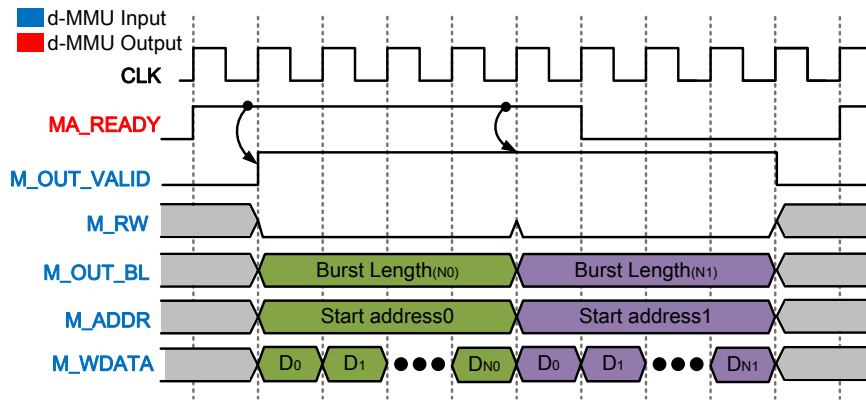


Fig.4. 3 Illustration of write operation

The maximum burst length is eight in the pre-defined protocol. In order to support that d-MMU can immediately check whether a memory burst request is miss when the request comes, two cache banks with 32 bytes (8 words) block size are allocated in d-MMU. With this allocation, a memory burst request would reference either a cache line in a cache bank or two cache lines in different cache banks, so the cache hit/miss detection can be finished in a cycle. The illustration of cache address mapping will be shown in Fig.4. 7. Note that 32Kbyte cache size and 4-way associativity configurations in each bank are applied in the illustration.

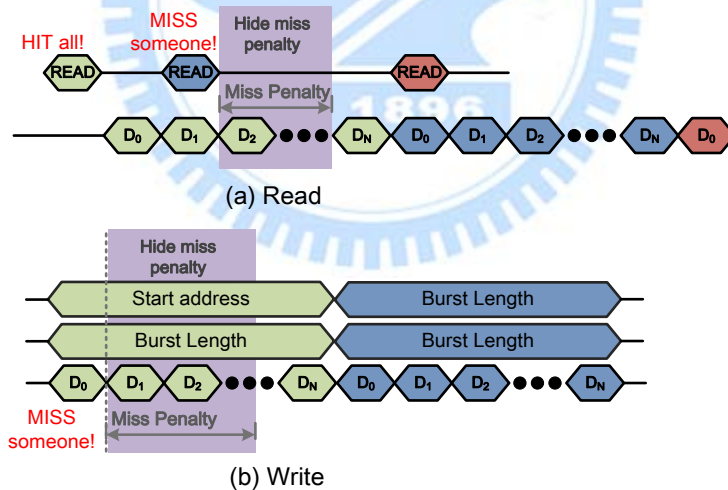


Fig.4. 4 Illustration of hiding miss penalty

Additionally, NI is designed as a bridge between the PEs and the OCIN [4.1]-[4.4]. 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 d-MMU can borrow the memory resources for buffering the blocking packets from the PEs, and the PEs can keep

computing for their tasks. Below the d-MMU with buffer borrowing mechanism will be introduced in detail.

4.1.1.2 Buffer Borrowing Mechanism

The architecture of proposed d-MMU and efficient Network Interface with buffer borrowing mechanism is shown in Fig.4. 5. The NI uses a buffering control to generate a borrowing request to the d-MMU for borrowing memory resources. And thus, the d-MMU checks the valid table and generates the borrowing address for the NI. Fig.4. 6 presents the buffer borrowing interface between the NI and d-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 d-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 d-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.

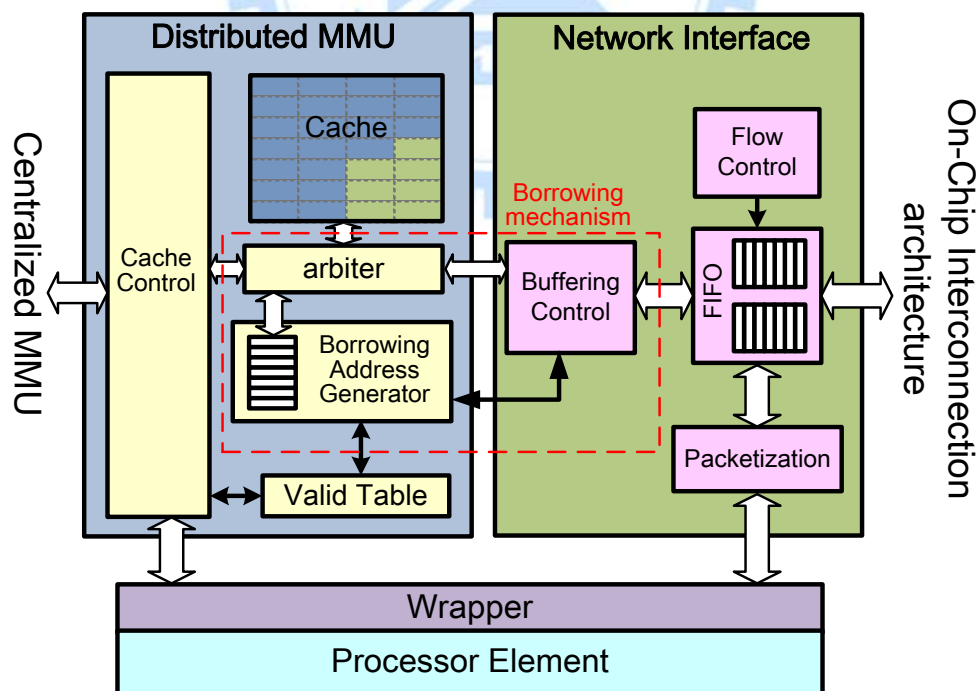


Fig.4. 5 d-MMU and efficient Network Interface

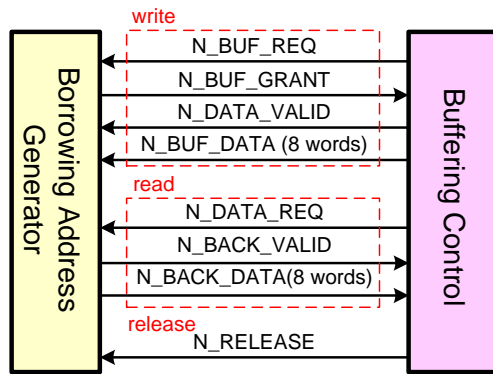


Fig.4. 6 Buffer borrowing interface between NI and d-MMU

4.1.1.2.1 Borrowing Address Generator

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.4. 7. 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 d-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 d-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.

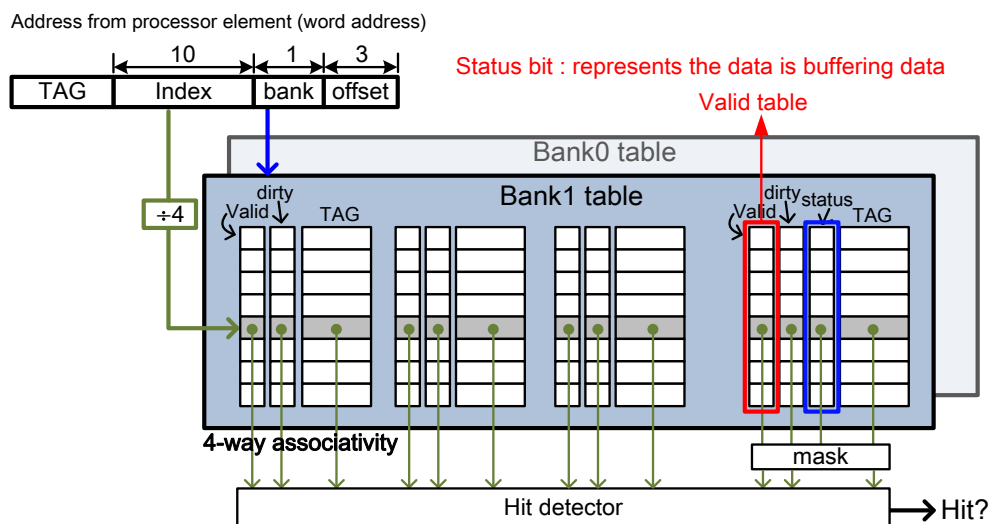


Fig.4. 7 Borrowing mechanism in d-MMU

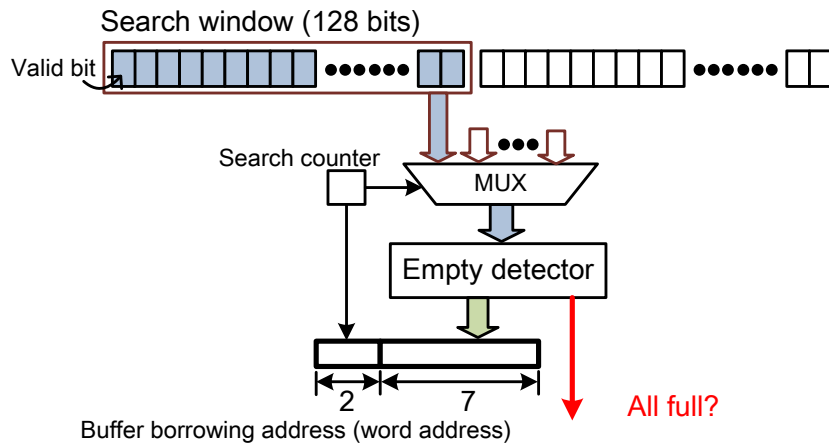


Fig.4. 8 Architecture of the empty memory block searching

After the NI send a borrowing request to the d-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 d-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.4. 8 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.4. 9 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.

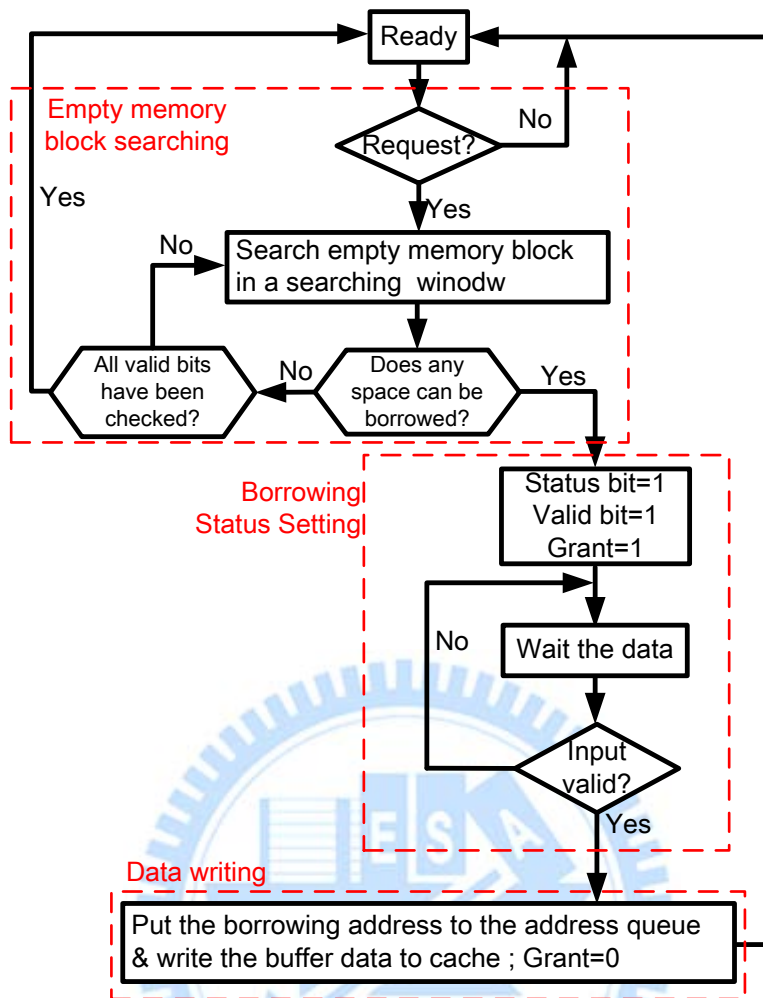


Fig.4. 9 Searching flow chart of the borrowing mechanism in d-MMU

4.1.1.2.2 Buffering Control

The buffering control in NI detects the empty size of the output queue and sends the borrowing operations to d-MMU. Fig.4. 10 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.4. 11. The borrowing mode indicates whether the blocking data stored in the d-MMU or not. Therefore, after receiving data from the PE, the data should be stored in the

d-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 d-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 d-MMU.

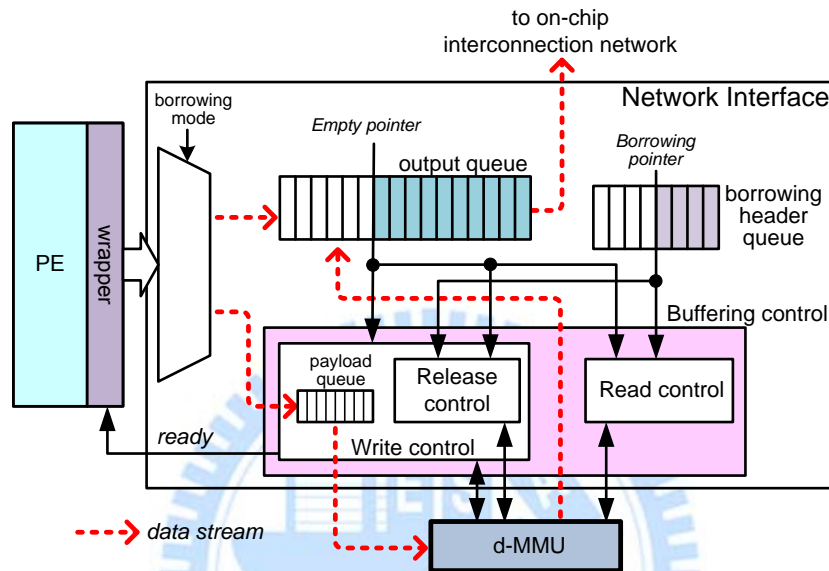


Fig.4. 10 Block diagrams of borrowing mechanism in network interface

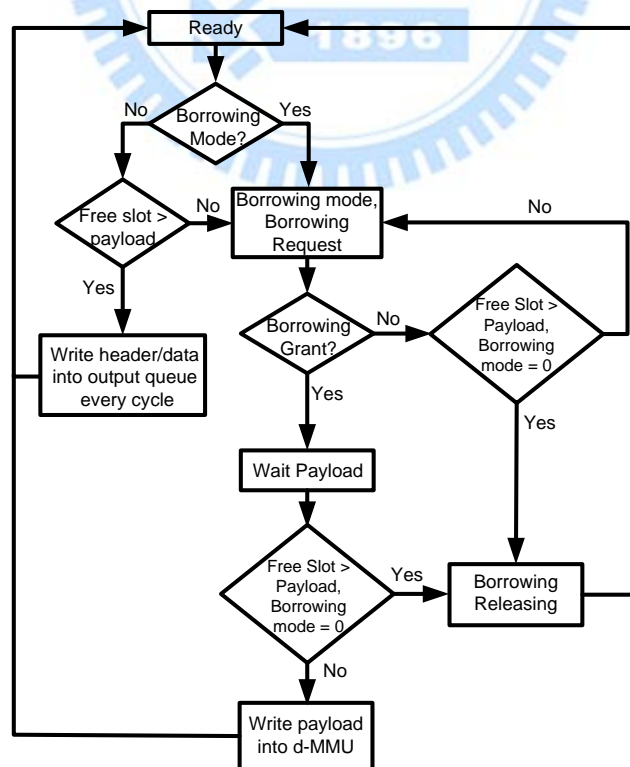


Fig.4. 11 Borrowing control policy of the buffering control

4.1.1.2.3 Simulation Results of Buffer Borrowing Mechanism

The proposed d-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.4. 12(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.4. 12(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.

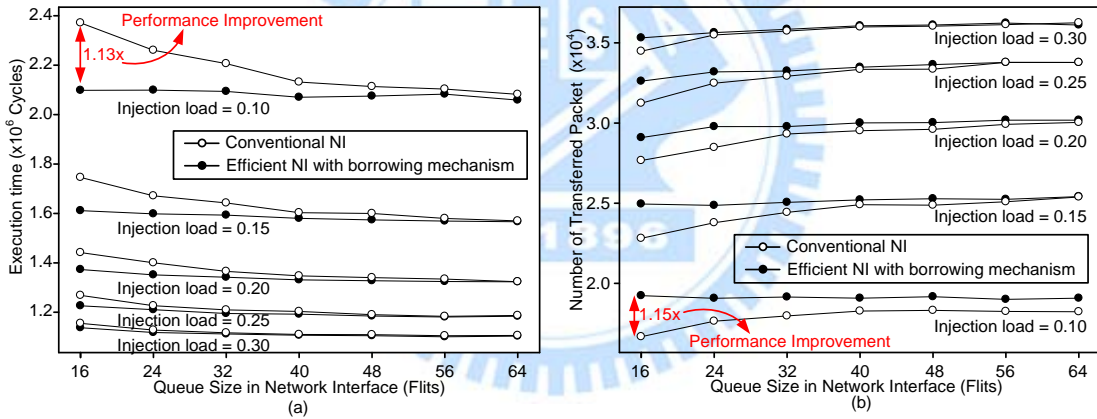


Fig.4. 12 (a) Execution time under various injection loads and queue sizes (b) Transferred packets under various injection loads and queue sizes.

4.2 Centralized Memory Management Unit Organization

The distributed memory resources may be insufficient for PEs. Lower level cache is applied to provide larger on-chip memory resources. Centralized cache and cache controller is included in centralized memory management unit(c-MMU). According to distinct memory resource requirements from different PEs, the proposed c-MMU can allocate different cache resources for each PE. In addition, the external memory is required for storing the huge data such as video frames in video processing. A DRAM controller is constructed in c-MMU to access DRAM device. The overall c-MMU

architecture with adaptive cache control and DRAM controller will be introduced in the following sections.

4.2.1 Design of c-MMU

The simple block diagram of the c-MMU is shown in Fig.4. 13. It is organized by an adaptive cache controller, switches, several SRAM sub-blocks and DRAM controller. Adaptive cache controller accepts the memory requests from d-MMUs. The requests issued by different d-MMU can simultaneously be executed if the used memory resources have no conflict. Cache controller will check the selected cache tables to determine whether the data is in the cache or not. According to the check result, the corresponding data and addresses are forwarded to the SRAM sub-block or DRAM controller by switch. For read requests, the read data forward to the output switch and send back to d-MMUs. In addition, the address translator and external memory interface are constructed to efficiently access the external memory.

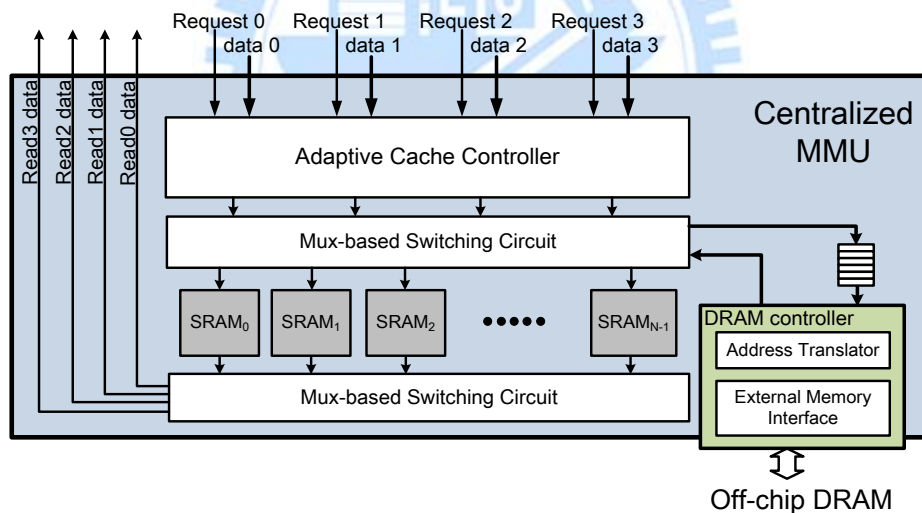


Fig.4. 13 c-MMU block diagram

Different applications may have different memory resource requirement. Even in the same application, it may have various memory behaviors at runtime. The proposed c-MMU can dynamically adjust and allocate suitable memory resources to each processor element. The concept of the adaptive memory resource allocation is shown in Fig.4. 14. Base on different memory requirement in different processor elements, unequal memory resources are allocated. Adaptive cache control scheme will be described in detail as follows.

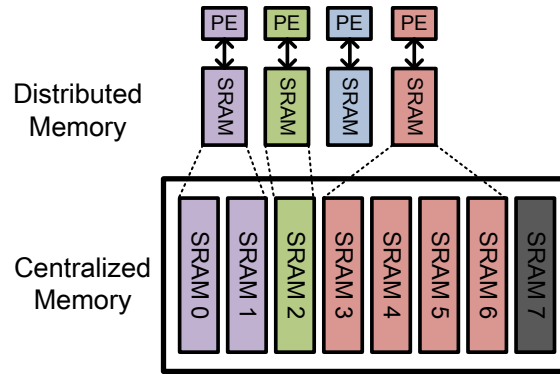


Fig.4. 14 Concept of the adaptive memory resource allocation

4.2.1.1 Adaptive cache control

In our work, the principle of adjusting the cache size is based on selective cache ways which had been proposed in [4.5]. With selecting different number of ways, the different cache size can be assigned for processor element. It is a simple method with less area and timing overhead for cache reconfiguration. In proposed c-MMU organization, associativity-based partitioning scheme is applied for the cache partition. Each SRAM sub-block represents a way and form a bank for the cache organization. Assume there are number of N SRAM sub-blocks in c-MMU, it represents there have N -way associativity capacity in centralized cache. For different processor elements, the SRAM blocks can be grouped into several groups for processor elements. Fig.4. 15 shows the example of SRAM bank partition. Assume the system have X processor elements and c-MMU has N SRAM banks. The memory partition can be achieved as illustrated in Fig.4. 15.

