ISSN:2320-0790

An International Journal of Advanced Computer Technology

Hierarchical Multiport Memories: A Survey

Ravi Babu P[1], Srinivasa Rao K[2]
[1]Department of CIS, University of Hyderabad, Hyderabad, India.
Email: perakalapudi@gmail.com
[2]TRR College of Engineering, Medak, India.
Email: principaltrr@gmail.com

Abstract: This paper presents a detailed survey of the technological evolution of Hierarchical Multi-port memory Architectures (HMA). Computers with multiport memories provide better performance when compared to conventional memory systems. The major issues to design multiport memories are high level of integration, large number of ports, high operating speed, and area. HMA is one of the solutions proposed in the literature to design an efficient multiport memory system. This paper presents a detailed survey on HMA.

Keywords: Centralized Crossbar, CMOS, Distributed Crossbar, HMA, Memory Cell, MPM, and SRAM.

I. INTRODUCTION

For over a decade, there is a rapid growth in the performance and capability of computer systems. Today's commercial applications depends on parallel computing to process large sets of data such as web searching, financial modeling, embedded processors and medical imaging. The major complexity involved in parallel computers is, to build an efficient memory, which, in turn leads to efficient communication between processors and memory nodes [1]. The need to design communication and memory architectures of massively parallel hardware is necessary for the implementation of highly demanding applications. A major problem in designing efficient data transfer and storage architecture for data parallel applications is to provide architectural support to achieve better performance in terms of throughput [2]. The optimization criteria to design an efficient system, is power consumption, area occupancy and temporal Performance. This can be achieved if both memory and communication latency is minimized [2]. However, the circuit area dominates the processors influence.

The advent of parallel computers with shared and distributed memory facilitated efficient communication between processors for better throughput, when compared to conventional memory architectures. This has paved the way for design and implementation of multiport memory architectures more easily, and also serving as an alternative for conventional memory systems [3][4]

System-on-chip (SoC) devices are one of the solutions that provide multiple ports, embedded memories with large random access bandwidth to support data processing applications. Traditional multiprocessor designs, focus on an effective design for data processing modules without taking into account the memory and communication interfaces. However, the effectiveness of memory and communication interfaces plays a decisive role in enhancing the circuit area, thereby increasing the size and power of the chip. This paper discusses extensively about various memory architecture designs and the challenges in the context of memory architectures. The paper is organized by sections categorized on the basis of the evolutionary patterns over time in the field of memory architectures. Section II discusses the principles of Conventional Memory Architectures. Section III gives the architecture of Multiport Memories. Section IV introduces the HMA and its design methodology. Section V describes the various schemes of HMA model reported in literature and in Section VI, presents conclusion by highlighting the issues.

II. CONVENTIONAL MEMORY ARCHITECTURES

Several parallel architectures have been proposed in the literature [3], [5], [6]. However, these architectures deliver a throughput of few hundreds of Mbps due to limited processing parallelism. The massive parallelism demands the simultaneous data access capability. The exploitation of data parallelism in a computing unit demands data in parallel for processing. The simultaneous access to memory has been achieved by using Vector Processors is proposed in [7], [8], and Multibank or Multi-port Memories in [9]. In [7], authors introduced the simultaneous transmission of the data using multiple

interconnections, whereas in [8], author introduced various data pre fetching techniques to perform the execution of the tasks in parallel.
Conventional memory architectures [6], [10] have parallel random access capabilities by using Single Port Memories, Memory Interleaving, Shared and Distributed Memories this gives a considerable throughput. The detailed description of these architectures is given in the section II-A.

A. Single Port Memories
Single port memories implemented as shared memories in general multiprocessor systems. The time-shared bus communication media is connected to several processors as shown in Fig.1. The bandwidth is a major bottleneck in this architecture where a single memory access can be done through single port at the same time. For the massively parallel multiprocessors, the single port memory must be superfast to meet the massive parallelism range. The present current access times of single port memory chips are incompatible to meet the memory issues. Therefore, the data have to be organized in multiple multi banks or vector memories to satisfy the required memory bandwidth, while keeping the delay and area of the memory architectures substantially lower.

B. Memory Interleaving
Memory interleaving is one of the memory system implementation methods, where system memory is divided into several independent banks that are connected to processors through a crossbar switch [10] [6] as shown in Fig.2. A memory bank is a block of memory and several processors can access several banks simultaneously. The advantage of memory interleaving is that the subsequent vector elements can be placed in subsequent banks and accessed simultaneously. These memory systems fail to provide the performance of ideal shared memories, where, simultaneous access to same bank may lead to increase in latency.

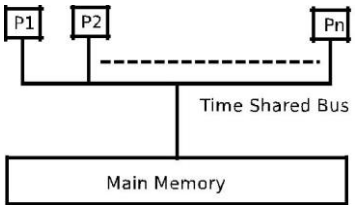

Fig. 1. Block Diagram of Conventional (Single Port) Memory

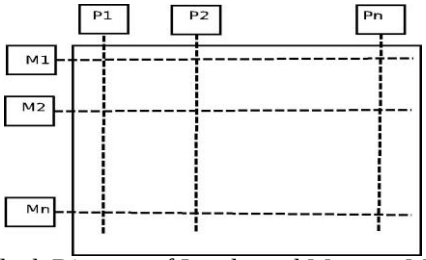

Fig. 2. Block Diagram of Interleaved Memory Multiprocessor system

C. Shared and Distributed Memories

A single main memory that has a symmetric relationship to all processors and a uniform access time from one processor to any other processor is called shared memory (symmetric) multiprocessor [4]. The most obvious way to implement shared memories is to use Multiport Random Access Memories (MRAM) as building blocks of true shared memory. Multiport RAM is a memory that has multiple ports that can be used to access memory cells simultaneously and independently of each other as shown in Fig.3.

The typical structure of Shared Memory consists of two address decoders, a memory cell array, and a bus interface. The memory chip is connected to other devices through address lines, data lines, read/write lines, and chip select lines. The memory cells are arranged in the form of a square array. To select appropriate memory cell, appropriate row and column select lines must be activated according to address information provided in the address lines by using address decoders. To transform a single port memory to multiport memory, ports are added by adding data/address/chip select lines for each port.

The structure of multiport cell is more complex than the single port shared memory. The relative cost effectiveness of multiport RAM is estimated by calculating wiring and component count complexity factors. Both multiport and single port chips have same count of memory cells. But in the multi-port case, the cells are more complex in terms of number of transistors in the cell. By using multi-port connections or by adding additional memory banks, a shared memory design can be scaled to a few dozen processors. To support larger processor counts, memory must be distributed among the processors rather than being centralized memory sys-tem. The basic architecture of distributed memory consists of individual nodes containing a processor with local memory, I/O and an interface to an Inter-connection Network that connects all the processors.

Distributing the memory among the nodes has major benefits. First, it is a cost effective way to scale the memory bandwidth, if most of the accesses are to the local memory in the node. Second, it reduces the latency for accessing the local memory. These two advantages make distributed memory attractive at smaller processor count as processors get even faster and require more memory bandwidth and low memory latency. The key disadvantage for distributed memory architecture is the communication of data between the processors which is more complex, and that it requires more effort in the software to take advantage of the increased memory bandwidth [4].