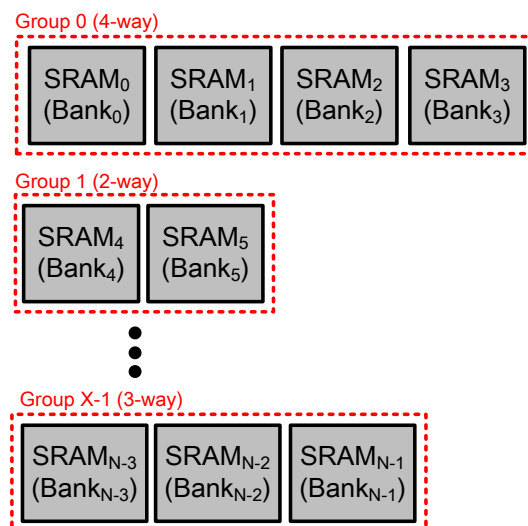


Fig.4. 15 Illustration of the memory partition

In order to dynamically allocate the memory resources for different processor elements at runtime, a Bank Assignment Table (BAT) is applied for recoding the memory usage information of three time intervals. Fig.4. 16 illustrates the cache table checking method when a request is served. According to the corresponding processor element node ID, the cache controller searches the BAT and returns the assigned bank numbers. These bank numbers indicate which bank tables need to be checked for the request. Fig.4. 16 shows the example that four banks are applied for node 3 in the first time interval. When a request from node 3 is served, Bank0, Bank1, Bank2 and Bank3 tables will be selected for hit checking. By this configuration, node 3 can own a 4-way associativity L2 cache memory resource for processing.

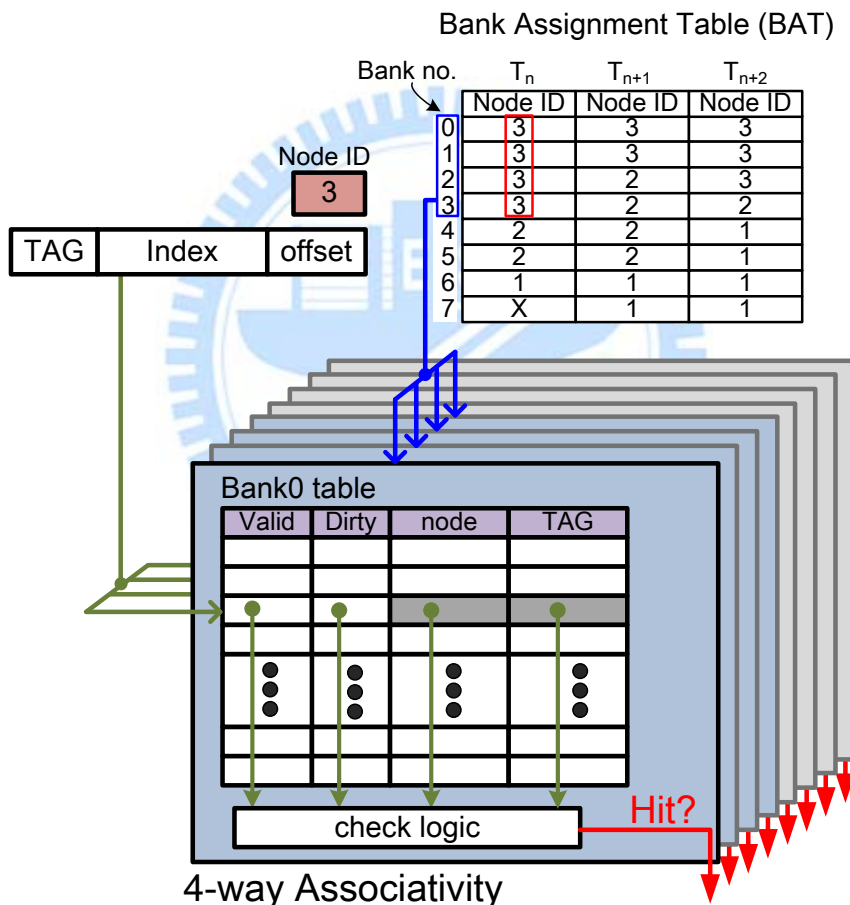


Fig.4. 16 Illustration of the cache table checking

For the multi-task system, multiple memory requests from different processor element can be served simultaneously in c-MMU because the checking tables are independent for different nodes generally. Fig.4. 17 shows the illustration of checking multiple requests. The target Bank tables are selected in accordance with BAT information, and the check functions are operated independently.

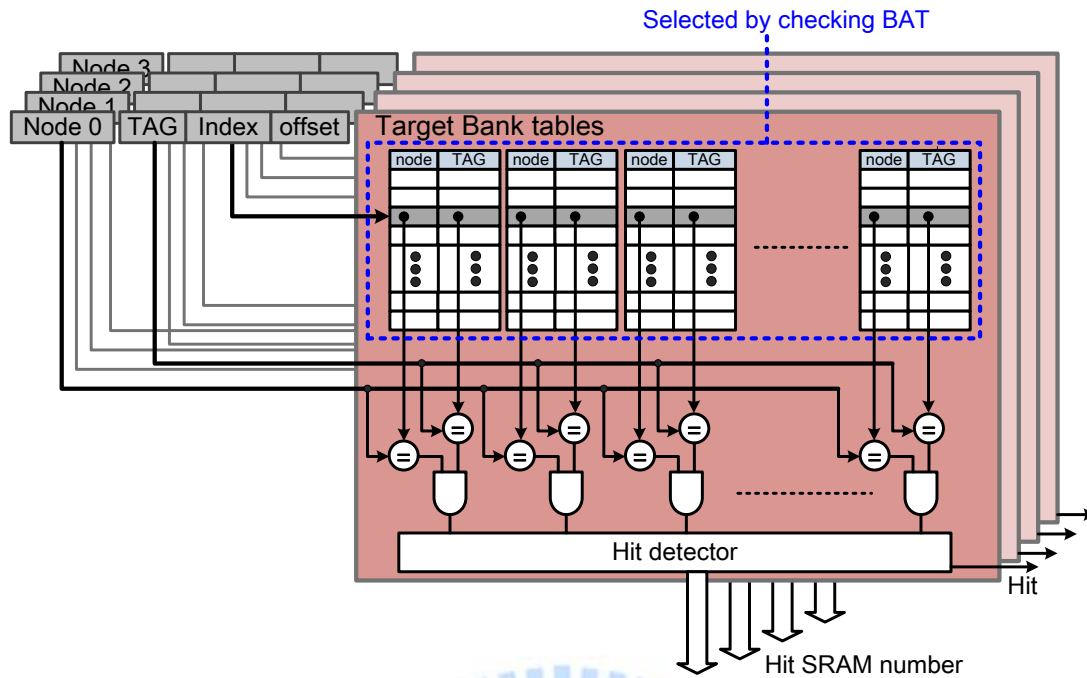


Fig.4. 17 Illustration of checking multiple requests

The processor elements may have different memory behaviors in different time interval at runtime. The BAT can recode the configuration in different time interval. It is updated by the processor element which can profile the memory requirements of the system. For the wireless video entertainment systems, 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 bitstream to satisfy the effective bandwidth. Based on various bitstream, 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.

With changing time intervals, the bank assignment for each processor element will be reorganized. The organization may be different from previous configurations. The Lazy-based transitioning scheme [4.6] is applied for maintaining data consistency. The basic concept of Lazy transitioning has been mentioned in the chapter 2. 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 c-MMU adaptive cache control is shown in Fig.4. 18. In Box 1, the read or write request will be chose 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. Follow that,

the corresponding tables are chose by BAT information (Box 2). If miss occur, the other tables would be checked because the corresponding data may still lie on the centralized memory. The situation will be happened when the memory resource for a particular processor element have been reduced. In Box 3, the BAT will be check again. If there have the other tables need to be checked, 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 (Box 5). The corresponding data will be moved from original location to the new location.

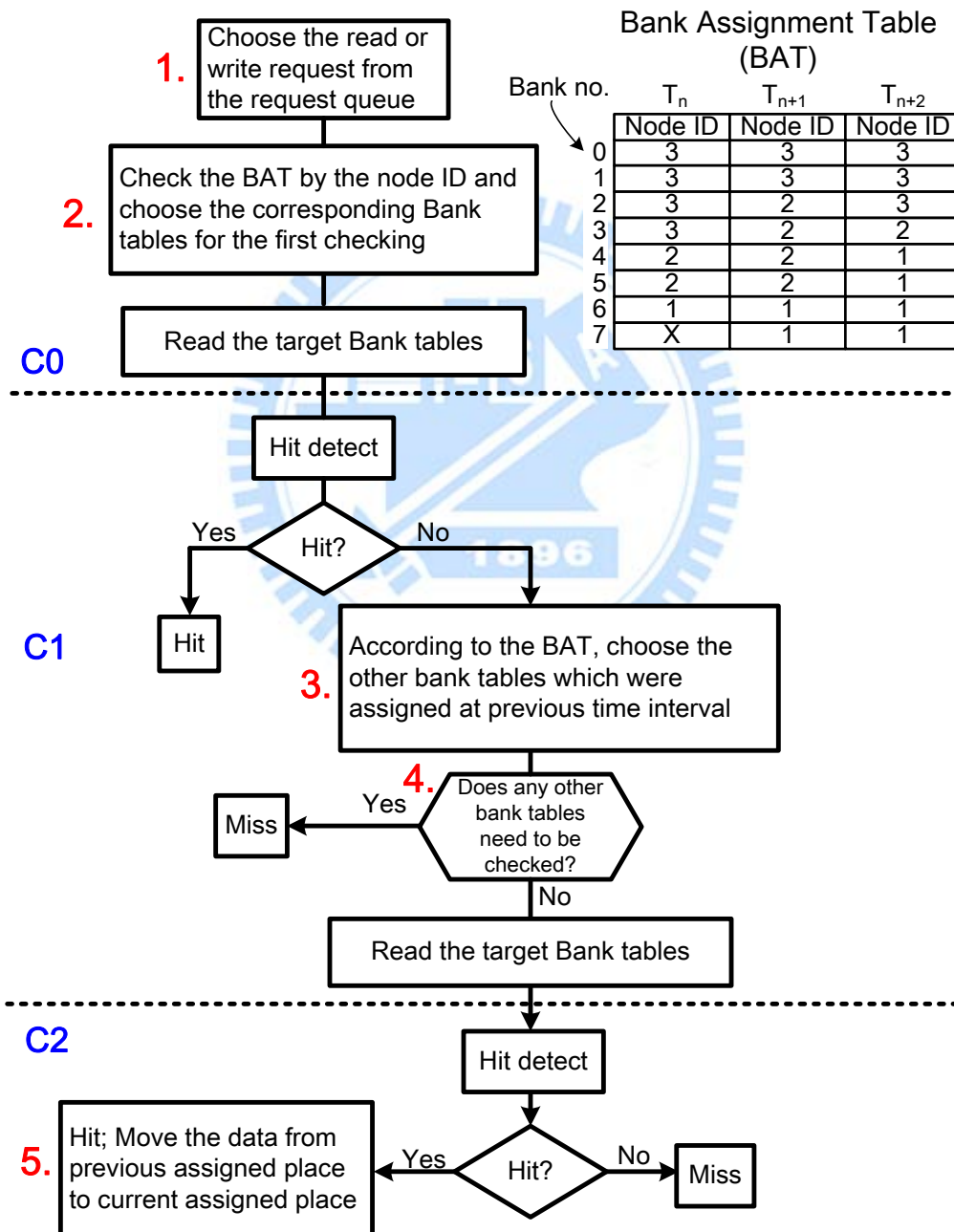


Fig.4. 18 Flow chart of adaptive cache control

4.2.1.2 Detail operation of the c-MMU

Having adaptive cache control, the overall architecture of c-MMU is constructed as shown in Fig.4. 19. The blue block in the figure is 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 bank assignment information according to the Node ID and selected time interval, and it determine what tables need to be check. At the following execution stage, the information in the bank tables has been read out, and then the hit detector outputs the corresponding read/write address and bank destination to the mux-based switch. If it wants to read data from external memory, the address will enter the corresponding bank pending buffer and external memory read queue. When the read data returns, the data can be directly write to the corresponding bank and transmit back to the corresponding d-MMUs by compare the address in the pending buffer(at third execution stage). The bank access input could come from cache controller, external memory and the other bank outputs generated by cache reorganization. An arbiter at the third execution stage is applied to determine the priority of these access requests for each memory banks. 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 to the lowest. At final execution stage, the cache data are read out and forward to the output, external memory or another SRAM banks by switch. For external memory request arbitration, read requests are served prior to the write requests.

In the DRAM controller, the address translator is applied to re-generate friendly 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 chapter 5. Furthermore, the external memory interface is constructed for accessing the DRAM data. The design of external memory interface will be intruded in the section 4.3.2.

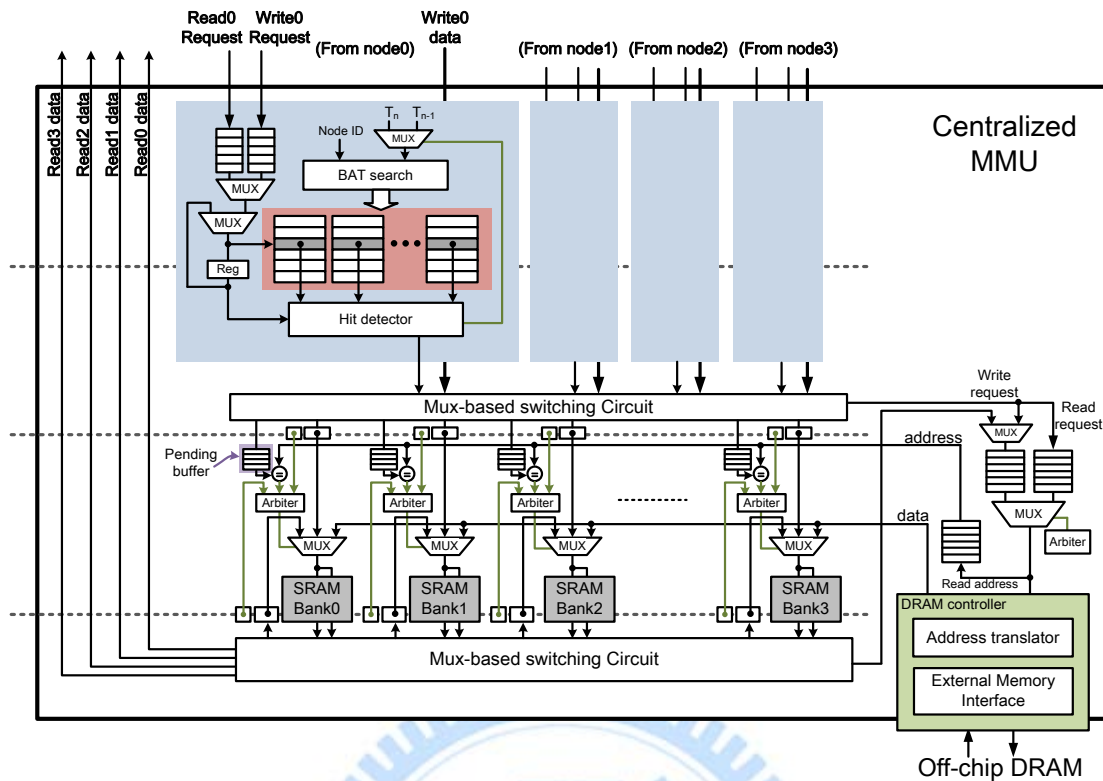


Fig.4. 19 Detail architecture of c-MMU

4.2.2 External Memory Interface in DRAM controller

In the design of memory hierarchy system, it usually needs an off-chip memory to be a hierarchy level for storage the large amount of data. The external memory interface is used to communicate with the external memory for the system. To deal with tremendous data transfer and storage in video processing, the external memory must provide high data bandwidth to achieve the real time request. The bandwidth of the external memory is limited due to the pin number of I/O is finite. Accordingly the external memory interface must provide high data bandwidth utilization by using some techniques. An external memory interface will be introduced in this section.

4.2.2.1 Concept of External Memory Interface & DRAM Model

The external memory interface (EMI) is an interface between on-chip system and off-chip DRAM devices. It will receive the physical addresses from the address translation machine, and generate DRAM commands to access DRAM data. EMI is designed to control the external memory. The simple connection of the EMI is show

in Fig.4. 20. EMI generates the appropriate commands defined in specification which have been introduced in chapter2. In addition, there are various and complex timing constrains for issuing the DRAM commands. EMI need to issue appropriate commands without any DRAM timing violation. In order to improve the bandwidth efficiency, a command scheduling would be applied to reschedule the DRAM commands. Because the banks in the DRAM can operate in parallel, the commands with different banks would enable issued without timing constrain. According to this concept, rescheduling DRAM commands enables higher bandwidth utilization than in-order issuing. The detail architecture of proposed EMI will be described in the next section.

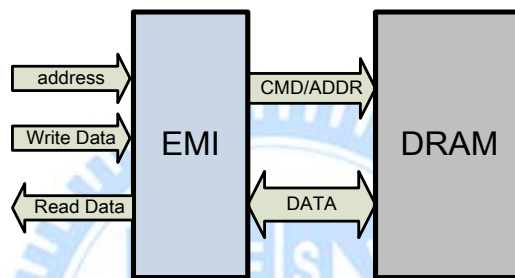


Fig.4. 20 Connection of EMI

In this work, 1Gb DDR3 SDRAM model provided by Micron Inc. [4.8] is used. Several speed grades and configurations can be chosen as shown in Table.4. 2. 15E speed grade and 64Megx16 configuration is chosen. There are 8 independent banks in a DDR3 device. The EMI would recode the bank status and generate appropriate commands according to the corresponding bank states. Different speed grades and configurations may have different timing constrain, the designer must follow these timing rules to build the memory interface. The detail timing issues will also be described in the following sections.

Speed Grade	Data Rate (MT/s)	Target ^t RCD- ^t RP-CL	^t RCD (ns)	^t RP (ns)	CL (ns)
-125 ^{1,2}	1600	11-11-11	13.75	13.75	13.75
-125E ^{1,2}	1600	10-10-10	12.5	12.5	12.5
-15 ³	1333	10-10-10	15	15	15
-15E ¹	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])

Table.4. 2 Micron`s DDR3 configurations

4.2.2.2 Architecture of EMI

The architecture of EMI is shown in Fig.4. 21. It consists of three finite state machines, FIFOs, command scheduler, Timing counters and I/O control circuit. The operation of proposed EMI can be briefly separated into three parts and each part is controlled by a finite state machine. In the following sections, these parts will be introduced.

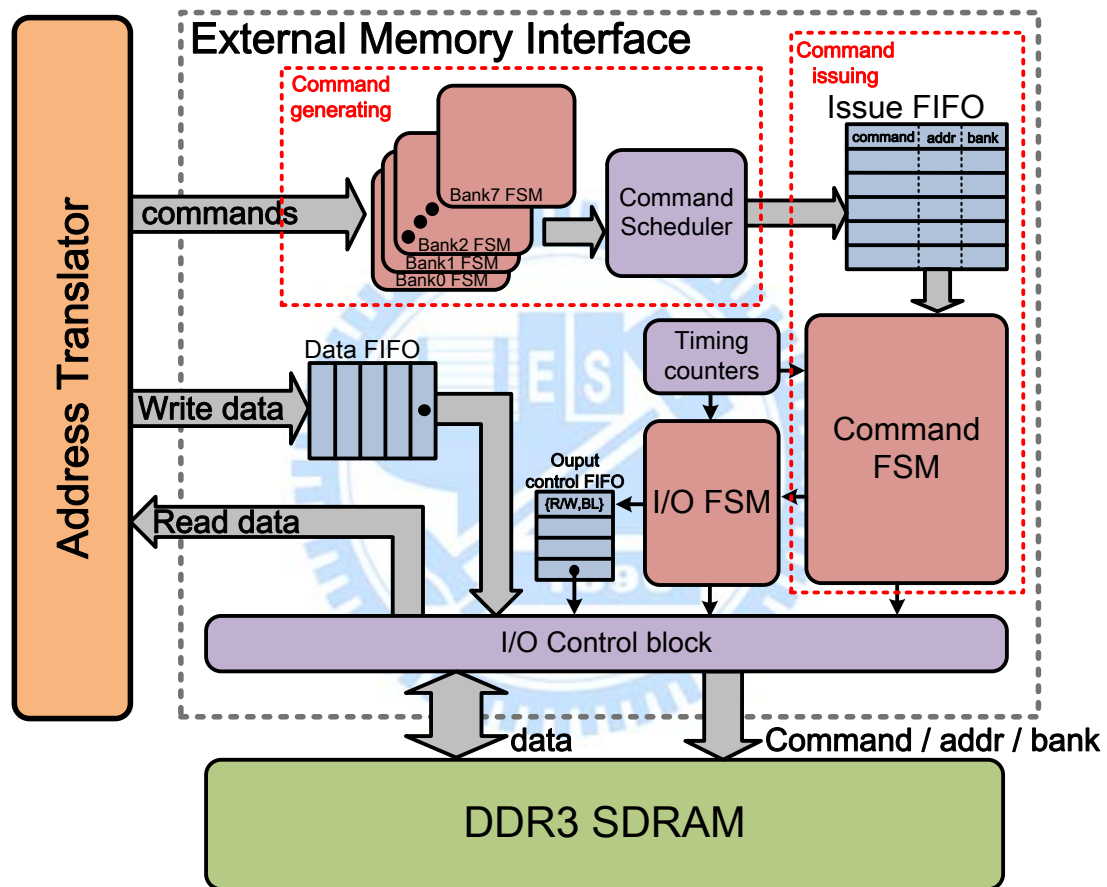


Fig.4. 21 Architecture of EMI

4.2.2.2.1 Operation of EMI

The first one is command generating part. In order to generate reasonable commands to access DRAM, 8 Bank Finite State Machines (FSM) are constructed to recode the status of eight DRAM internal banks. The state diagram of Bank FSM is shown in Fig.4. 22(a). When an input command addresses to one of DRAM banks, the state of the corresponding Bank FSM would be checked. According to different bank

status, correct commands are issued to Command Scheduler for rescheduling.

The second one is command issuing part. After rescheduling the command, the DRAM commands are stored in 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. It is controlled by several timing counters which recode the cycle margins of different timing constraints. When a command is issued, the relative timing counters will be set to a certain value and start to decrease until the counter is return to zero. The timing counters will be checked when issuing new commands from issue FIFO. If there is no timing violation, the command can be issued to DRAM. Otherwise, additional stalls will occur. During the time of waiting, EMI will issue NOP commands to external memory. The common timing parameters are shown in Table.4. 3. In addition, The DRAM needs a long latency to power up and initialization including ZQ calibration and mode register loading. Fig.4. 22(b) shows command FSM state diagram. It includes initialization states, issue states and several waiting states. Initialization states handle the DRAM initializations. Issue states generate the appropriate DRAM commands to I/O control block. Additional waiting states would stall the command issuing until the following command can be issued legally.

The third part is I/O control. When a write command is issued, the write data must be sent after column write latency. Also, the read data would appear in the data bus after column read latency when a read command is issued. The I/O control block controls the timing of access data, and it is controlled by I/O FSM. Furthermore, the Data Strobe (DQS) signal would need to be controlled by I/O for DRAM access data aligning. Fig.4. 22(c) shows the state diagram of I/O FSM.

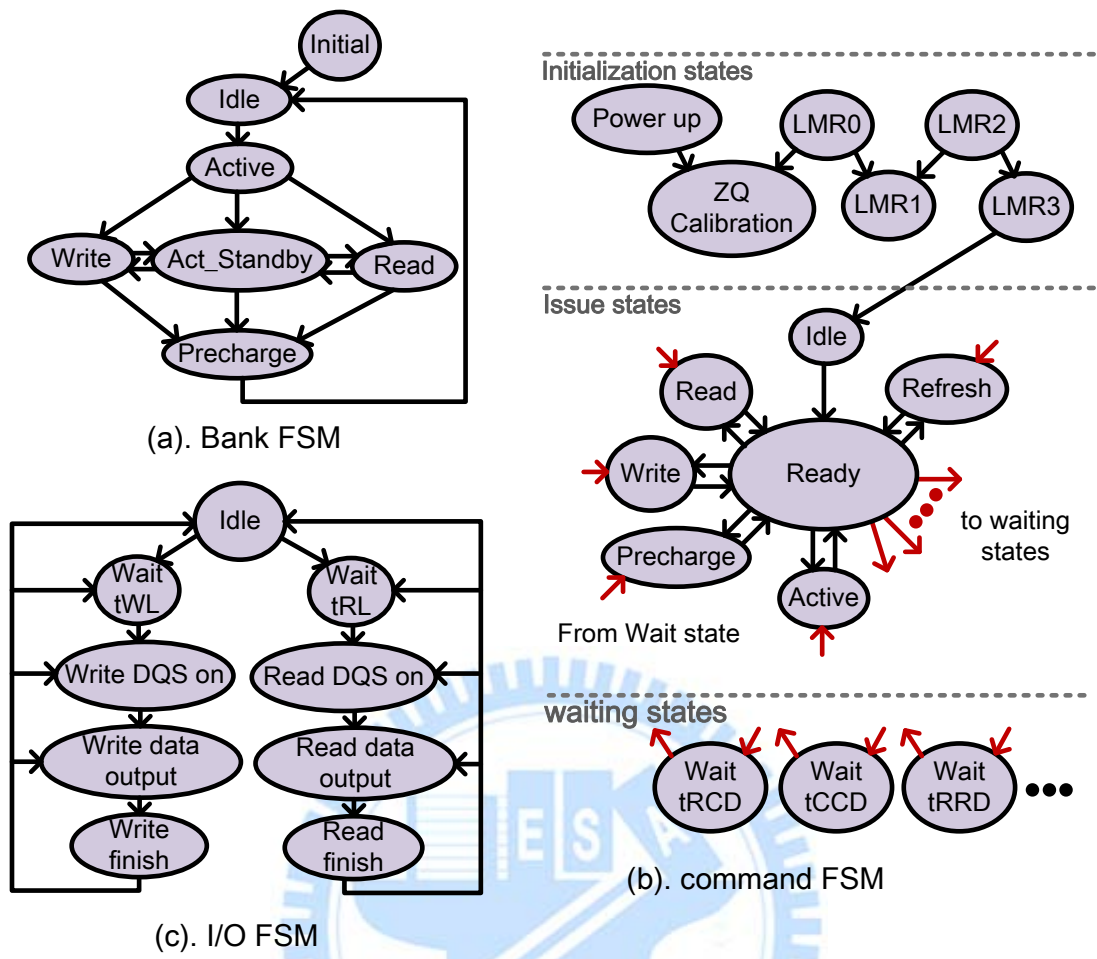


Fig.4. 22 State diagram of EMI Finite State Machines

Parameter	Symbol	Micron DDR3 (-15E x16) (MIN value)	Unit
Minimum Clock Cycle Time	tCK	1.5	ns
Active to Read or write delay	tRCD	13.5	ns
Pre-charge period	tRP	13.5	ns
Write recovery time	tWR	15	ns
Write to Read delay	tWTR	7.5	ns
Load Mode Register time	tMRD	4	clk
Active to pre-charge	tRAS	36	ns
Active a to Active b	tRRD	7.5	ns
Active to Active	tRC	49.5	ns
Four Bank Active window	tFAW	40	ns
Cas to Cas command	tCCD	4	clk
Minimum CAS Latency	tCL	13.5	ns

Table.4. 3 Common timing parameters of Micron DDR3 SDRAM

4.2.2.2.2 Command Scheduler

In order to improve the bandwidth efficiency, the Command Scheduler is applied to reschedule the command sequence. To fully utilize the DRAM bandwidth, it is necessary to parallelize the accessing which address to different banks. With different situations, appropriate scheduling will be applied.

When the successive accesses address to different banks, the bank-miss will occur. Fig.4. 23(a) shows the original command sequence without any scheduling, and it has the worst bandwidth efficiency. Fortunately, the banks in a DRAM device can operate in parallel, so we can activate the banks first and then issue the column access commands as shown in Fig.4. 23(b). The bank activate time can be hidden. However, no more than four bank ACTIVATE commands may be issued in a given tFAW (MIN) period. If the number of successive accesses with different banks exceeds four, the optimal sequence is that interleaving ACTIVATE and column access commands as shown in Fig.4. 23(c). The proposed Command Scheduler can schedule the ACTIVATE and column access command with different banks to the optimal sequence. Note that the calculation of cycles in Fig.4. 23 is base on the minimum clock cycle time defined in Table.4. 3.

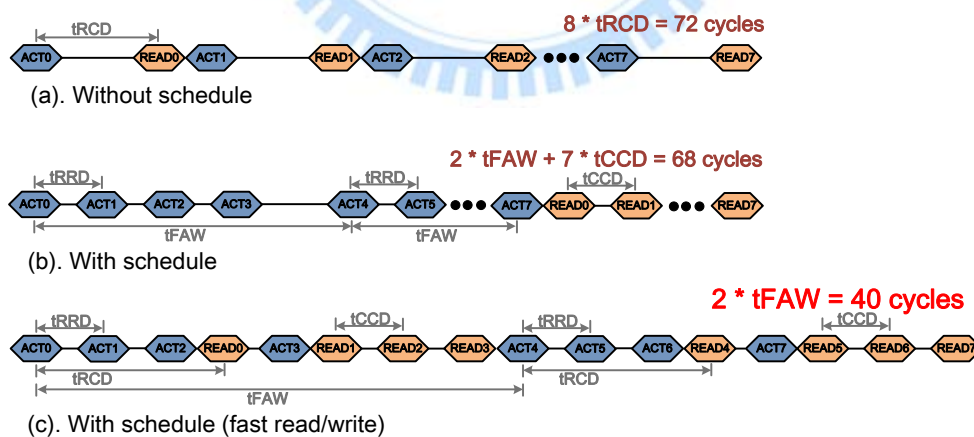


Fig.4. 23 bank-miss scheduling

For accessing DDR3 devices, data for any write burst may be followed by a subsequent read command after tWTR has been met. It may cause worse bandwidth efficiency when the read and write commands are interleaved frequently. Fig.4. 24(a) illustrates the example of issuing the read bursts after write bursts. If the successive

read and write commands have no data dependency, the issue sequence can be exchanged so that the bandwidth efficiency can be improved. The example with scheduling is shown in Fig.4. 24(b).

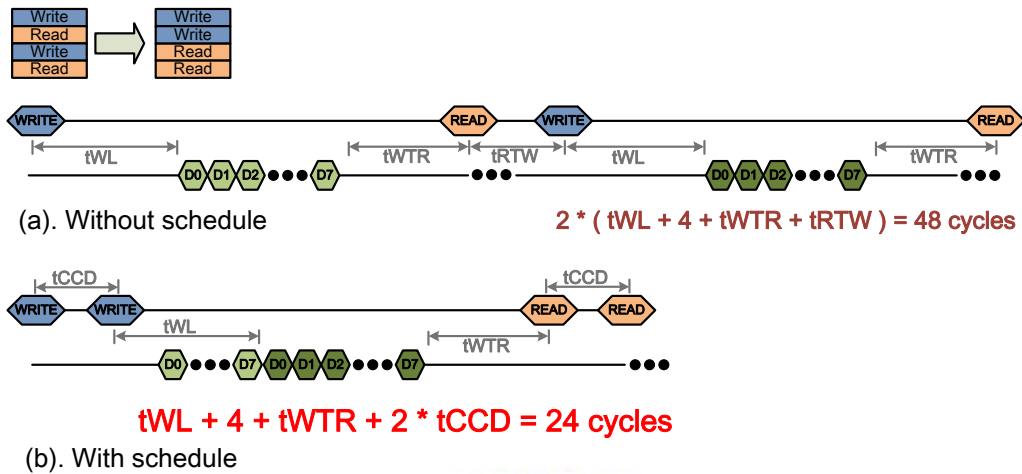


Fig.4. 24 read / write scheduling

When row-conflict occurs, the PRECHARGE and ACTIVATE commands must be issued to deactivate the open row and re-activate new row. Fig.4. 25 shows the example of four successive row-conflict reads with different banks. With scheduling, the PRECHARGE and ACTIVATE commands can be issued in advance so that the precharge and activate time can be hidden.

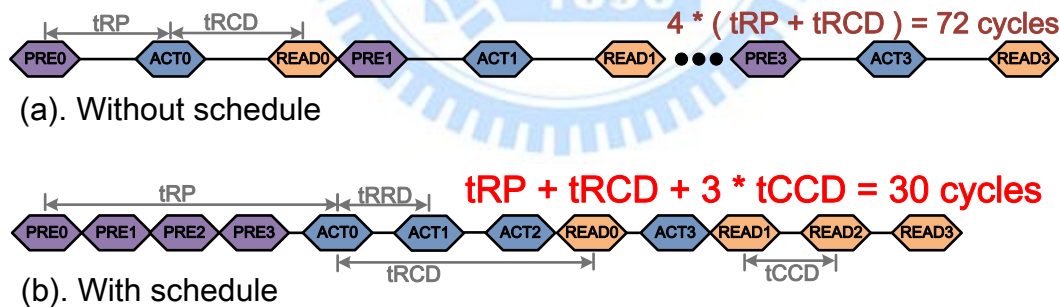


Fig.4. 25 row-conflict scheduling

4.2.2.3 Bandwidth Improvement with Command Scheduler

Command Scheduler can improve the memory bandwidth efficiency by rescheduling the DRAM commands. Especially for the irregular DRAM accesses, 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, we define the bandwidth utilization as shown in

the following equation to calculate the DRAM bandwidth utilization.

$$\text{DRAM Bandwidth Utilization} = \frac{\text{Total cycles of outputting and inputting data between DRAM}}{\text{Total cycles of processing access commands}} \times 100\%$$

For simulation, the random access pattern is applied for measuring the DRAM bandwidth utilization in the worst case. The successive access requests are random, so the DRAM row and bank conflict would happen frequently. For the other simulation information, the summary of simulation configurations is shown in Table.4. 4.

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

Table.4. 4 Simulation summary

Base on the defined DRAM Bandwidth Utilization equation, the bandwidth utilization can be estimated, and the simulation result is shown in Table.4. 5. With Command Scheduler, it can improve 42.8% bandwidth utilization.

	Without scheduler	With Scheduler	Improvement
Bandwidth Utilization	18.58%	26.53%	42.8%

Table.4. 5 Simulation of the bandwidth utilization

4.2.3 Simulation Results of the Adaptive Cache

In this section, the simulation results of the adaptive cache will be introduced. In the beginning of this section, the access latency and energy estimation method of memories will be introduced. Base on the measurement method, the simulations for static bank assignment and dynamic bank assignment will be described in section 4.2.3.2.

4.2.3.1 Latency & Energy Estimation method

For measuring the execution latency and energy consumptions including on-chip cache and off-chip DRAM, the estimation methods will be introduced in the following sub-sections.

4.2.3.1.1 Cache Latency Estimation

For verification and simulation, a cycle-driven model is development by SystemC. With constructed systemC models of the memory management units, the cache access latency can be considered in simulation.

4.2.3.2.2 Cache Energy Estimation

To approximately estimate the cache energy consumptions in system level, CACTI 5.3 model [4.7], which is provide by HP Labs, is applied to characterize the energy consumption of memory elements in d-MMU and c-MMU. CACTI is an powerful model that enable 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 so that total energy consumption can be calculated.

4.2.3.2.3 DRAM Latency Estimation

The memory access latency can be estimated according to the Centralized MMU block size and selected DRAM configuration. When a cache miss occur, Centralized MMU will send the memory requests to DRAM and read the required block data and write back the replaced block data. Assume the block size is 64-byte in the Centralized MMU configuration, so four DRAM access commands with eight burst length will be generated for a block access. Fig.4. 26 shows the DRAM read latency for a block data in different situations. When the reference bank is in row closed state,

the activate command need to be issued for opening the particular row. After t_{RCD} cycles, the read command can be issued for reading data and the read data will successively be read out after t_{CL} cycles. The read burst length is set to 8 so the total data outputting time are 16 cycles. Note that the data in a block are generally located into the same row so the row-conflict status would not occur for a block data accessing. The timing diagram and cycle estimation is shown in Fig.4. 26(a). When the present row address is equal to the previous activated row in the reference bank, the activated command can be reduced and the relative cycle estimation is shown in Fig.4. 26(b). Adversely, the row-conflict would occur when the present row address is different to the previous activated row in the reference bank. The pre-charge and activate commands must be issued to change the row. Additional t_{RP} cycles would be added in cycle estimation. The timing diagram with row-conflict is shown in Fig.4. 26(c).

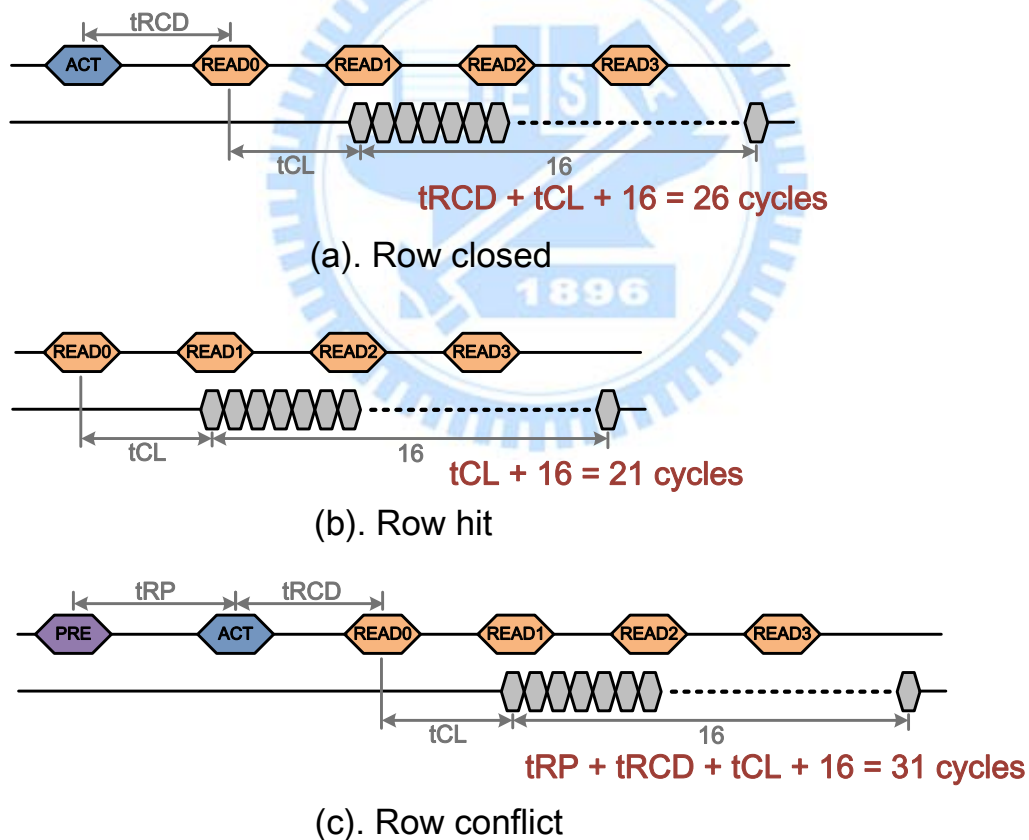


Fig.4. 26 DRAM latency estimation for different situations

4.2.2.2.4 DRAM Energy Estimation

In order to measure the DRAM energy consumption, the system power calculators

are provided by Micron Technology Inc. [4.9]. These models can estimate the power requirement of SDRAM devices in a system environment. These tools provide a friendly interface for estimating the memory power requirements needed in making important system architecture and design decisions. With an accurate estimation of power consumption, the system designer can quickly handle complex system trade-offs to optimize the system performance [4.9].

According to the selected DRAM and system configurations, the DRAM power consumption can be automatically calculated. The configuration summary in our simulation environment is shown in Table.4. 6. The example of the DDR3 configuration interface in the System Power Calculator is shown in Fig.4. 27 and the system configuration interface is shown in Fig.4. 28. To simplify the simulation, only one DRAM device is applied for measuring the power consumption produced by SVC memory accesses, so the number of rank is set to 1. For setting system configurations, the percentage of time that all banks are in a pre-charged state can be set to zero because the used DRAM page policy is open page policy. In addition, the DRAM page hit rate and the percentage cycles of access data between DRAM, which are marked in Fig.4. 28, would need to be measured for power calculation.

After setting these configurations, the DRAM power would be generated. Fig.4. 29 shows a summary of the power measurement result including background power, activate power and read/write/termination power. According to these results, the DRAM power in the system can be measured accurately. The detail documentation and tools of the System Power Calculator can be downloaded in website [4.9].

DRAM configuration	
DRAM Model	Micron 1Gb DDR3 SDRAM (MT41J64M16)
Configuration	64Meg x 16
Speed Grade	-15E
System configuration	
VDD	1.5V
Clock frequency	333MHz
Burst length	Fixed to 8
Number of Rank	1
DRAM Page Policy	Open page policy (The percentage of time that all banks on the DRAM are in a precharged state is set to 0)

Table.4. 6 Summary of system and DRAM Configuration



DDR3 SDRAM Configuration

DRAM Density	1Gb	▼
Number of DQs per DRAM	x16	▼
Speed Grade	-15E	▼
Mode Register bit 12: Precharge PD Exit Mode	1.Past	▼

Inputs for DRAM selection are chosen here.

The table below is automatically updated from Device Spec Worksheet

Parameter	Condition	Value	Units
	Maximum Vcc	1.575	V
	Minimum Vcc	1.425	V
	Number of DQ strobes (DQS) per DRAM	4	
	Number of data mask (DM) per DRAM	2	
IDD0	Maximum active precharge current	110	mA
IDD2P	Maximum precharge power-down standby current	35	mA
IDD2N	Maximum precharge standby current	65	mA
IDD3P	Maximum active power-down standby current	40	mA
IDD3N	Maximum active standby current	60	mA
IDD4R	Maximum read burst current	300	mA
IDD4W	Maximum write burst current	355	mA
IDD5A	Maximum burst refresh current	240	mA
	^t CK used for current measurements (see current notes)	1.5	ns
^t RRD	Minimum activate-to-activate timing (different bank)	7.5	ns
^t RC	Minimum activate-to-activate timing (same bank)	49.5	ns
^t RAS	^t RAS used for IDD0 calculation	36	ns
^t RFC (MIN)	Minimum refresh-to-refresh cycle time	110	ns
^t REFI	Average periodic refresh cycle time	7.8	µs
^t CK (MIN)	Minimum ^t CK cycle rate	1.5	ns
^t CK (MAX)	Maximum ^t CK cycle rate	3.3	ns

Values are extracted from "DDR3 Spec" tab. Please confirm the numbers in that worksheet are updated prior to use.

Fig.4. 27 DDR3 Configuration interface of the System Power Calculator



DRAM Usage Conditions in the System Environment

	System Vcc	1.5	V	
	System CK frequency	333	MHz	
	Burst length	8		must be either 4 or 8
PdqRD	DDR3 SDRAM output power per individual DQ on this DRAM	4.9	mW	This value is the output driver power per DQ on the DRAM. It is specific to each system design and must be calculated based on the termination
PdqWR	DDR3 SDRAM termination power per individual DQ during WRITES to this DRAM	9.3	mW	This value is the output driver power per DQ on the DRAM. It is specific to each system design and must be calculated based on the termination
PdqRDoth	DDR3 SDRAM termination power per individual DQ during READS from other DRAM	0	mW	This value is the output driver power per DQ on the DRAM. It is specific to each system design and must be calculated based on the termination
PdqWRoth	DDR3 SDRAM termination power per individual DQ during WRITES to other DRAM	0	mW	This value is the output driver power per DQ on the DRAM. It is specific to each system design and must be calculated based on the termination
BNK_PRE%	The percentage of time that all banks on the DRAM are in a precharged state	0%		
CKE_LO_PRE%	The percentage of the all bank precharge time for which CKE is held LOW	0%		
CKE_LO_ACT%	The percentage of the at least one bank active time for which CKE is held LOW	0%		
PH%	Page hit rate	90%		
RDsch%	The percentage of clock cycles which are outputting read data from the DRAM	19%		
WRsch%	The percentage of clock cycles which are inouting write data to the DRAM	34%		
termRDsch%	The percentage of clock cycles which are terminating read data to another DRAM	0%		must be 0% for a 1-rank system
termWRsch%	The percentage of clock cycles which are terminating write data to another DRAM	0%		must be 0% for a 1-rank system
^t RRDsch	The average time between ACT commands to this DRAM (includes ACT to same or different banks in the same DRAM device)	239.3	ns	

Values that may be updated are shown in green. This will be dependent on how the system accesses the DRAM.