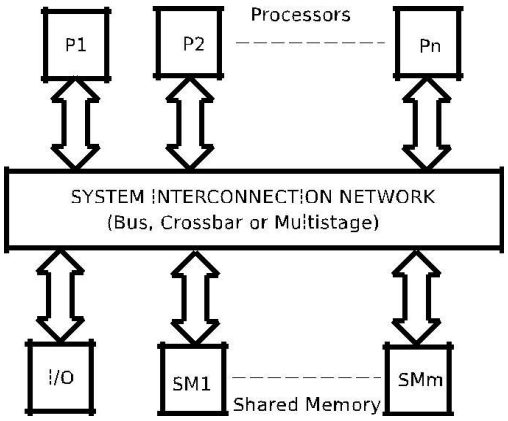

Fig. 3. Block Diagram of Multiprocessor system with shared memory

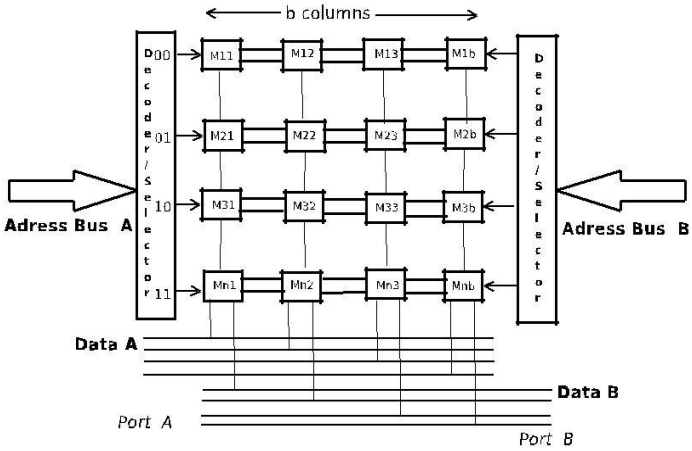

Fig. 4. Block Diagram of Multiport memory

III. MULTIPORT MEMORIES

A multiport memory system employs separate ports between each memory module and each CPU. The content of a multi-port memory can be accessed through different ports simultaneously, so that high access bandwidth and efficient use of memory is possible. Each port provides seperate independent access path for reading data from an array or writing data to an array. This array may be accessed in a random access manner through each port. Thus, a N-port multiport memory almost acts like N independent data arrays, except that, the contents of the arrays are always identical. Such a N-port memory can support N-way parallel access to the data array, allowing it to run N times faster than if the accesses were processed sequentially. Fig.4 illustrates a dual port memory, which is paradigm of multiport memory. The two input-output ports are labelled as Port A and Port B. Each port consists of an input n-bit address bus, a b-bit input/output external data bus and read/write signal. The memory is organized into b-columns for a memory of 2^n with b-bit words. Decoder/Selector selects the address lines. The advantage of the multiport memory organization is the high transfer rate that can be achieved because of the multiple paths between processors and the memory. Multiport memories offer a great way to interface between two buses, particularly between processors that need to communicate together. The complexity of multiport memories prevent the manufacturing of multiport memory technology, but even with small port count, computers with multiport memories can provide performance comparable to distributed memory systems with remarkably greater processor count [1] [3] [4] [11] [12].

Since the conventional N-Port register file which has N-Port memory cells with N word-lines and N bit-lines is generally used, the area of the register file becomes huge because the quantity of wiring space increases by the square of the number of ports. Therefore, design of a register file with many ports causes problems, such as, enlargement of chip size, deterioration of register access speed and high power consumption. This limits the performance of the processors [13].

The Hierarchical Multiport-Memory Architecture (HMA) plays a vital role in area reduction by using bank-based multi-port architectures with 1-port or 2-port banks. The memory-access time and power dissipation are also reduced substantially. It features parallel read/write access with low access conflict probability from all ports, although only 1-port memory cells are used [14]. In this paper, HMA design paradigms and issues related to handling access conflicts and various access scheduling algorithms reported in literature are discussed in detail.

IV. HIERARCHICAL MULTIPORT MEMORY ARCHITECTURE

Research has been carried out on Multi Port Memories (MPMs) for several issues/aspects. These memories have been used to enhance system performance in various domains, such as, Embedded Systems, Digital Signal Processing, Wafer Scale Integration, Multi-Chip module production and Optical Communication etc. When compared to conventional multi-port cell architecture, Ideal multi-port memories leads to a large penalty in silicon area, access time and power consumption due to the increase of port-related signal lines, which blow-up the size of bit-storage cell [15]. The major drawback of MPMs is requirement of expensive memory control logic and a large number of cables and connectors [16].

HMA is one of the solutions proposed in literature to overcome the complexities involved to build multiport memories. Current research may turn multiport memories into a more attractive alternate for conventional memory systems [1] [17].
HMA is a regular 2-dimensional arrangement using 1-port memory banks. Block diagram of Hierarchical multiport-

memory architecture with distributed crossbar switch is shown in Fig.5. The memory bank modules of second hierarchical level are exploited with memory banks, row and column bank selector, access conflict manager circuit and read/write circuit. Memory bank structures are arranged in rows and columns with distributed crossbar switch as the shared memory and activation of each bank takes place only on demand. Access-conflict manager circuit resolves the conflict of the port addresses. 1-port memory cells are grouped on hierarchy level-1 into blocks with conventional word/bit line decoder to resolve the port addresses. This structure leads to high access bandwidth, small area and low power consumption [12]. The design and developments in HMA technology to mitigate the drawbacks of multiport memories are discussed in Section V.

V. DEVELOPMENTS IN HMAS

H.J. Mattaush [16] introduced the concept of Hierarchical N-port memory architecture with parallel read/write access with a low access conflict probability from all ports, although only