Fig.4. 28 System configuration interface of the System Power Calculator

1Gb DDR3 SDRAM with 16 DQs and a -15E Speed Grade

System is operating at 333 MHz clock with VDD = 1.5V. Read bandwidth is 253.4 MT/s with write bandwidth of 449.7 MT/s. The DRAM terminates 0 MT/s. ACT commands are separated by 239.3ns on average. All parameters are calculated and require no user input.

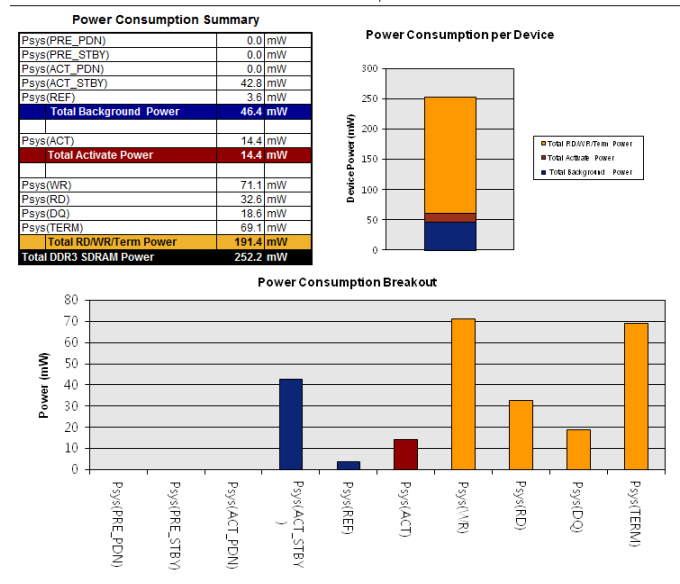


Fig.4. 29 Summary of the power measurement result in the System Power Calculator

4.2.3.2 Simulation of Adaptive Cache

For various memory demands required by different PEs in memory-centric on-chip data communication platform, c-MMU can support reconfigurable bank assignment for PEs. In order to simulate the behavior of the stream applications in a heterogeneous system, task-level pipeline organization is applied for the simulation as illustrated in Fig.4. 30. Assume the stream application can be separated into four tasks and mapped to four nodes in platform. Each node forms a pipeline stage for application. According to different memory behavior in nodes, c-MMU allocates different number of SRAM banks for different nodes.

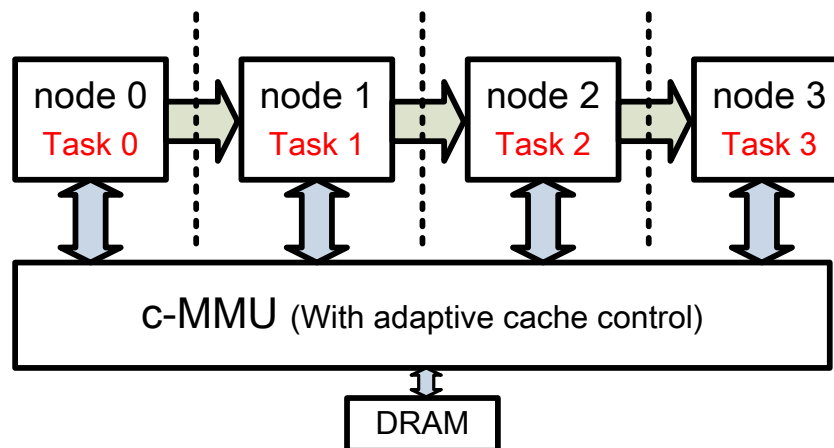


Fig.4. 30 Organization of simulation

For adaptive cache simulation, the tasks with random memory access are applied in each node. A task is composed of 100 number of random memory accesses with a particular range. For pipeline behavior, the task in pipe N can be lunched when the task in pipe N-1 is done. In the simulation, it is assumed that the memory requirement of each node in different intervals of time can be profiled by system. By updating BAT in proposed c-MMU with profiled information, adaptive memory resources partition can be achieved. Suitable adaptive bank assignment can improve the execution time and energy consumption compare to the fixed bank assignment (every node owns equal number of banks statically).

Table.4. 7 lists the simulation information of the memory configurations. Here a pattern with memory requirement assumption is simulated in my simulation. Table.4. 8 lists the information of this pattern. By the assumption of memory requirements, BAT in c-MMU can be updated by system for bank assignment. Additionally, assuming the memory requirements would be different at runtime, the assumption for three intervals of time are listed in Table.4. 8. For simulation, 500 tasks would be finished in a time interval. With profiling the memory requirement and re-allocating the bank assignment, the memory resources in c-MMU can be utilized effectively. When finishing 1500 tasks (three time intervals), 40.41% execution cycles and 48.54% memory energy reductions can be achieved compared to the fixed bank assignment method.

L1 cache (d-MMU) configuration	
Cache Size	4KB
Number of banks	2
Associativity	4-way
Block size	32-byte
Replacement policy	LRU
Write policy	Write back
L2 cache (c-MMU) configuration	
Cache Size	512KB
Number of banks	16
Associativity	N-way, $1 \leq N \leq 16$ (depend on bank assignment)
Block size	64-byte
Replacement policy	LRU
Write policy	Write back
External Memory configuration	

Device	DDR3 SDRAM
Channel/Rank/Bank	1/1/8
Size	128MB
Number of banks	8
Burst length	Fixed to 8
DRAM Page Policy	Open page policy

Table.4. 7 List of simulation information

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

Table.4. 8 Memory requirement assumption and corresponding bank assignment for c-MMU

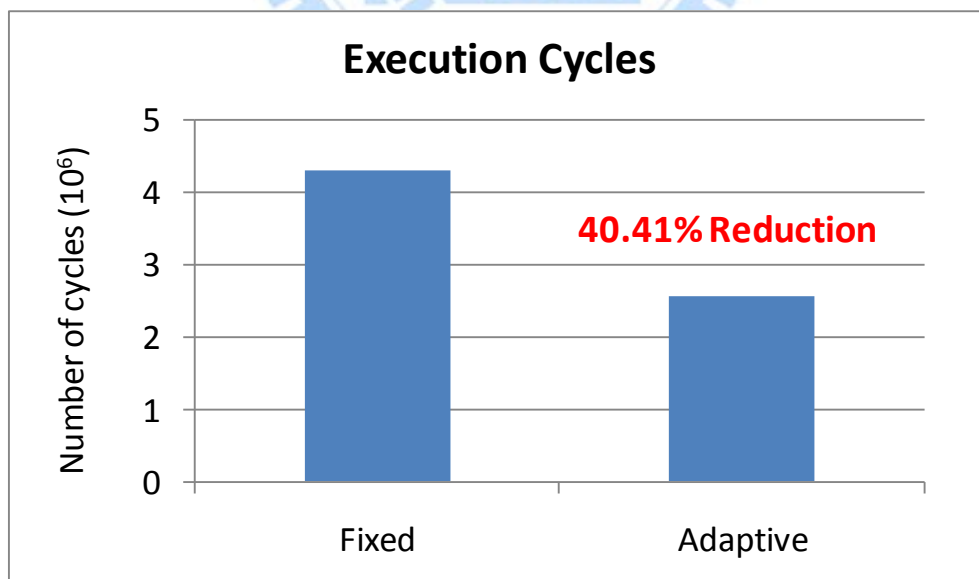


Fig.4. 31 Total execution cycles

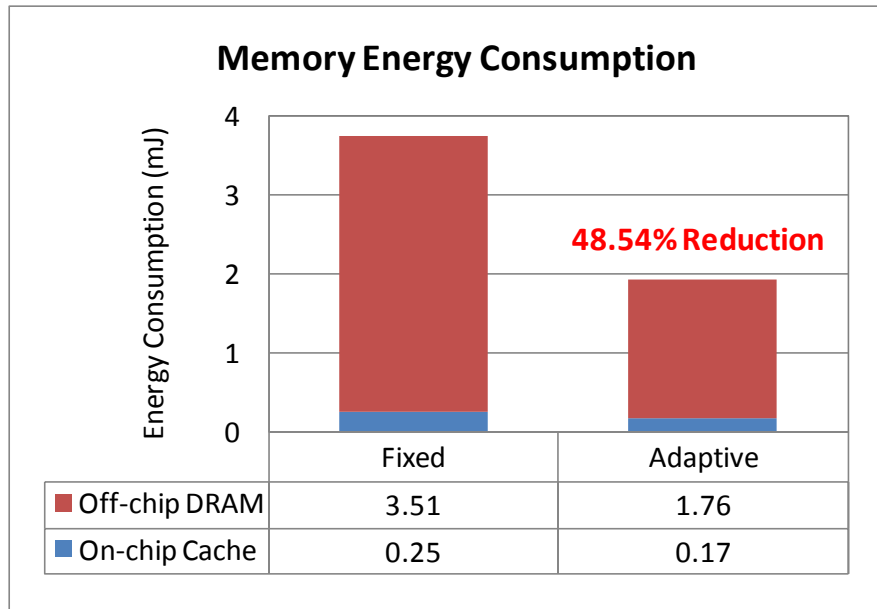


Fig.4. 32 Total memory energy consumption

4.3 Summary

For on-demand memory system presented in chapter 3, it needs efficient memory management units to process the memory access and control the memory resource allocation. In this chapter, the design of distributed memory management unit (d-MMU) and centralized memory management unit (c-MMU) have been described.

In on-demand memory system, d-MMU performs as a high level cache for the dedicated PE. In order to access continuous data easily, PE uses the pre-defined burst-based memory access protocol to access memory. With burst transmission mechanism, the cache miss penalty can be hidden. In addition, a novel buffer borrowing mechanism is proposed to reduce the stall caused by small packet buffer size in network interface. If the utilization of distributed cache is low, the d-MMU can borrow the cache blocks for buffering the blocking packets from PEs, and the PEs can keep processing for their tasks.

The distributed memory resources may be insufficient for PEs. Centralized memory management unit(c-MMU) is designed for managing and providing larger centralized memory resources for system. However, PEs may have different memory requirements at runtime. Proposed c-MMU can support cache resource re-allocation by updating the bank assignment in c-MMU. Base on the associativity-based partitioning scheme, centralized SRAMs can be separated into several groups which

own different number of SRAM banks, and these groups are assigned to different PE to be a lower level cache. Additionally, an external memory interface (EMI) in DRAM controller is constructed to access external memory efficiently. By re-scheduling DRAM commands, the effective bandwidth of DRAM can be improved.



Chapter 5

On-Demand Memory System for Wireless Video Entertainment Systems

In this chapter, on-demand memory system is applied for wireless video entertainment systems. A pre-fetch and address translation mechanism will be proposed to improve the performance for Scalable Video Coding (SVC) processor element in this chapter. SVC processor element is the largest memory user in the system because there have multiple quality and spatial layers of the video frames need be stored and accessed. Thus, a pre-fetch approach for SVC is proposed to improve the memory access performance. 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, a suitable DRAM data arrangement for SVC data is applied to reduce the DRAM access latency and power consumption. The address translation machine can increase the probability of row-hit and bank-hit status in the memory controller so that the additional activated power and the pre-charge time can be saved. The pre-fetch mechanism is presented in section 5.1 and the address translation is presented in section 5.2. In addition, SVC has different memory requirements for decoding different layers. With Proposed adaptive cache control in c-MMU as mentioned in chapter 4, optimizing on-chip memory utilization can be achieved for SVC. The analysis and simulation results will be described in section 5.3.

5.1 Data Pre-fetch for SVC

5.1.1 Introduction

In realizing the video coding hardware system, the most critical issue not only focuses on the hardware costs but the power consumptions. However, the property of intensive memory data accessing, especially for external memory access, contributes significant part of entire video coding system power consumptions. Therefore, if the

external memory accesses can be reduced, both of the power consumption and system performance can be improved. To lighten the overhead of external memory access, several works [5.1]-[5.4] discussed how to reduce the search range for motion estimation in part of encoder. For the video decoder, the literatures [5.5]-[5.8] explored the possibility of data reuse for motion compensation. Extended from the concept of memory organization, works [5.9]-[5.11] adopted the concept of pre-fetch to early acquire the data which will be used for reconstructing the video signal and thus increase the system performance and take the advantage of regular data accessing. In the traditional single layer video codec structure, such as H.264, the mechanism of pre-fetch is usually applied on motion estimation and compensation. However, due to the irregular moving of video content, the performance of pre-fetch is usually inefficient. It is mainly come from the irregular object moving in video content, said that the video decoder has no idea about where the object in video content will move to before decoding the motion vectors and failed to pre-fetch suitable data. Therefore, if the mechanism of pre-fetch can be applied on the data which will be surely used in the following operation, the hit rate can be significant improved and thus increase the system performance.

5.1.2 Inter-layer prediction of the SVC

In the latest video coding standard called scalable video coding (SVC), three scalabilities are supported to achieve spatial, temporal and quality adaptation for satisfying the application diversities. In addition, to further increase the coding performance, SVC additional adopts several prediction modes called inter-layer predictions to join the competition of mode decision with other traditionally available prediction modes. In the inter-layer prediction, the motion information, residuals and reconstructed video signals in base layer will be used as the references to predict the current macroblock. The three inter-layer prediction modes adopted in SVC are described as follows.

A. Inter-layer motion prediction

In this 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.5. 1. In this manner, the

macroblock partition of the enhancement layer is acquired from 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.

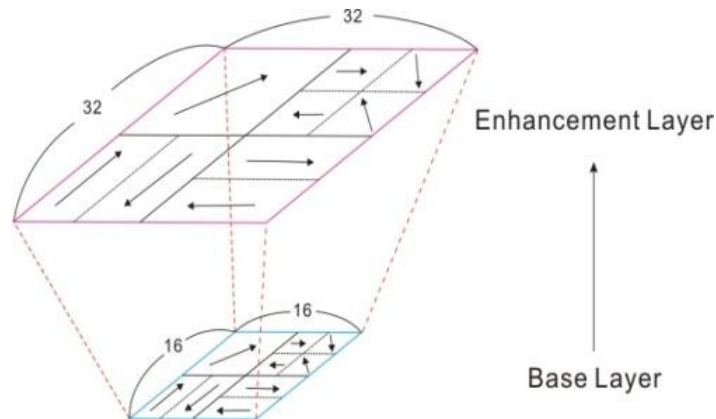


Fig.5. 1 Illustration of inter-layer motion prediction [5.12]

B. Inter-layer residual prediction

Fig.5. 2 shows the concept of inter-layer residual prediction mode. When inter-layer residual prediction is performed, the residual data is up-sampled from 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.

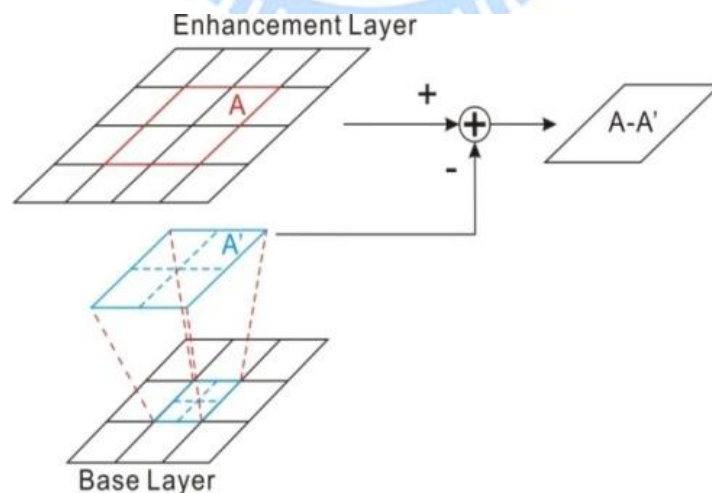


Fig.5. 2 Illustration of inter-layer residual prediction [5.12]

C. Inter-layer intra prediction

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. That is, the enhancement layer macroblock can be predicted by up-sampling the reconstructed macroblock of the base layer and the concept of inter-layer intra prediction is shown in Fig.5. 3. For up-sampling the reconstructed macroblock in base layer, one-dimensional four-tape and bilinear filter are used for up-sampling the luminance and chrominance components, respectively.

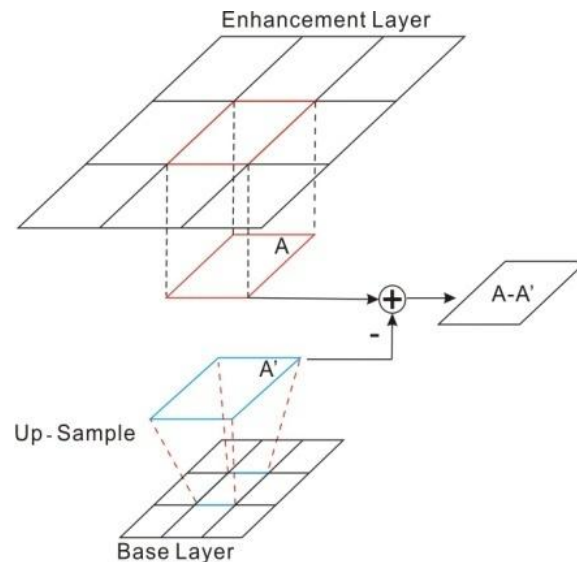


Fig.5. 3 Illustration of inter-layer intra prediction [5.12]

From the prediction flow of inter-layer prediction, we can observe a property that the needed data for inter-layer prediction can be known ahead before checking the inter-layer prediction mode for current encoding macroblock. Therefore, a data pre-fetch scheme, called Inter-layer Pre-fetch Scheme(IPS), for inter-layer prediction is proposed in this thesis. In our proposal, all information required for inter-layer prediction will be pre-fetched ahead before the inter-layer prediction. The design details will be described as follows.

5.1.3 Proposed Inter-layer Pre-fetch Scheme

For supporting spatial scalability, inter-layer prediction mechanisms are developed to enable the usage of as much lower layer information as possible for improving rate-distortion efficiency of the enhancement layers [5.13]. Fig.5. 4 shows the data relations for inter-layer prediction. When decoding advanced spatial layer of video frames, low layer frames will be referenced frequently. 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.5. 4 Data relations of three spatial layers for inter-layer prediction

For the inter-layer prediction, the prediction signals are usually formed by motion-compensated prediction inside 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 processor element read 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.5. 5 gives an explanation of the proposed IPS. In this figure, the green frame represents base layer frame and the blue frame represents enhancement layer frame. When reconstructing a macroblock(as illustrated by red block in Fig.5. 5), 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 in Fig.5. 5 illustrates the pre-fetch parts for IPS. By this proposed scheme, the cache hit rate can significantly been improved for inter-layer prediction.

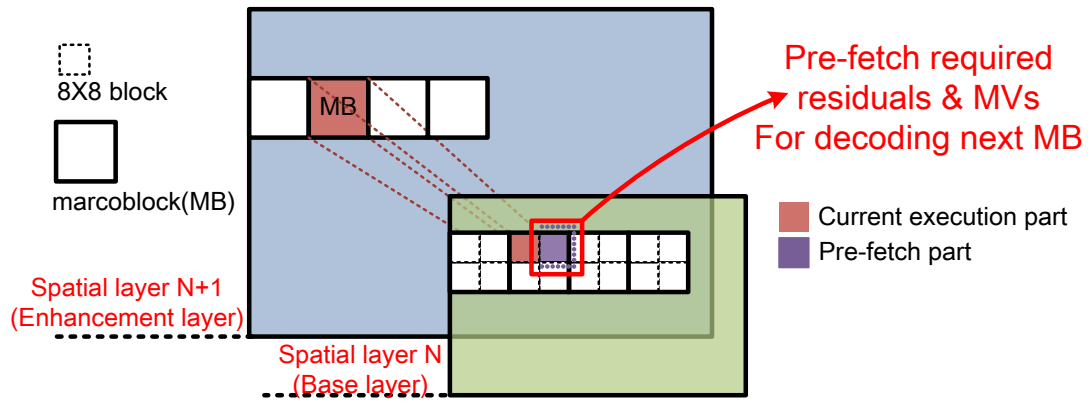


Fig.5. 5 Illustration of the Inter-layer Pre-fetch Scheme

For the proposed IPS, Pre-fetch Command Generator(PCG) is constructed in distributed memory management unit(d-MMU) as shown in Fig.5. 6. PCG receives the frame and macroblock information from SVC processor element and generates the corresponding pre-fetch commands. According to the pre-defined address allocation, PCG can produce the pre-fetch addresses and sent to the cache control unit. When the cache control unit is idle or waiting miss data back, the pre-fetch commands will be served. It 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). It can easily be pre-fetched during reconstructing a macroblock.

Proposed IPS may induce more d-MMU cache power consumptions produced by additional pre-fetch command accesses. However, the scheme can effectively reduce the cache miss rate by pre-fetching the base layer residuals and MVs. It can also reduce the unnecessary cache misses in L1 cache(d-MMU) by keeping useful data in the cache so that the number of L2 cache(c-MMU) access can be reduced, too. These effects can significantly reduce the total memory energy consumptions including on-chip cache and off-chip memory in the system. For more detail discussions and simulations, it will be described in section 5.3.

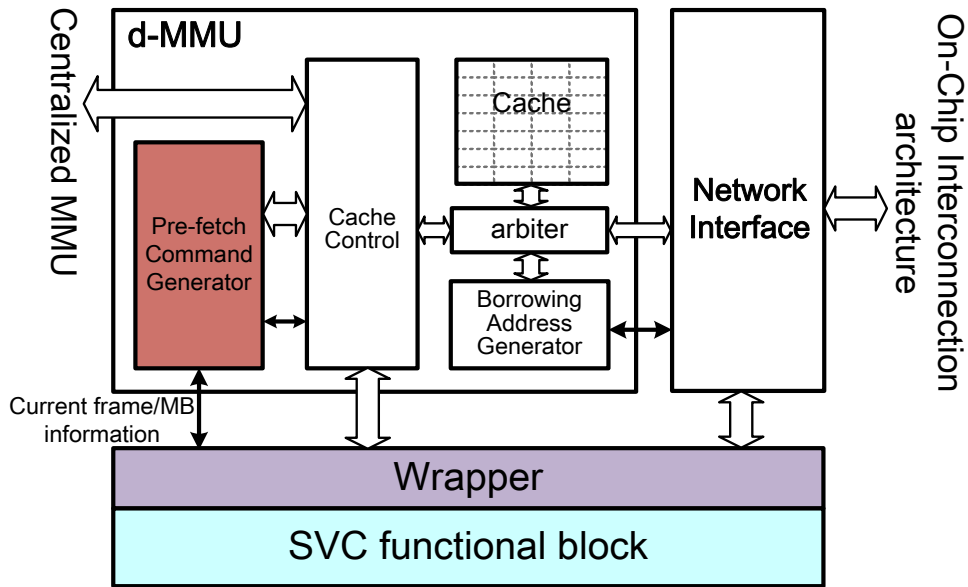


Fig.5. 6 d-MMU architecture with Pre-fetch Command Generator

5.2 Address Translator for SVC

5.2.1 Introduction

To improve memory bandwidth and power consumption in video applications, a new address translation machine is proposed. This address translation machine is used for SVC decoder. The advantage of the address translation machine is the accessing to external memory can become more regular. Since the translation can minimize the number of overhead cycles needed for row-activations in synchronous DRAM (SDRAM), the memory bandwidth and energy consumption can be improved significantly. The features of SDRAM and memory-access patterns of video-processing applications are considered to find a suitable address translation which can improve the performance of whole system. As the resolution of video-processing applications becomes high and H.264 supports the high compressing efficiency, video signal processors should deal with a large amount of data within a tightly bounded time. Due to the large amount of data transfer, video data are stored in the external memory that are usually slow, and thus the system performance strongly depends on the memory bandwidth between processors and external memories. The data transfer in the video decoder is especially huge in order to support different level and complex mode. To meet the requirement, we must exploit the characteristics of

video-processing algorithms. From the deterministic characteristic, most memory access patterns can be known at compile time. The regularity of memory-access patterns can be effectively used to reduce the number of clock cycles required in array accesses. Besides the high memory bandwidth, low-power consumption becomes an important factor to be considered in system design. As the power related to memory accesses dominates the whole system power in data-dominated systems, it is essential to reduce the memory power consumption. In the external memory, row-activation and pre-charge operation are dominant in dynamic power consumption. When the accesses to memory become more regular the number of active and pre-charge operation can be decreased. Therefore the dynamic power consumption is greatly decreased.

5.2.2 Centralized MMU with Address Translator

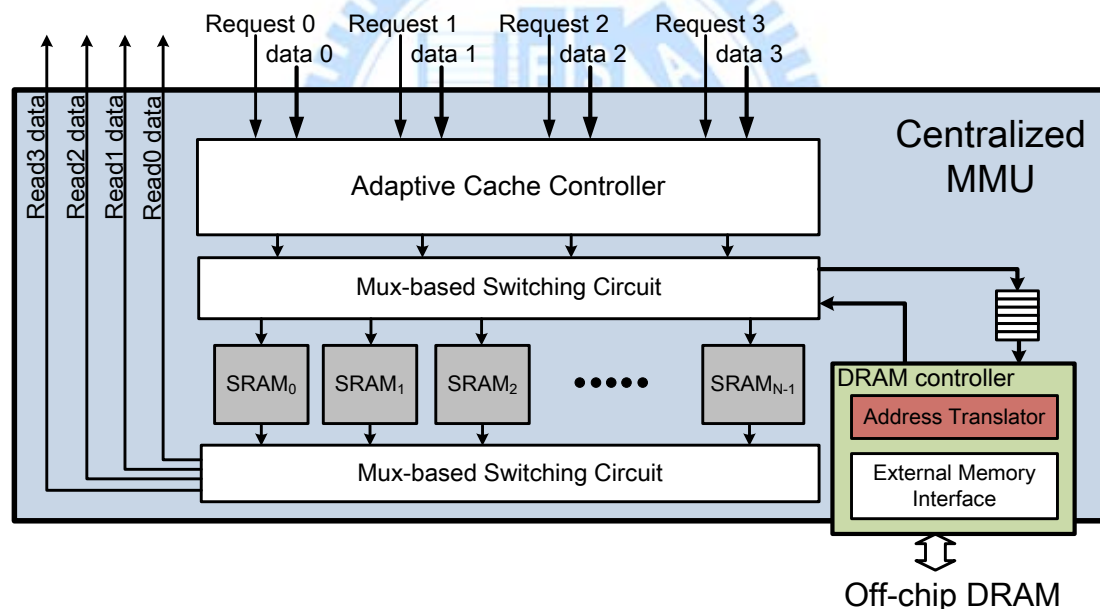


Fig.5. 7 Centralized MMU architecture with Address translator

According to the proposed c-MMU described in chapter 4, the Address Translator(AT) is a layer which translated the original access address to a suitable DRAM physical address in DRAM controller. Fig.5. 7 shows the simple block diagram of the c-MMU. When the cache needs to access the DRAM data, the address is translated to a physical address by AT represented by a red block in the Fig.5. 7. According to the re-generated physical address, the external memory interface(EMI) in DRAM controller can successfully access the DRAM data by generating

corresponding DRAM commands. Generally, the EMI use the characteristic of external memory and the AT use the characteristic of video processing to improve the DRM access performance.

In this work, DDR3 SDRAM provided by Micron Inc. [5.14] is applied. The configuration parameters of this DDR3 model are shown in Table.4. 2. In order to have high DRAM bandwidth, the 64MegX16 configuration is selected because the DRAM data bus width is the widest in these models. For the wireless video entertainment systems, two DDR3 devices are applied and share the same bus. The detailed DRAM organization and data arrangement will be described in the next section.

5.2.3 Data Arrangement

The DRAM organization is shown in Fig.5. 8. There are two DDR3 devices in our system, In order to reduce the chip I/O port, these two DRAMs share the same address, data and command bus controlled by chip select signal. Because the SVC has huge memory requirement and the regular data access behavior, we particularly arrange a DRAM for SVC processor element, and the other DRAM is arranged for another processor elements.

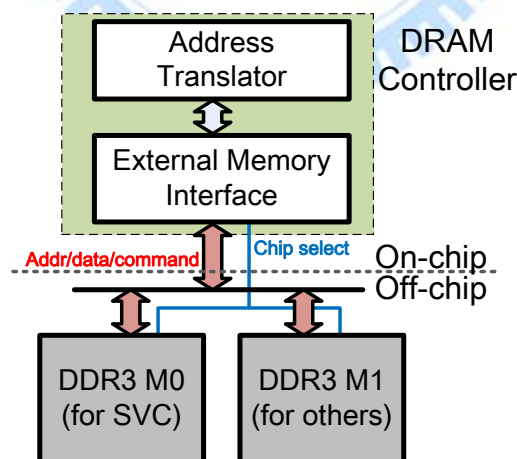


Fig.5. 8 Architecture of the DRAM organization

5.2.3.1 Conventional Memory Mapping Method

In general cache memory system, adjacent blocks would address to the same row,

same bank and same rank as much as possible for open-page DRAM policy to reduce the row conflict and bank conflict. The selected DDR3 size parameters are shown in Table.5. 1 and the related addressing parameters are shown in Table.4. 2. With the conventional memory mapping method, the address translation between cache address and DRAM physical address can be accomplished which the mapping scheme is shown in Fig.5. 9. One bit Byte offset is set because the data width of the selected DRAM configuration is 2-bytes(16-bits). And then, 10 bits for DRAM 1K column address, 3 bits for 8 bank address and 13 bits for 8K row address. As this mapping scheme, a block access can locate in the same bank and the same row so that the row-conflict and bank-conflict would not occur. For the adjacent block access, the row-conflict and bank-conflict probability can be minimized.

Configuration	8 Meg x 16
Number of banks	8
Column Size	2KB
Row size	8K columns = 16MB
Total size	128MB

Table.5. 1 Selected Micron DDR3 size parameters

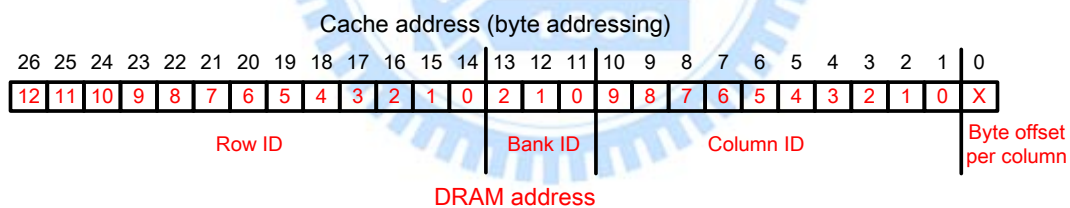


Fig.5. 9 Conventional mapping scheme for the selected DRAM

5.2.3.2 Proposed Memory Mapping Method for SVC

The video processing has the regular memory access behavior according to the deterministic coding scheme. The conventional mapping scheme would not be the most suitable scheme for video data arrangement. In the advanced video coding standard, bidirectional prediction is applied to reduce bit rate. Fig.5. 10 shows the temporal and decoding relations of video frames in a group of pictures(GOP), which is supported by the SVC in our system. In order to reduce the DRAM row miss rate, the video frames are allocated to different banks according to the decoding references.

For instance, F4 will probably reference F0 or F8 to reconstruct the frame, so these three frames are allocated to different banks. Hence, the video decoder writes the reconstruct data to the new DRAM bank in regular, and would not be disarranged by read. It enables high row-hit rate for data write because of the regular write behavior for reconstructing a frame. Compared to the modern memory mapping methods which have been used in many works for video frames [5.15]-[5.17], the proposed memory mapping method is more suitable for the traditional cache-based hierarchy memory system because the row-hit rate of the DRAM write can be improved. Although the row-hit rate of the DRAM read would be higher than the modern method, 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 would strongly depend on the range of search window defined in the encoder. The write is more regular than the read for reconstructing frames because the write data is in raster-scan sequence but the read data may in random sequence. Accordingly, the proposed method can have low row-miss rate with reducing the DRAM row-conflict caused by writing reconstruct frame data.

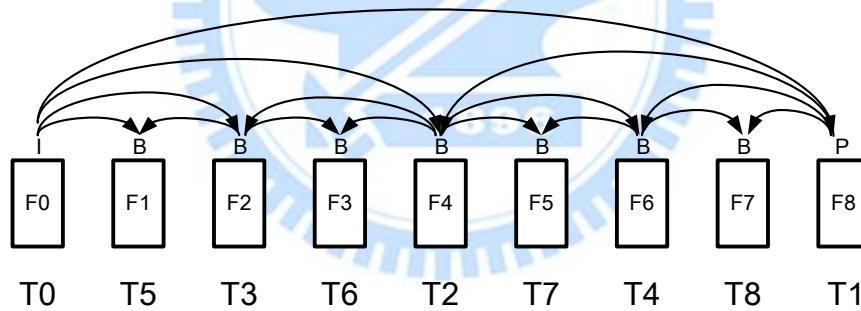


Fig.5. 10 Video frame arrangement of a GOP

The detail memory mapping method is shown in Fig.5. 11. Take the QCIF resolution as an example, the reconstruct Luma and Chroma data are stored into the bank which is assigned by the bank interleaved scheme. Luma data will assigned to bank 0, bank 1 or bank 2 and the Chroma data will assigned to bank3, bank4 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 store in to the same row. Fig.5. 11 shows the QCIF frame data allocation in the DRAM. The read data for motion compensation may be irregular, but the produced

reconstruct data is in raster-scan sequence. By the proposed mapping scheme, the data read and write for the reconstruct processing address to different banks. The produced reconstruct 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.

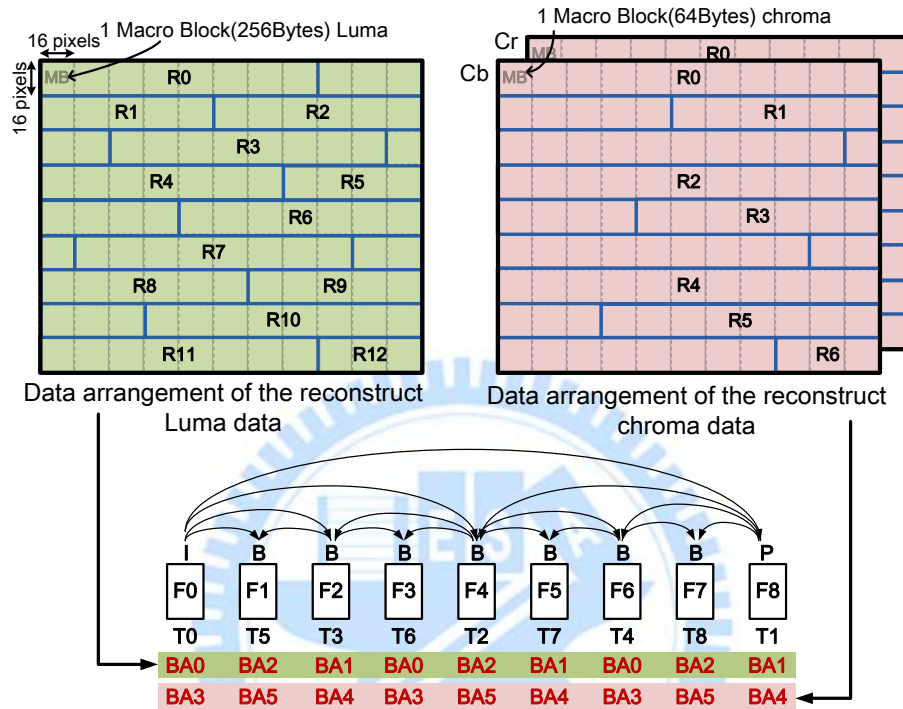


Fig.5. 11 Frame map to memory

5.3 Analysis & Simulation Results

In this section, the results of proposed pre-fetch and data allocation methods will be described in section 5.3.1 and 5.3.2, respectively. For the simulation, The SVC application is applied, and the summary is listed in Table.5. 2. Note that the latency and energy estimation methods have been introduced in chapter 4.

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-B-P)
Sequence	Stephen

Table.5. 2 Summary of SVC information

5.3.1 Improvement of adding IPS

With proposed IPS as mentioned in section 5.1, the miss rate of distributed memory (L1 cache) can be reduced for the SVC application. Fig.5. 12 shows the simulation result of the miss rate. By adding the pre-fetch scheme (shown in Green line), it can reduce 30.01% on average. Note that 4-way associativity cache configuration and Least Recently Used (LRU) replacement policy are applied in the simulation. The simulation for other associativity is done for 64KB cache size, and it is shown in Fig.5. 13. As the observation, higher way-associativity may have no obvious miss-rate reduction when the number of way is over 4 for this test pattern, so we choose a 4-way associativity configuration for L1 cache.

Furthermore, IPS can reduce unnecessary cache misses in L1 cache by keeping useful data, which will be accessed recently, in the cache, so the number of L2 memory access can also be reduced. Moreover, the number of DRAM access caused by L2 cache data replacement also can be reduced. As shown in Fig.5. 14 and Fig.5. 15, the number of memory access in the centralized memory(L2 cache) can be reduced by 24.6% and the number of DRAM access can be reduced by 34% on average.

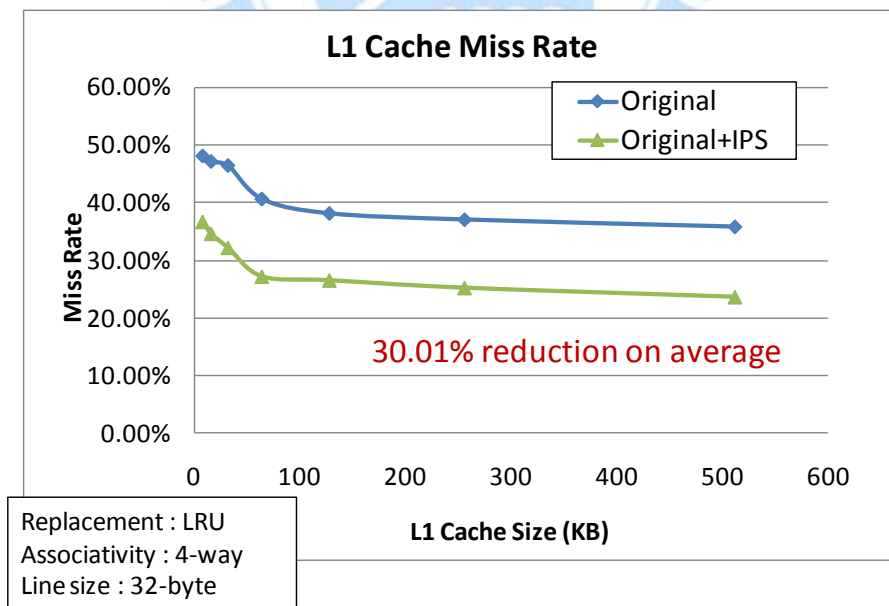


Fig.5. 12 Miss rate of the L1 cache versus L1 cache size

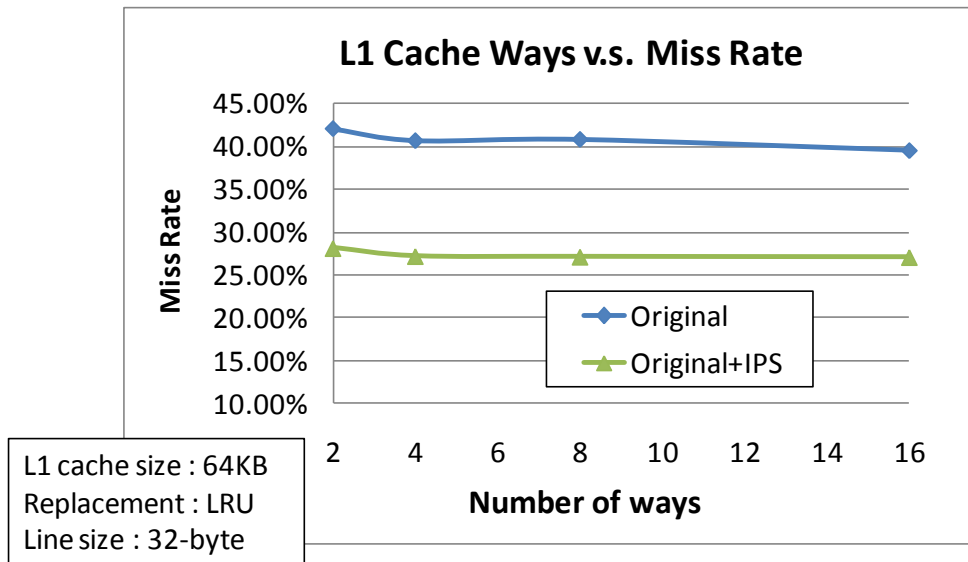


Fig.5. 13 L1 cache ways v.s. Miss Rate

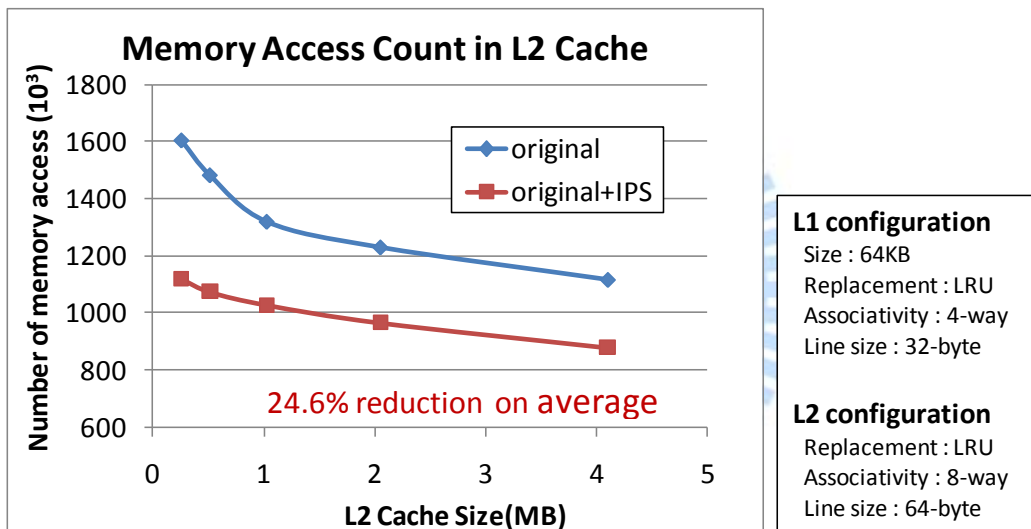


Fig.5. 14 Memory access count of L2 Cache

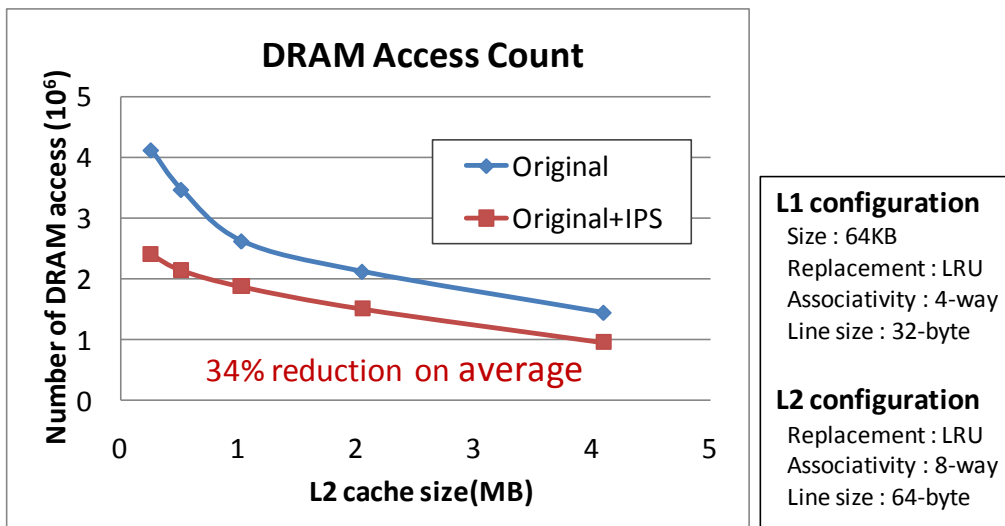


Fig.5. 15 DRAM access count

Pre-fetch mechanism may induce additional energy overhead because there have extra cache access generated by pre-fetch requests. By using CACTI model, the energy measurement of L1 cache can be achieved and the result is shown in Fig.5. 16. The green part is the pre-fetch energy overhead produced by the miss pre-fetch requests. The red part is standby leakage energy consumption, and the blue part is the access energy. As shown in this figure, the access energy would increase with pre-fetch mechanism because of the additional pre-fetch requests. With pre-fetching, the standby leakage energy can be saved because it reduces the cache miss rate and execution time. By the cache configuration as illustrated in Fig.5. 16, pre-fetch have additional 18.9% energy overhead compared to the original design. Although the IPS may induce larger energy consumptions in d-MMU, it can reduce the execution time and number of L2 cache access so that total memory energy consumption would be reduced. The detail simulation will be introduced in the next section.