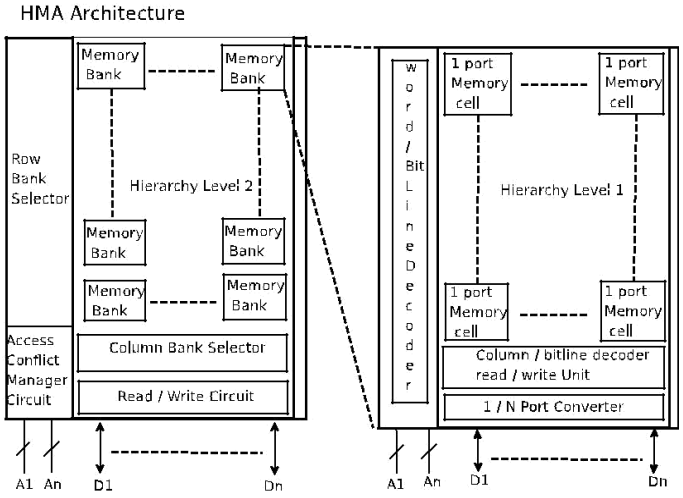

Fig. 5. Block Diagram of HMA

1-port memory cell is used. This design achieves accessing 'N' port numbers simultaneously with small area increase for additional ports, shorter access time and low conflict probability. The principle to achieve N-port access with 1-port memory cells is shown in Fig.5. The two levels of hierarchies are exploited where; 1-port memory cells are grouped on hierarchy level-1 into blocks with conventional word/bit line decoding with 2^m address bits. Transition from 1 to N ports is realized with an 'Active-Address-Select' circuit for switching the m1-bit address lines to access the required port i to the 1-port decoder. An 'Active-Port-Select' buffer switches the data to and from the accessed port i. These two circuits are of the multiplexer/de multiplexer type, respectively, and are realized in a conventional method. Hierarchy level 2 contains 2^m memory blocks, however, a given memory block is not accessible from more than one port simultaneously. An 'Access-Conflict-Resolve' circuit detects such conflict situations by comparing level-2 port-addresses A_{i2}, and resolves by ranking of port importance. The 'Row-select-signal and Column-Select-Signal Generators' on hierarchy level 2 generates activation signal RS_{i1} and CS_{i1} from port address parts A_{i2}. For each port i, these signals activate and control just one block of memory cells, with the 'Active-Address-Select' circuit" and the 'Active-Port-Select' buffer serving for correct switching of level 1 address parts A_{i1} and data D_{i1}.

TABLE-I

Ranking of architecture concepts wrt design criteria for multiport memories

Design	Switching Networks	N-port memory	Hierarchical Memory
Area	2	3	1
Access Time	3	1	1
Power Loss	2	3	1
Conflict Probability	3	1	2
Large N	2	3	1

All multiport memory architectures, which implement parallel access in real time, suffer from the problem of possible access conflicts (at least for write access).

The author introduced "Port-Importance-Hierarchy"(PIH) algorithm in [16], where the highest ranking port gets priority in a conflict situation and resolves the conflicts in hierarchical N-port memory architecture. In comparison with conventional implementation of all N-ports in each memory cell, approximately 28% (2 ports) of area reduction was achieved whereas 68% of area reduction was achieved for 16-ports, while access times are nearly equivalent as listed in Table I.

H.J. Mattausch proposed a two level hierarchy [18] for area efficient integrated N-port memory architecture, based on 1-port memory cells. The architecture is applicable to all types of dynamic, static and nonvolatile memory. This approach allows simultaneous read/write access from all ports, with access rejection probability adjustable to application needs. Johguchi K et al., in [15] presented a 16-port SRAM design, which realized the highest published random access bandwidth for SRAMs by using a multi-stage sensing scheme and a distributed crossbar memory architecture on 2-port with a 2 Kbit bank-based SRAM cells. The 16-port SRAM is designed in 90 nm logic CMOS technology with 6 metal layers, and occupies a silicon area of only 0.91 nm^2 per bit. Access conflicts are handled using access-scheduling algorithm. Experimental results show that, the power dissipation is 5.3 times larger (value 220 mW) at 500 MHz for conventional design, substantially higher clock frequency and larger storage capacity. Area per bit, power consumption and maximum clock frequency are about a factor of 16, 5 and 2 respectively, compared to 16-port SRAM designs.

The work reported in [15] is extended by introducing a

multistage sensing scheme developed in [19] with 2-stage pipeline and 2-port SRAM cells. This method achieves stability in read and writes access operations at the same time, with high-access reliability and high-access speed when compared with the results reported in [18] and also bit-area reduction is achieved by a factor of 16.5.