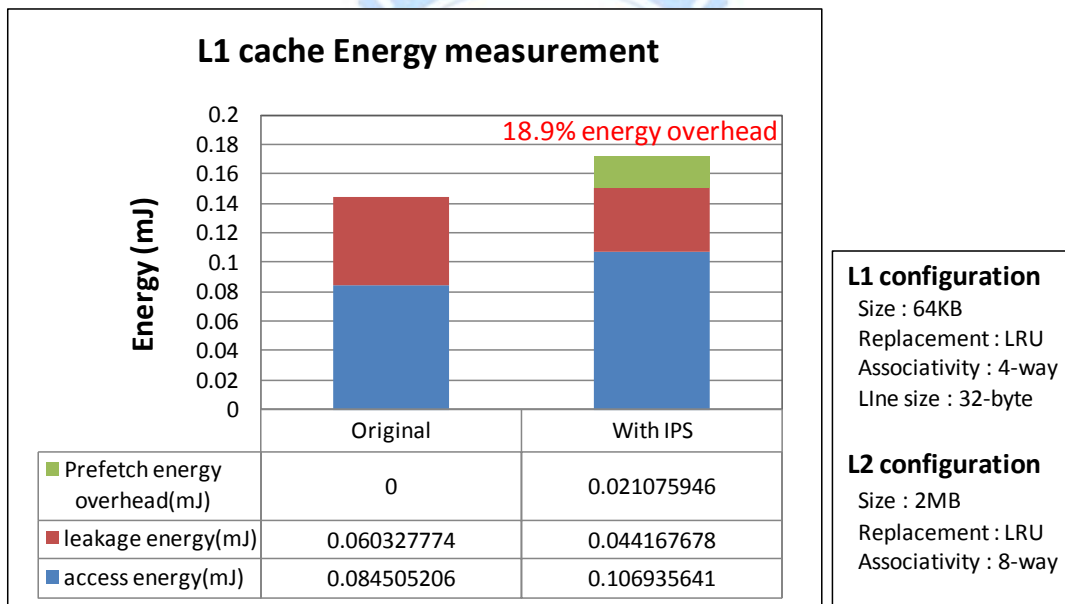


Fig.5. 16 L1 cache energy measurement

5.3.2 Improvement of adding Address Translator

The address translator converts the original address to a suitable DRAM address according to the proposed data allocation method as mentioned in section 5.2, and it can reduce the DRAM row-miss rate successfully. Fig.5. 17 and Fig.5. 18 show the simulation result of the DRAM row-miss rate and number of DRAM row-conflict respectively with different L2 cache size. Compared to the original data allocation as

described in section 5.2.3.1, the proposed frame data allocation method have lower row-miss rate and number of row-conflict in DRAM for video application. In these figures, blue line shows the row-miss rate of original allocation method. With pre-fetch mechanism, the row-miss rate and row-conflict also can be reduced because the number of DRAM access is reduced. The red line shows the row-miss rate of adding pre-fetch mechanism and the green line shows the row-miss rate of adding pre-fetch and proposed data allocation mechanisms. On average, 60.64% row-miss rate and 74.35% number of row-conflict reduction can be achieved by adding these proposed methods.

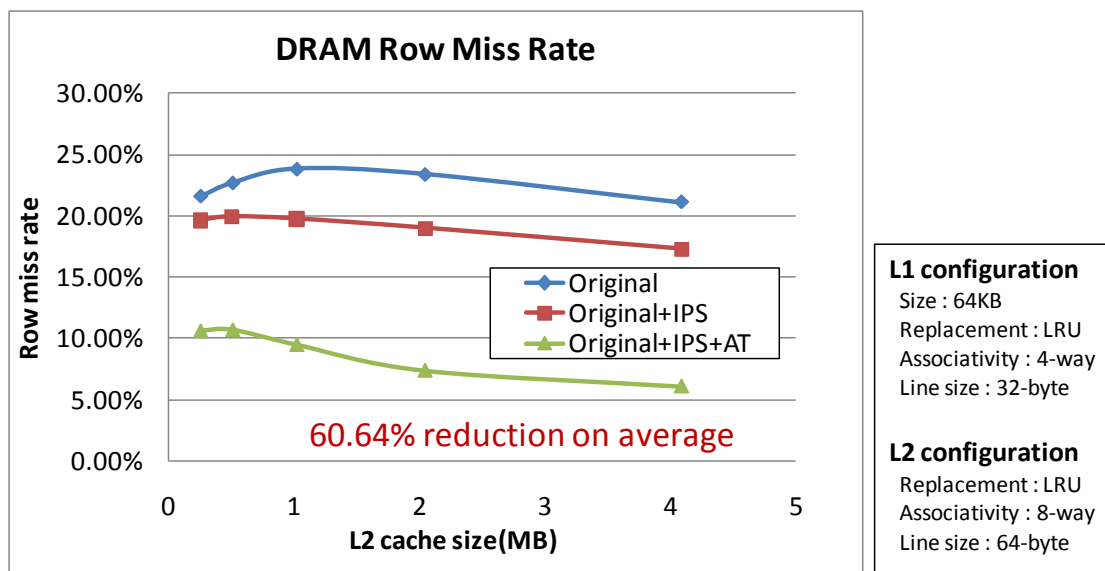


Fig.5. 17 DRAM row-miss rate

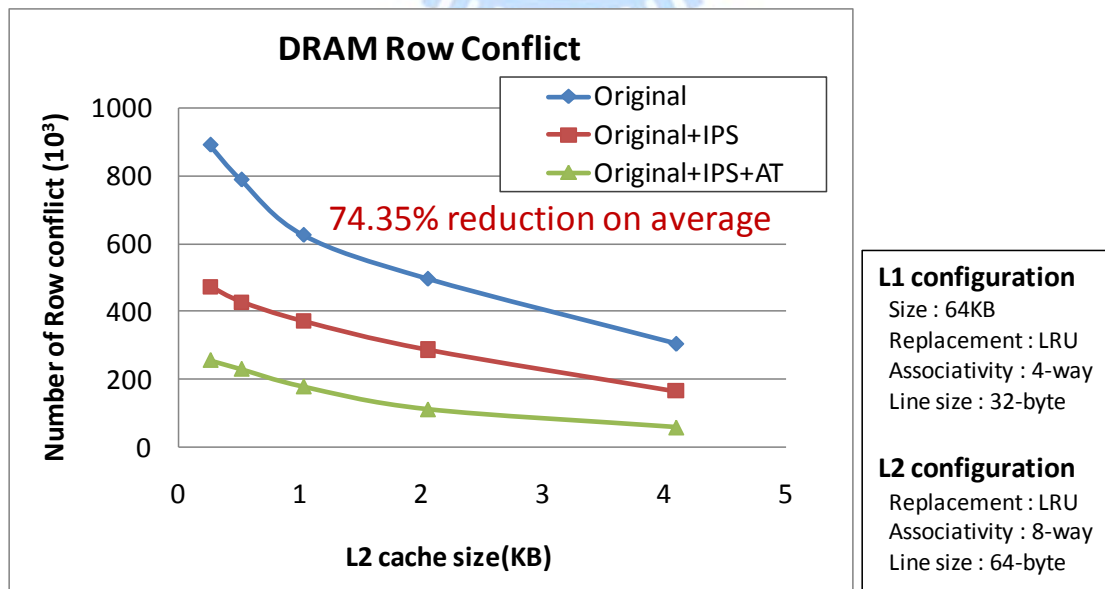


Fig.5. 18 Number of DRAM row-conflict

Reducing row-miss rate can decrease the DRAM activate power and improve the DRAM access bandwidth utilization. Fig.5. 19 shows the measurement result of the DRAM activate power. As expected, the proposed pre-fetch and data allocation method can reduce the activate power about 57.19% on average. For measuring DRAM memory access efficiency, we define the bandwidth utilization as shown in the following equation to calculate the DRAM bandwidth efficiency.

$$\text{DRAM Bandwidth Utilization} = \frac{\text{Total cycles of outputting and inputting data between DRAM}}{\text{Total cycles of processing access commands}} \times 100\%$$

The simulation result is shown in Fig.5. 20, and the proposed methods can improve 24.87% DRAM bandwidth utilization compared to the original method.

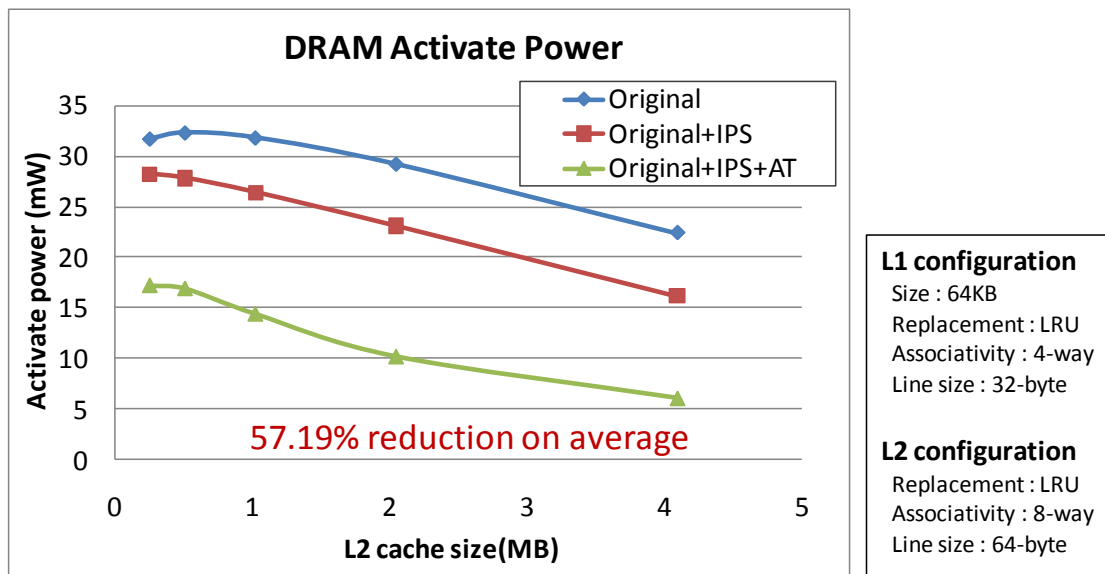


Fig.5. 19 DRAM activate power

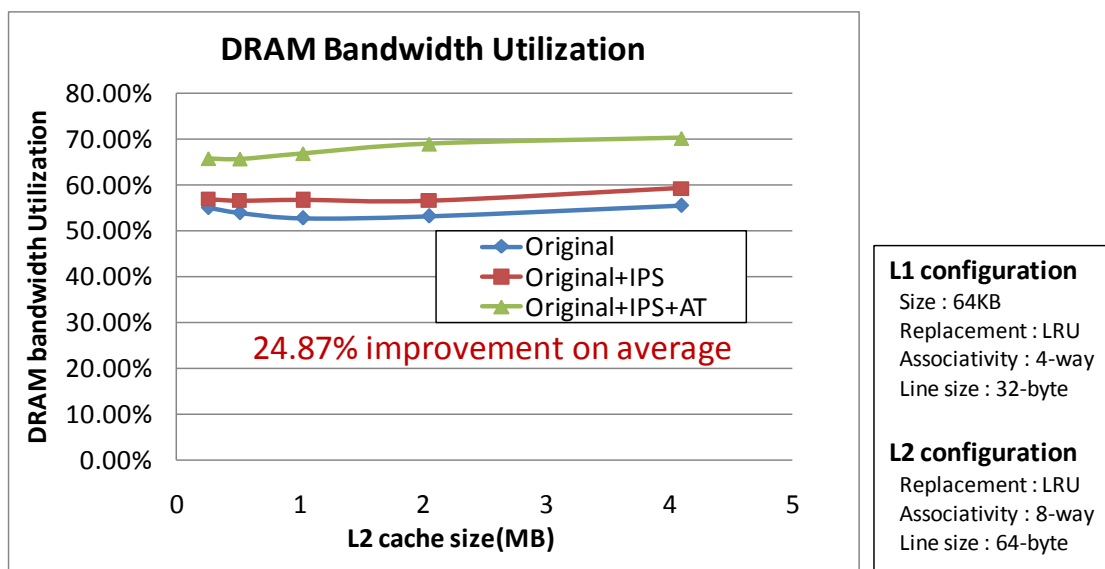


Fig.5. 20 DRAM bandwidth utilization

Consequently, DRAM energy consumption and total execution time and can be reduced by the proposed mechanisms as shown in Fig.5. 21 and Fig.5. 22 respectively. The DRAM read/write power dominates the DRAM power consumption. Reducing number of DRAM access is the most efficient way for decreasing the DRAM energy consumption. In our observation, the pre-fetch mechanism can reduce the execution time and energy consumption significantly because it directly reduces the number of DRAM access with lowering cache miss probability. Furthermore, activate power can be saved by address re-allocation mechanism so that DRAM energy can be reduced again. For execution time, pre-fetch mechanism significantly reduces the cache miss rate, so it has lower average memory access time than the original. By DRAM data re-allocation, the average miss penalty can be reduced. Accordingly, the proposed mechanisms reduce the execution time successfully.

In addition, we measure total on-chip cache energy consumption with different L2 cache size as shown in Fig.5. 23. Larger L2 cache size has more cache energy consumptions, but lower DRAM energy consumption can be achieved because of the low cache miss-rate. Fig.5. 24 illustrates the total memory energy consumption including on-chip cache and off-chip DRAM with different L2 cache size. As shown in this figure, larger L2 cache size has lower total memory consumption in the range from 256KB to 4MB. On average, 37.53% energy reduction can be achieved when adding proposed mechanisms.

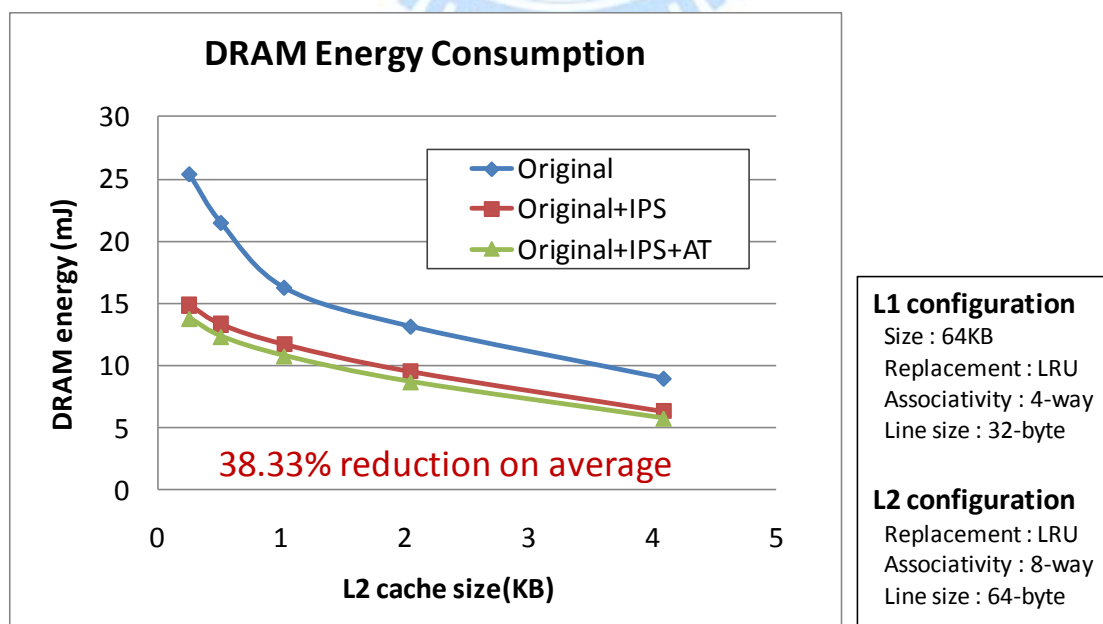


Fig.5. 21 DRAM energy consumption

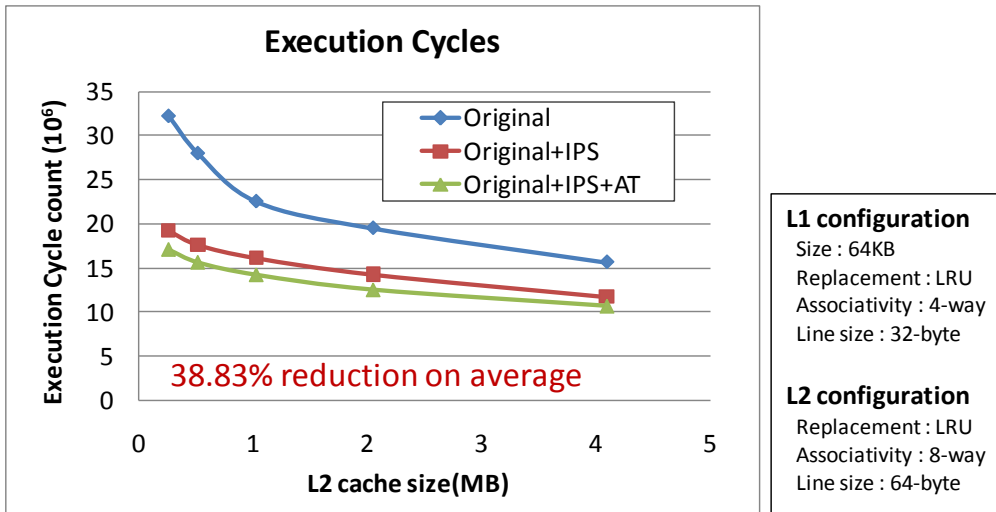


Fig.5. 22 Total Execution cycles

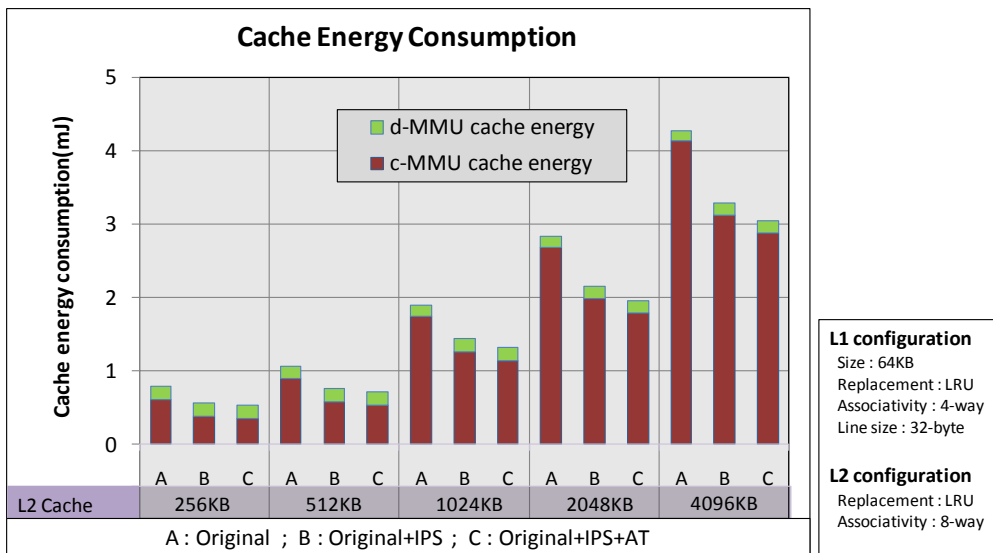


Fig.5. 23 On-chip cache energy consumption

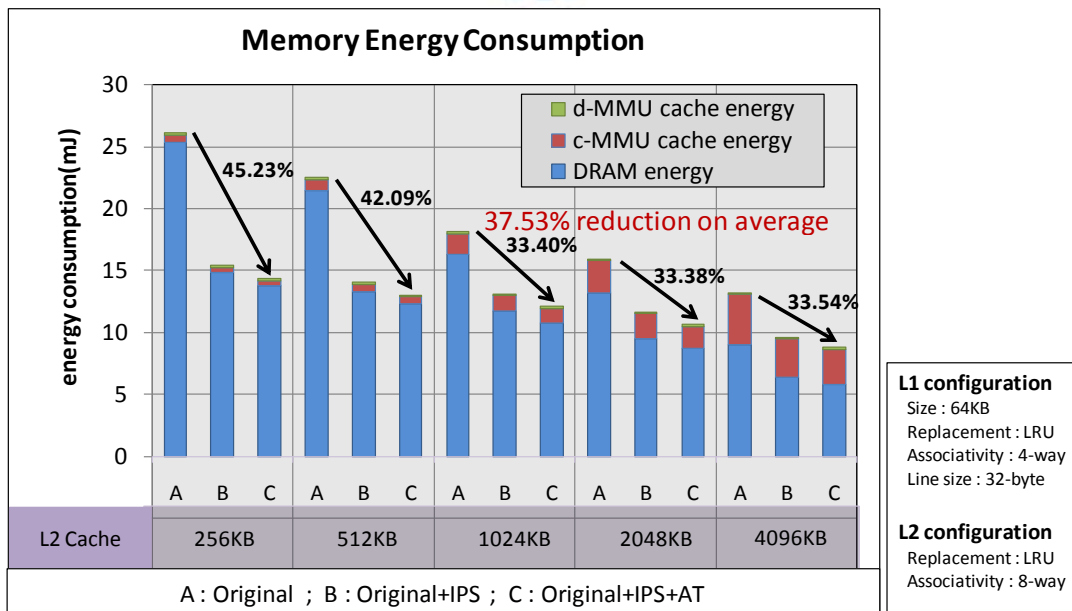


Fig.5. 24 Total memory energy consumption

5.3.3 Analysis and Simulation Results of Adaptive Cache Control for Wireless Video Entertainment Systems

For wireless video entertainment systems, 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 [5.18]. 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 together with the base layer and can provide better visual quality. The complete bitstream (i.e., combination of all the layers) provides the highest quality. In receiver, according to different channel situation or different application in end-user device, the most suitable quality and resolution of the video can be reconstructed by SVC. Fig.5. 25 illustrates the video performance for different channel bit rate. In this figure, the distortion-rate curve represents the upper bound in quality for any coding technique at the given bit rate. With SVC technique, the non-scalable single staircase curve is changed to a curve with several stairs.

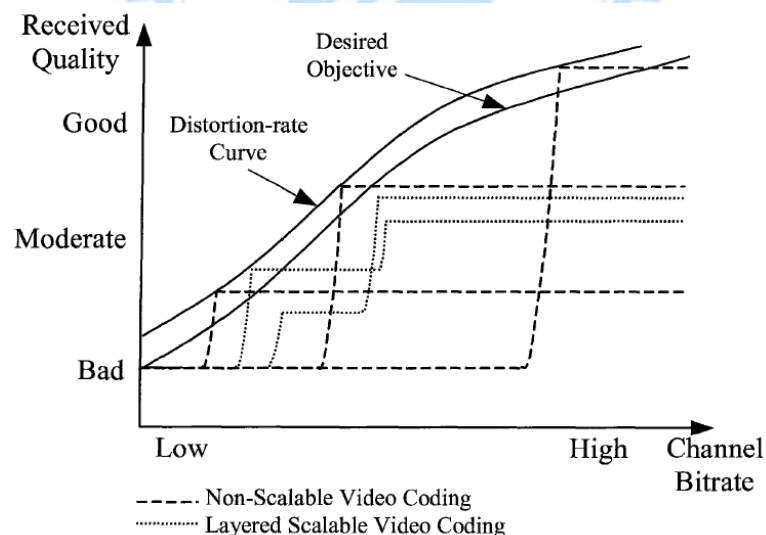


Fig.5. 25 Video coding performance [5.18]

The most memory starved processor element in wireless video entertainment systems is SVC because the video coding needs huge memory to store the frame data. However, decoding different scalable level in SVC would have 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.5. 26. It

is quite different when various scalable layers are decoded. Therefore, adaptive cache control in c-MMU can be used for optimizing the on-chip memory utilization of SVC.

In wireless video entertainment systems, 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 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, suitable bank assignments for SVC in different situations can be achieved.

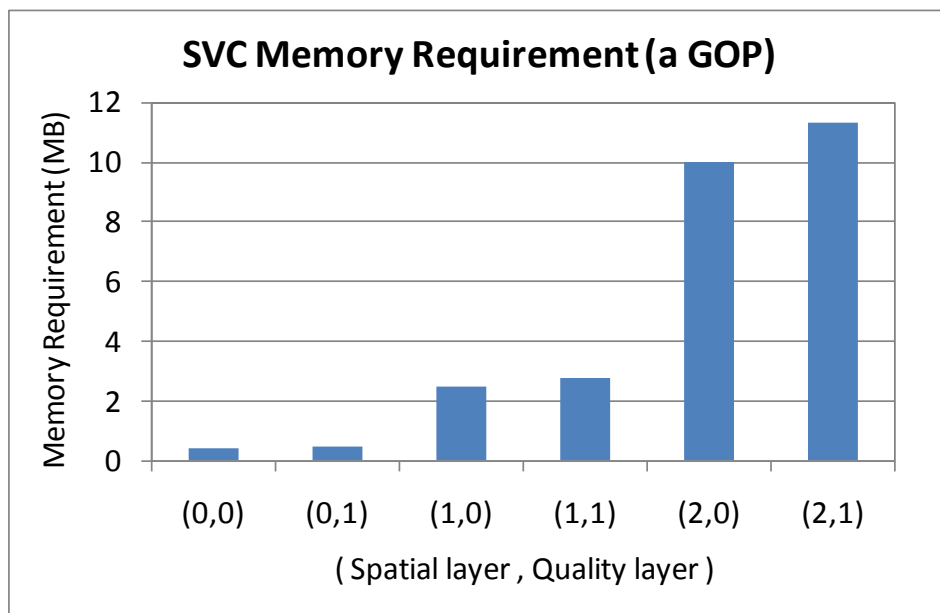


Fig.5. 26 SVC memory requirements of different scalable layers for a GOP

For wireless video entertainment systems, the summary of memory configuration is listed in Table.5. 3. The bank assignment in c-MMU is also been profiled for different SVC decoding levels as shown in Table.5. 4. When SVC needs to decode high spatial and quality layer frames, adaptive bank assignment enables c-MMU to assign more banks for SVC. It can reduce the miss rate of frame reconstructing so that number of DRAM access can be decreased. In contrast, adaptive bank assignment can turn-off some banks in c-MMU when decoding low spatial and quality layer frames. By this configuration, Fig.5. 27 shows the execution time for decoding a GOP with different SVC level. Compared to the fixed bank assignment (every processor elements assign equal banks), adaptive bank assignment can reduce the execution time for decoding enhancement layers in SVC. In addition, the memory energy

comparison is done, and the result is shown in Fig.5. 28. Note that IPS and data allocation mechanism which are proposed in section 5.1 and 5.2 are applied in the simulation.

L1 cache (d-MMU) configuration	
Cache Size	64KB
Number of banks	2
Associativity	4-way
Block size	32-byte
Replacement policy	LRU
Write policy	Write back
L2 cache (c-MMU) configuration	
Cache Size	2MB
Number of banks	16
Associativity	N-way, $1 \leq N \leq 16$ (depend on bank assignment)
Block size	64-byte
Replacement policy	LRU
Write policy	Write back
External Memory configuration	
Device	DDR3 SDRAM
Channel/Rank/Bank	1/1/8
Size	128MB
Number of banks	8
Burst length	Fixed to 8
DRAM Page Policy	Open page policy

Table.5. 3 List of simulation information

SVC level (Spatial layer, Quality layer)	(1,0)	(1,1)	(2,0)	(2,1)
Bank Assignment (processor element --> # of banks)	WPU → 1 MAC → 2 LT → 2 SVC → 7	WPU → 1 MAC → 2 LT → 2 SVC → 8	WPU → 1 MAC → 1 LT → 1 SVC → 13	WPU → 1 MAC → 1 LT → 1 SVC → 13
Number of turn-off banks	3	2	0	0

Table.5. 4 c-MMU bank assignment for wireless video entertainment systems

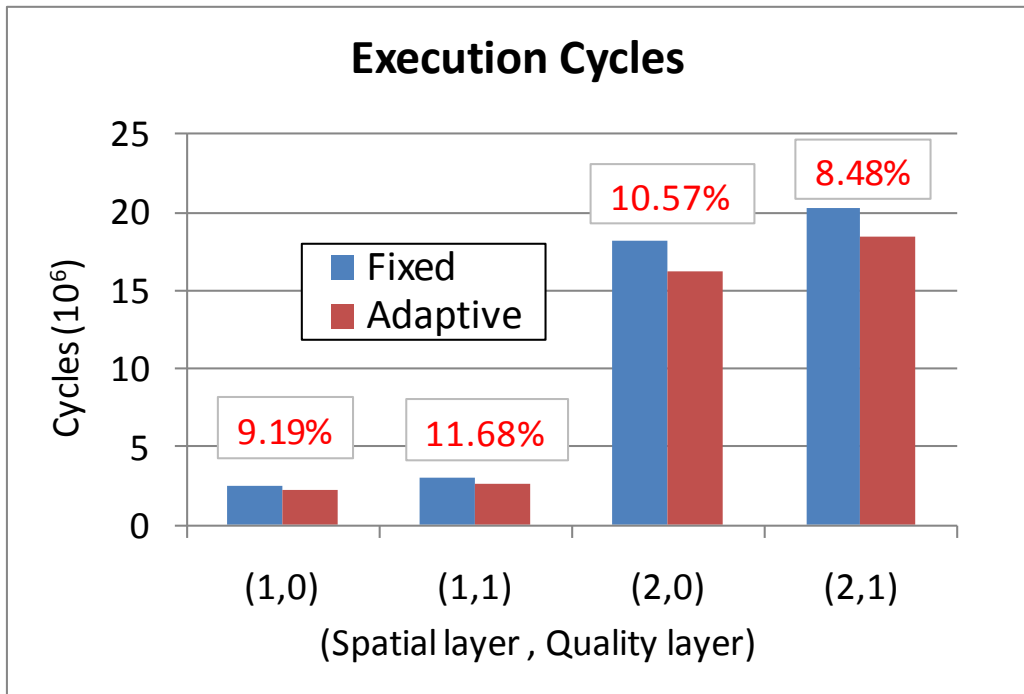


Fig.5. 27 Execution cycles for different SVC levels

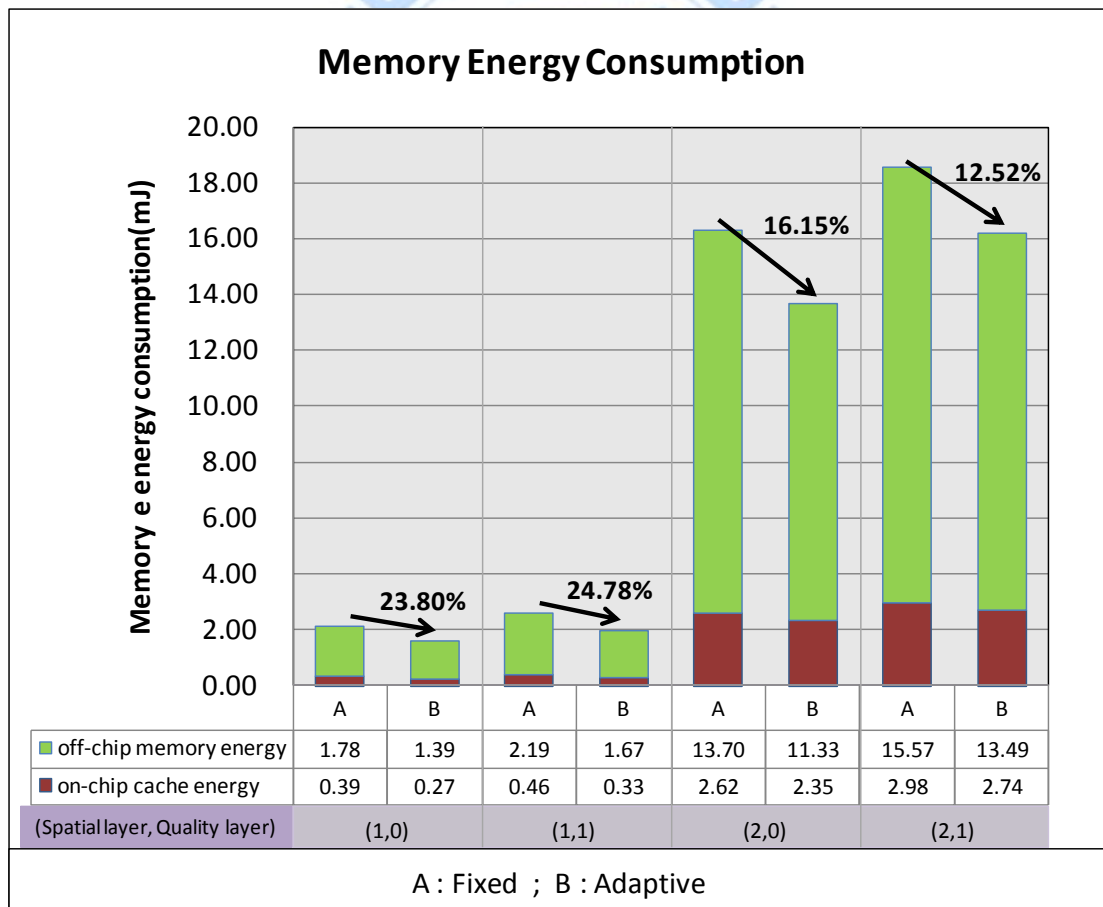


Fig.5. 28 Memory energy consumption for different SVC levels

Proposed c-MMU can support cache reconfiguration for different bank assignments in different time intervals by checking BAT. Assuming BAT can be

updated by MAC at runtime, dynamic bank assignment can be achieved. Here a simulation for dynamic bank assignment is done. Fig.5. 29 shows the relation between time interval and decoding SVC level in the simulation. The corresponding bank assignment of different SVC level is listed in Table.5. 4. To simplify the simulation, a GOP is decoded in each time interval. The simulation results of execution cycle and memory energy consumption are shown in Fig.5. 30 and Fig.5. 31, respectively. As result, 7.13% execution time and 10.53% memory energy consumption reduction can be achieved.

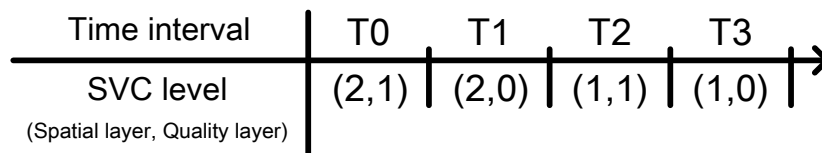


Fig.5. 29 Relation between simulation time interval and decoding SVC level

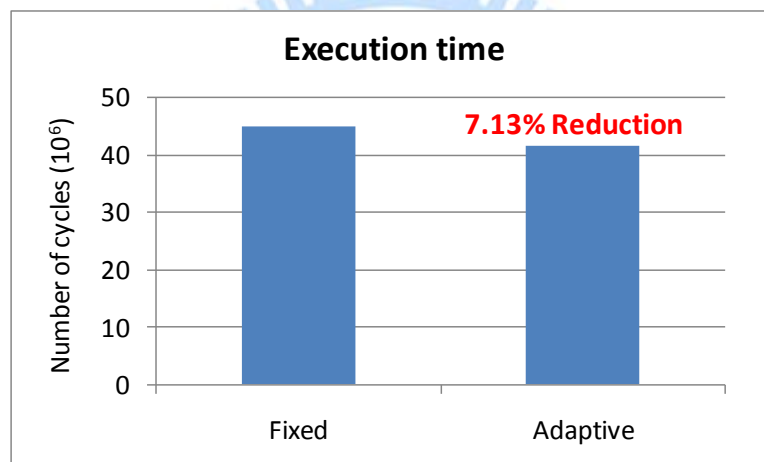


Fig.5. 30 Simulation result of total execution cycles

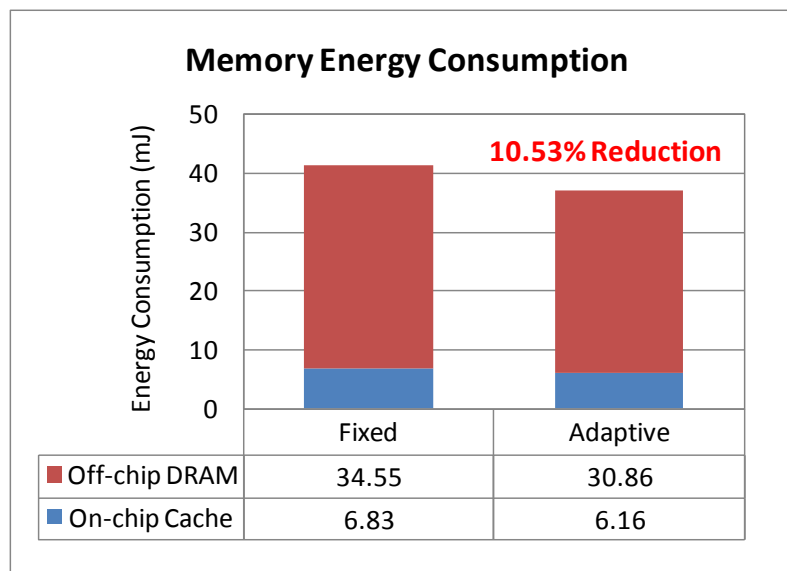


Fig.5. 31 Simulation result of memory energy consumption

5.4 Summary

The memory management units (MMUs) have been constructed for the on-demand memory system in the chapter 4. In this chapter, additional inter-layer pre-fetch scheme (IPS) and address translation mechanism are proposed and integrated in MMUs to improve the performance for scalable video coding (SVC) processor element. These proposed methods not only reduce cache miss rate but also reduce total memory energy consumptions.

For proposed IPS, the required information for inter-layer prediction in SVC technique will be pre-fetched ahead when reconstructing the frames, so the cache miss can be reduced significantly. Furthermore, IPS can reduce unnecessary cache misses in L1 cache and the number of DRAM access caused by cache data replacement. Accordingly, the execution time and memory energy consumptions can be reduced by IPS. In d-MMU of on-demand memory system, pre-fetch command generator (PCG) is constructed for generating the pre-fetch commands. Even though proposed IPS may have additional power overhead in d-MMU, the overall memory energy consumptions including total on-chip cache and off-chip DRAM can significantly be reduced.

Additionally, to improve DRAM memory bandwidth efficiency and reduce DRAM power consumption, a new address translation mechanism for video applications is proposed in this chapter. It is achieved by an address translator in DRAM controller constructed in c-MMU. Proposed translation mechanism can minimize the number of overhead cycles needed for row-activations in DRAM. In the advanced video coding standard, bidirectional prediction is applied. In order to increase the DRAM row hit rate, the video frames are allocated to different banks according to the decoding and reference sequence. With bank interleaved scheme, the video decoder writes the reconstruct data to the new DRAM bank in regular, and would not be interrupted by read. It enables high row-hit rate for data write because of the regular write behavior for reconstructing a frame.

With Proposed IPS and DRAM data allocation, the execution time and energy consumption can be saved. On average, 38.83% execution time reduction and 37.53% memory energy reduction can be achieved for different L2 cache size when decoding a GOP of SVC. In addition, decoding different layers in SVC would have different

memory requirements. The adaptive cache control in c-MMU can be applied for optimizing memory utilization in wireless video entertainment systems. With profiling the memory requirements for decoding different SVC spatial and quality layers, suitable bank assignment in c-MMU can be determined off-line. According to the profile information, MAC could update BAT in c-MMU and achieve the adaptive bank assignment. Hence, the optimizing memory utilization for the system can be realized. Moreover, assuming MAC can dynamically detect the channel situation and control the scalable level of SVC. The BAT is also can be update dynamically, so the adaptive memory resource allocation can be achieved at runtime.



Chapter 6

Conclusions and Future Work

6.1 Conclusions

For constructing a multi-task platform, how to manage and utilize the memory is an important issue. This thesis proposes a message-passing based memory-centric on-chip data communication platform with on-demand memory system, and it can be applied for wireless video entertainment systems. In on-demand memory system, memory management units (MMUs) can efficiently control the memory access and memory resource allocation for processor elements (PEs).

Proposed distributed memory management unit (d-MMU) performs as a high level cache for the dedicated PE in the on-demand memory system. Burst-based memory access protocol is applied to access continuous data easily, and the cache miss penalty also can be hidden. Furthermore, in order to reduce the stall caused by high traffic in network interconnection and small packet buffer size in network interface, a novel buffer borrowing mechanism is proposed. It enables d-MMU to borrow the cache blocks for buffering the blocking packets from PEs. The simulation result shows that number of transferred packet can be increased when the packet buffer size is small, and the execution time of PE can be reduced because the stall has been reduced.

Centralized memory management unit (c-MMU) is designed for managing and providing larger centralized memory resources for system. PEs may have different memory requirements at runtime. With adaptive cache control, proposed c-MMU can support cache resource re-allocation for different PEs. By assigning suitable number of SRAM banks to PEs, the utilization of centralized on-chip cache can be optimized. Additionally, an external memory interface (EMI) in DRAM controller is applied to access external memory efficiently. By re-scheduling DRAM commands, the effective bandwidth of DRAM can be improved.

For SVC in wireless video entertainment systems, inter-layer pre-fetch scheme (IPS) and address translation mechanism are proposed and integrated in MMUs to improve the decoding performance. These proposed methods not only reduce cache miss rate but also reduce total memory energy consumptions. For proposed IPS, the

required information for inter-layer prediction in SVC technique will be pre-fetched ahead when reconstructing the frames so that the cache miss can be reduced significantly. The simulation result shows that even though proposed IPS may have additional power overhead, the overall memory energy consumptions including total on-chip cache and off-chip DRAM can significantly be reduced.

Furthermore, an address translator for video applications in DRAM controller is proposed to improve DRAM memory bandwidth efficiency and reduce DRAM activated power consumption. In general, bidirectional prediction technique is applied in the advanced video coding standard. With proposed bank interleaved scheme for frames, the video decoder writes the reconstruct data to the new DRAM bank in regular, and would not be interrupted by read. It enables high row-hit rate for data write because of the regular write behavior for reconstructing a frame. By simulation, IPS and DRAM data allocation mechanisms can reduce 38.83% execution time and 37.53% memory energy consumption for different L2 cache size when decoding a GOP of SVC.

In addition, adaptive cache control in c-MMU can be applied for optimizing memory utilization in wireless video entertainment systems. Decoding different scalable levels in SVC would have different memory requirements. With profiling the memory requirements for different SVC spatial and quality layers, suitable bank assignment in c-MMU can be determined. Assuming MAC can dynamically detect the channel situation and control the scalable level of SVC. According to the profile information, MAC could dynamically update BAT in c-MMU and achieve the adaptive bank assignment. Therefore, the optimizing memory utilization for the system can be realized at runtime.

6.2 Future Work

For designing the reconfigurable cache, a simple associativity-based partitioning scheme is used in this work. However, the flexibility of cache configuration would be limited. More fashion partitioning method can be used for improving the configuration flexibility such as the works in [6.1]-[6.4]. Moreover, the profiling of memory behaviors is done off-line in our work. Ideally, it would be achieved by a powerful profiling engine in system. We assume MAC in wireless video

entertainment systems can handle the profiling task. In the future, more complete profiling mechanism will need to be constructed.

Additionally, the standby power of DRAM would induce large static power. Modern DRAM devices can support sleep mode for reducing the standby power significantly. When the system access DRAM infrequently, sleep control mechanism can be applied to power-down the banks in DRAM dynamically. The mechanism can control by the powerful profiling engine which can detect the memory behavior at runtime. Therefore, the overall memory energy consumptions can be reduced.

The eHome project is still going on. For eH-III project, a femtocell home multimedia center will be developed for supporting multi-view 3D video, high-speed MIMO OFDM and gigabit cross-layer RRM in a heterogeneous platform. The architecture is shown in Fig.6. 1. In the future, in order to support huge memory bandwidth and data transmitting requirements, it will be necessary that constructing a heterogeneous memory-centric multi-core platform for multimedia center.

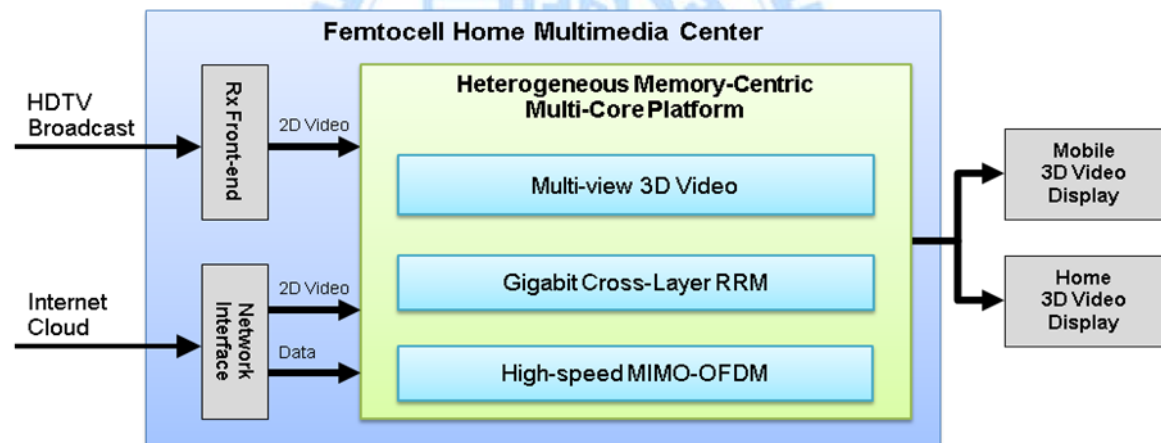


Fig.6. 1 Architecture of femtocell home multimedia center

Bibliography

References of Chapter 1

- [1.1] 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 (ISSCC)*, pp.100-101, 7-11 Feb. 2010.
- [1.2] H. Kondo, S. Otani, M. Nakajima, O. Yamamoto, N. Masui, N. Okumura, M. Sakugawa, M. Kitao, K. Ishimi, M. Sato, F. Fukuzawa, S. Imasu, N. Kinoshita, Y. Ota, K. Arimoto, T. Shimizu, "Heterogeneous Multicore SoC With SiP for Secure Multimedia Applications", *IEEE Journal of Solid-State Circuits*, pp.2251-2259, Aug. 2009.
- [1.3] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, Liewei Bao, J. Brown, M. Mattina, Chyi-Chang Miao; C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, J. Zook, "TILE64 - Processor: A 64-Core SoC with Mesh Interconnect", *IEEE International Solid-State Circuits Conference, 2008, ISSCC 2008*, pp.88-598, 3-7 Feb. 2008.
- [1.4] 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.
- [1.5] Yo-Sung Ho; Kwan-Jung Oh, "Overview of Multi-view Video Coding", *14th International Workshop on Systems, Signals and Image Processing*, pp.5-12, 27-30 June 2007.
- [1.6] G. Delagi, "Harnessing technology to advance the next-generation mobile user-experience," *Digest of Technical IEEE International Solid-State Circuits Conference Papers (ISSCC)*, pp.18-24, 7-11 Feb. 2010.

References of Chapter 2

- [2.1] Bruce Jacob, Spencer W. Ng, David T. Wang, "Memory Systems : Cache, DRAM, Disk", Morgan Kaufmann, 2007.
- [2.2] Domingo Benitez, Juan C. Moure, Dolores Rexachs, Emilio Luque, "A reconfigurable cache memory with heterogeneous banks", Design, Automation and Test in Europe Conference and Exhibition (DATE), 8-12, pp.825-830. March, 2010.
- [2.3] N. Hardavellas, I. Pandis, R. Johnson, N. Mancheril, A. Ailamaki, B. Falsafi, " Database Servers on Chip Multiprocessors: Limitations and Opportunities", in 3rd Conference on Innovative Data System Research, Asilomar, CA, USA, pp.79-87, 2007.
- [2.4] P. Ranganathan, S. Adve, N. P. Jouppi, "Reconfigurable Caches and their Application to Media Processing", in Proc. of the 27th Symposium on Computer Architecture, ACM Press, pp.214-224, 2000.
- [2.5] D.H. Albonesi, "Selective cache ways: on-demand cache resource allocation", in Proceedings of the 32nd Symposium on Microarchitecture, IEEE Computer Society, pp.248-259, 1999.
- [2.6] C. Zhang, F. Vahid, W. Najjar, "A Highly Configurable Cache Architecture for Embedded Systems", in Proc. 30th Symp. ISCA, 136-146, 2003.
- [2.7] 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 Proceedings of the Seventh IEEE Symposium on High-Performance Computer Architecture, Jan. 2001.
- [2.8] S. Yang, M. Powell, B. Falsafi, T.N. Vijaykumar, "Exploiting Choice in Resizable Cache Design to Optimize Deep-Submicron Processor Energy-Delay", Proc. 8th Symp. HPCA, IEEE Comp. Soc., pp.151-161, 2002.
- [2.9] R. Iyer, "CQoS: a framework for enabling QoS in shared caches of CMP platforms". In ICS '04: Proceedings of the 18th annual international

- conference on Supercomputing, ACM Press., pages 257–266, 2004.
- [2.10] K. Varadarajan, S.K. Nandy, V. Sharda, A. Bharadwaj, R. Iyer, S. Makineni, D. Newell, “Molecular Caches: A caching structure for dynamic creation of application-specific heterogeneous cache regions”, in Proc. 39th Symp. Microarchitecture, IEEE Computer Soc., pp.433-442, 2006.
- [2.11] D. Kaseridis, J. Stuecheli, L.K. John, “Bank-aware Dynamic Cache Partitioning for Multicore Architectures”, International Conference on Parallel Processing, 2009. ICPP '09, pp.18-25, 22-25 Sept. 2009.
- [2.12] JEDEC DDR3 Standard document, JESD79-3C, NOVEMBER 2008, <http://www.jedec.org/standards-documents>.
- [2.13] H. Kim, and I. C. Park, “High-performance and low-power memory-interface architecture for video processing applications,” in IEEE Trans. Circuits and Systems for Video Technology, vol. 11, no 11, pp.1160–1170, 2001.
- [2.14] 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, November, 2003.
- [2.15] C. Chang, M. Chang, and W. Hwang, “A Flexible Two-Layer External Memory Management for H.264/AVC Decoder”, IEEE International SOC Conference, Page(s):219 – 222, Sept. 2007.
- [2.16] J. Zhu, L. Hou, R. Wang, C. Huang, and J. Li, “High Performance Synchronous DRAMs Controller in H.264 HDTV decoder” in Proc. IEEE Int. Conf. Solid-state and Integrated Circuits Technol., vol. 3, pp. 1621-1624, 2004.
- [2.17] Hu Hongqi, Sun Jingnan, Xu Jiadong, ”High Efficiency Synchronous DRAM Controller for H.264 HDTV Encoder”, 4th IEEE Conference on Industrial Electronics and Applications, 2009. 25-27 Page(s):2132 – 2136, May 2009.
- [2.18] 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”, ISCAS pp. 273-276, August 2005.

- [2.19] Y. Li, Y. Qu, and Y. He, "Memory Cache Based Motion Compensation Architecture for HDTV H.264/AVC Decoder", ISCAS 2007, pp. 2906 - 2909, May 2007.
- [2.20] Tzu-Der Chuang, Lo-Mei Chang, Tsai-Wei Chiu, Yi-Hau Chen, and Liang-Gee Chen, "BANDWIDTH-EFFICIENT CACHE-BASED MOTION COMPENSATION ARCHITECTURE WITH DRAM-FRIENDLY DATA ACCESS CONTROL", ICASSP, pp. 2009 – 2012, 2009.
- [2.21] H.Y. Kang, K. A. Jeong, J. Y. Bae, Y. S. Lee, S. H. Lee, "MPEG4 AVC/H.264 decoder with scalable bus architecture and dual memory controller", ISCAS Circuits and Systems Volume 2, 23-26, pp. II-145-8, May 2004.
- [2.22] S. Heithecker, A Carmo Lucas, R. Ernst, "A mixed QoS SDRAM controller for FPGA-based high-end image processing", IEEE workshop signal processing System, pp. 322-327, 2003.
- [2.23] 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.
- [2.24] Hristo Nikolov, Todor Stefanov, Ed Deprettere, "Efficient External Memory Interface For Multi-Processor Platforms Realized On FPGA Chips", In Proc. International Conference on Field Programmable Logic and Applications, 2007, 27-29, pp. 580 – 584, Aug. 2007.
- [2.25] Juha-Pekka Soininen, Antti Pelkonen and Jussi Roivainen, "Configurable memory organisation for communication applications", Euromicro Symposium on Digital System Design, Proceedings, 4-6, pp. 86 – 93, Sept. 2002.
- [2.26] E. Ipek, O. Mutlu, , J.F. Martinez, R. Caruana, "Self-Optimizing Memory Controllers: A Reinforcement Learning Approach", 35th International Symposium on Computer Architecture, ISCA., pp. 39 – 50, June 2008.
- [2.27] H. Zheng, J. Lin, Z. Zhang and Z. Zhu, "Memory Access Scheduling

Schemes for Systems with Multi-Core Processors”, In Proc. of the 37th International Conference on Parallel Processing on IEEE computer society, pp. 406-413, Sept. 2008.

- [2.28] J. Shao and B. T. Davis, “A burst scheduling access reordering mechanism”, In HPCA '07: 13th International Symposium on High-Performance Computer Architecture, pp. 10-14, February 2007.
- [2.29] Jingtong Hu, Chun Jason Xue, Wei-Che Tseng, Meikang Qiu, Yingchao Zhao, and Edwin H.-M. Sha, “Minimizing Memory Access Schedule for Memories”, 15th International Conference on Parallel and Distributed Systems (ICPADS), pp. 104 – 111, 2009.
- [2.30] JEDEC Organization, website : <http://www.jedec.org/>
- [2.31] Micron, “Mobile SDRAM power-saving features”, Jun 2002, available on <http://www.micron.com/products/>
- [2.32] Infineon, “Infineon specialty DRAMs-Mobile-RAM” Feb. 2003, available on <http://www.infineon.com/cgi/>

References of Chapter 3

- [3.1] 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, S. Borkar, "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS", IEEE Journal of Solid-State Circuits, Vol. 43, No. 1, Jan 2008.
- [3.2] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, Liewei Bao, J. Brown, M. Mattina, Chyi-Chang Miao; C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, J. Zook, "TILE64 - Processor: A 64-Core SoC with Mesh Interconnect", IEEE International Solid-State Circuits Conference, 2008, ISSCC 2008, pp.88-598, 3-7 Feb. 2008.
- [3.3] S. Nomura, F. Tachibana, T. Fujita, Chen Kong Teh, H. Usui, F. Yamane, Y. Miyamoto, T. Yamashita, H. Hara, M. Hamada, Y. Tsuboi, "A low-power

- multi-core media co-processor for mobile application processors," *IEEE International Conference on IC Design and Technology, 2009. ICICDT '09.*, pp.129-134, 18-20 May 2009.
- [3.4] D. Kim, K. Kim, J.-Y. Kim, S. Lee, H.-J. Yoo, "Memory-centric network-on-chip for power efficient execution of task-level pipeline on a multi-core processor", *IET Computers & Digital Techniques*, Volume 3, Issue 5, pp. 513 – 524, September 2009.
- [3.5] Joo-Young Kim; Junyoung Park; Seungjin Lee; Minsu Kim; Jinwook Oh; Hoi-Jun Yoo, "A 118.4 GB/s Multi-Casting Network-on-Chip With Hierarchical Star-Ring Combined Topology for Real-Time Object Recognition," *IEEE Journal of Solid-State Circuits*, vol.45, no.7, pp.1399-1409, July 2010.
- [3.6] S. Vakili, S.M. Fakhraie, S. Mohammadi, "Evolvable multi-processor: A novel MPSoC architecture with evolvable task decomposition and scheduling", *Computers & Digital Techniques, IET*, vol.4, no.2, pp.143-156, March 2010.
- [3.7] Donghyun Kim; Kwanho Kim; Joo-Young Kim; Seungjin Lee; Hoi-Jun Yoo, "Implementation of Memory-Centric NoC for 81.6 GOPS object recognition processor", *IEEE Asian Solid-State Circuits Conference, 2007. ASSCC '07.*, pp.47-50, 12-14 Nov. 2007.
- [3.8] Yunxin Li, "Cognitive and Integrated Digital Home via Dynamic Media Access", *IEEE International Symposium on* , pp. 1 – 6, May 2009.
- [3.9] 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.
- [3.10] H. Saito, M. Nakajima, T. Okamoto, Y. Yamada, A. Ohuchi, N. Iguchi, T. Sakamoto, K. Yamaguchi, M. Mizuno, "A Chip-Stacked Memory for On-Chip SRAM-Rich SoCs and Processors," *IEEE Journal of Solid-State Circuits*, , vol.45, no.1, pp.15-22, Jan. 2010.

References of Chapter 4

- [4.1] A. Radulescu, J. Dielissen, K. Goossens, E. Rijpkema, P. Wielage, “An Efficient On-Chip Network Interface Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Configuration,” *In Proc. Design, automation and test in Europe(DATE)*, pp. 1–6, Mar. 2004.
- [4.2] Yong-Long Lai, Shyue-Wen Yang, Ming-Hwa Sheu, Yin-Tsung Hwang, Hui-Yu Tang, Pin-Zhang Huang, “A High-Speed Network Interface Design for Packet-Based NoC,” in *Proc. IEEE Int. Conf. Communication, Circuits and Systems*, pp. 2667–2671, 2006.
- [4.3] F. Clermidy, R. Lemaire, Y. Thonnart, P. Vivet, “A Communication and Configuration Controller for NoC based Reconfigurable Data Flow Architecture,” in *Proc. ACM/IEEE Int. Symp. Networks-on-Chip*, pp. 153–162, May. 2009.
- [4.4] N. Concer, L. Bononi, M. Soulie, R. Locatelli, L.P. Carloni, “The Connection-Then-Credit Flow Control Protocol for Heterogeneous Multicore System-on-Chip,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, Vol. 29, no. 6, pp. 869–882, Jun. 2010.
- [4.5] D. H. Albonesi, “Selective cache ways: on-demand cache resource allocation”, in *Proceedings of the 32nd Symposium on Microarchitecture*, IEEE Computer Society, pp.248-259, 1999.
- [4.6] P. Ranganathan, S. Adve, N. P. Jouppi, “Reconfigurable Caches and their Application to Media Processing”, in *Proc. of the 27th Symposium on Computer Architecture*, ACM Press, pp.214-224, 2000.
- [4.7] HP Labs : CACTI model, available on <http://www.hpl.hp.com/research/cacti/>
- [4.8] Micron Technology, Inc., Website : <http://www.micron.com/>
- [4.9] Micron System Power Calculators, available on http://www.micron.com/support/dram/power_calc.html.
- [4.10] Po-Chun 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
- [4.11] R. Iris Bahar, Dan Hammerstrom, Justin Harlow, William H. Joyner Jr., Clifford Lau, Diana Marculescu, Alex Orailoglu, Massoud Pedram, “Architectures for Silicon Nanoelectronics and Beyond”, *IEEE Comput.*, vol. 40, no. 1, pp. 25–33, Jan. 2007.
- [4.12] L. Benini and G. De Micheli, “Network on Chips: Technology and Tools”, Morgan Kaufmann, 2006
- [4.13] W. J. Dally and B. Towles, “Principles and Practices of Interconnection Networks”, Morgan Kaufmann, 2004.
- [4.14] 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, T. Mattson, “A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS,” in *Proc. IEEE Int. Solid-State Circuits Conf.*, pp. 108–110, Feb. 2010.

References of Chapter 5

- [5.1] 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.
- [5.2] 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, February 2007.
- [5.3] 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.

- [5.4] 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.
- [5.5] 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 *Proceedings of IEEE International Midwest Symposium on Circuit and Systems*, vol. 2, pp. 1199–1202, Aug. 2005.
- [5.6] 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*, vol. 5, pp. 97-100, March 2005.
- [5.7] 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, August 2008.
- [5.8] H.-C. Tseng, C.-R. Chang and Y.-L. Lin, "A Motion Compensator with Parallel Memory for H.264 Advance Video Coding," in *Proceedings of the 16th VLSI Design/CAD Symposium*, Aug. 2005.
- [5.9] 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 International Symposium on Design, Automation and Test*, pp. 247- 250, April 2006.
- [5.10] 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 *Proceedings of IEEE International Conference on Circuits and Systems*, pp. 256-259, May 2008.
- [5.11] C.-H. Li, C.-H. Chang, W.-H. Peng, W. Huang, and T. Chiang, "Design of Memory Sub-System in H.264/AVC Decoder," in *proceeding of IEEE International Conference on Consumer Electronics*, pp.1-2, January 2007.

- [5.12] Po-Chun 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.
- [5.13] 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*, Sept. 2007, vol.17, no.9, pp.1103-1120.
- [5.14] Micron Technology, Inc., <http://www.micron.com/>
- [5.15] C. Chang, M. Chang, and W. Hwang, "A Flexible Two-Layer External Memory Management for H.264/AVC Decoder", *SOC Conference, 2007 IEEE International* 26-29, Page(s):219 – 222, Sept. 2007.
- [5.16] J. Zhu, L. Hou, R. Wang, C. Huang, and J. Li, "High Performance Synchronous DRAMs Controller in H.264 HDTV decoder" in *Proc. IEEE Int. Conf. Solid-state and Integrated Circuits Technol.*, vol. 3, pp. 1621-1624, 2004.
- [5.17] Tzu-Der Chuang, Lo-Mei Chang, Tsai-Wei Chiu, Yi-Hau Chen, and Liang-Gee Chen, "BANDWIDTH-EFFICIENT CACHE-BASED MOTION COMPENSATION ARCHITECTURE WITH DRAM-FRIENDLY DATA ACCESS CONTROL", *ICASSP*, pp. 2009 – 2012, 2009.
- [5.18] M. Mrak, M. Grgic, S. Grgic, "Scalable video coding in network applications", *International Symposium on Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8*, vol., no., pp. 205- 211, 2002.
- [5.19] Chia-Ho Pan and I. Hsien Lee and Sheng-Chieh Huang and Chung-Jr Lian and Liang-Gee Chen, "A Quality-of-Experience Video Adaptor for Serving Scalable Video Applications", in *IEEE Transactions on Consumer Electronics*, vol. 53, number 3, pp. 1130-1137, 2007.
- [5.20] Tzu-Der Chuang and Pei-Kuei Tsung and Pin-Chih Lin and Lo-Mei Chang and Tsung-Chuan Ma and Yi-Hau Chen and Liang-Gee Chen, "A 59.5mW Scalable/Multi-view Video Decoder Chip for Quad/3D Full HDTV and Video

Streaming Applications", in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp.104-110, 8-12, 2010.

- [5.21] Tzu-Der Chuang and Pei-Kuei Tsung and Pin-Chih Lin and Lo-Mei Chang and Tsung-Chuan Ma and Yi-Hau Chen and Liang-Gee Chen, "Low Bandwidth Decoder Framework for H.264/AVC Scalable Extension", in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 273-276, 2010.

References of Chapter 6

- [6.1] Domingo Benitez, Juan C. Moure, Dolores Rexachs, Emilio Luque, "A reconfigurable cache memory with heterogeneous banks", Design, Automation and Test in Europe Conference and Exhibition (DATE), 8-12 March, pp.825-830, 2010.
- [6.2] K. Varadarajan, S.K. Nandy, V. Sharda, A. Bharadwaj, R. Iyer, S. Makineni, D. Newell, "Molecular Caches: A caching structure for dynamic creation of application-specific heterogeneous cache regions", in Proc. 39th Symp. Microarchitecture, IEEE Computer Soc., pp.433-442, 2006.
- [6.3] D. Kaseridis, J. Stuecheli, L.K. John, "Bank-aware Dynamic Cache Partitioning for Multicore Architectures", International Conference on Parallel Processing, 2009. ICPP '09, vol., no., pp.18-25, 22-25 Sept. 2009.
- [6.4] S. Yang, M. Powell, B. Falsafi, T.N. Vijaykumar, "Exploiting Choice in Resizable Cache Design to Optimize Deep-Submicron Processor Energy-Delay", Proc. 8th Symp. HPCA, IEEE Comp. Soc., pp.151-161, 2002.

張 雍 Yung Chang

PERSONAL INFORMATION

Birth Date: August. 01, 1986

Birth Place: Taipei, TAIWAN

E-Mail Address: derrick7722@gmail.com

EDUCATION

09/2008 – 07/2010 M.S. in Electronics Engineering, National Chiao Tung University
Thesis: On-Demand Memory System for Wireless Video
Entertainment System

09/2004 – 06/2008 B.S. in Department of engineering science, National Cheng Kung
University

PUBLICATIONS

Yung Chang, Po-Tsang Huang, Wei Hwang, “A Capacitive Boosted Buffer for Energy-Efficient and Variation-Tolerant Sub-Threshold Interconnect” 2009 Electronic Technology Symposium (ETS), June 19, 2009.

Po-Tsang Huang, **Yung Chang**, Shiang-Fei Wang and Wei Hwang, “An Efficient Network Interface for Memory-Centric On-Chip Interconnection Network”, IEEE Asia Pacific Conference on Circuits and Systems, IEEE APCCAS, 2010 (Submitted)

HONOR

2009 ETS Outstanding Oral Paper Award

PATENTS

Yung Chang, Po-Tsang Huang, and Wei Hwang, “Inter-layer Pre-fetch Scheme for Scalable Video Coding” US/TW Patent Pending (submitted)

Yung Chang, Po-Tsang Huang, and Wei Hwang, “Adaptive Memory resource allocation for Multi-task System” US/TW Patent Pending (submitted)