Weixing Ji et al. in [20] proposed multiport memory design methodology based on block read/write operations. Multiport memory architectures exploring interleaving technology are proposed and applied for multi processor system based on single port memory banks. Independent ports are connected to banks dynamically via switching network. The arbitrator in front of each bank determines the port to be accessed to the corresponding bank at each cycle. Such memory architectures allow tradeoff between bandwidth and area, but, the increase in number of banks increases low Port Access Rejection Probability (PARP. The proposed methodology, Block Based Multi-port Memory keeps low PARP, but few banks are needed to build the multiport memory system. The authors in [21] introduced a novel hierarchical multi-port cache, which implements the Hierarchical Multi-port memory Architecture based on single port banks. This type of cache has the advantage of high access bandwidth, low power dissipation and small area. A test chip design of a 4-port HMA cache with 0.18 μm^2 CMOS technology has been made and compared the performance with conventional level-2 cache. Memory system performance is measured by Average Memory Access Time (AMAT), that is, the average latency per memory reference.

For conventional single-port cache, AMAT can be expressed as:

$$AMAT = hittime + missrate \times misspenalty$$

$$AMAT = T_{H1\_cache} + CMR \times 2 \times T_{H2\_cache}$$

$$T_{H2\_cache} = T_{H2\_cache\_array} + Linesize/Bandwidth$$

For HMA cache, AMAT is computed as:

$$AMAT = \frac{1}{N(1+BCR)} T_{H1\_cache} + CMR \times 2 \times T_{H2\_cache}$$

Where CMR is Cache Miss Rate and BCR is Bank Conflict Rate.

For a 32 bank 4-port HMA, AMAT achieved 27.3% decrease for HMA cache and also achieved reduction in power dissipation when compared with the conventional 1-port cache. Experimentation using HSPICE was carried out for hierarchical 4-port cache design and observed that, the bank write time is about 0.91 ns and read time is about 0.7 ns. The power consumption is only 22.6 mW at 250 MHz, which is quite small when compared with the conventional multi-port memory.

A multiport RAM generator with the novel memory cell has been developed by Koji Nii et al., in [22] and fabricated on 0.5μm^2 CMOS Sea-of-Gates. Also, the fourfold read bit line technique has been adopted to minimize the access time without increase in area. The authors demonstrated using SPICE circuit simulation that these new techniques are effective in comparison with the conventional technique. The experimental results using SPICE simulation show that, each RAM operates at over 1.4 V and that the typical address access time of the 3-port RAM (16b x 256w) is 4.8 ns at 3.3V.

Fast and compact central arbiter circuits are proposed for detection and regulation of access conflicts in memories with multiple ports by N. Omori and H.J. Mattausch [23]. A layout study in 0.5μm^2, 2 metal CMOS technology verifies that the area overhead and access time penalty are small up to 32 ports. The proposed central arbiter concept allows the arbitration delay time to be largely removed from the critical path by carrying out arbitration and address decoding in parallel. Arbitration results can be fed into the critical path after the decoder at the cost of just one additional gate. The objectives for the arbiter circuitry are simplicity, minimum sequential gate number in critical path and minimum latency in terms of access cycles.

Sueyoshi T et al., [24] designed a compact, high speed, and low power bank type 12-port register file test chip of 0.35μ m^2 in CMOS technology for achieving highly parallel processing. In this full custom test chip design, 72% smaller area, 25% shorter access cycle time, and 62% lower power consumption are achieved in comparison to the conventional 12-port cell based register file. A new method called Simultaneous Multi-Threading (SMT) processing, divides the processors work load into largely independent program parts called threads. By executing these threads in parallel, SMT processors achieve a further increase in the total number of instructions, which are executed in parallel. The conventional multiport cell based architecture when used for such a register file, the problems of unacceptable increase in area, access time and power consumption arise. The major difference between HMA and the conventional crossbar architecture is, HMA realizes the crossbar function in distributed form with a 1-to-N port converter attached to each bank, which is not in the conventional centralized form. In this way, the necessary number of transistors and global wirings can be reduced without degrading the functionality. In order to solve access conflicts to the banks, an access conflict management circuit is included on the level-2 hierarchy of HMA. Access conflicts to the banks for each port are detected, and permission/prohibition of these accesses is decided. The bank type register file verified and also applicable to superscalar processors without loss in processor performance, when applying an appropriate access scheduling methodology.

Fukae Set al., [25] proposed distributed crossbar technique in bank-based multiport memories. By distributing the cross-points of the crossbar into each bank, the sub-division of the critical access path will overcome the two subsequent global signal nets at the centralized crossbar simultaneously. This is

led to access time reduction for bank based multiport memories with distributed crossbar technique. Also, with the additional introduction of a column/row bank selector concept, a smaller transistor number was achieved. This in turn lead to a smaller layout area for the distributed crossbar. Indeed a design comparison in 0.5µm CMOS with N = 4, M = 32, W = 1 and Bandwidth = 1024 resulted in 13% smaller area and 14% shorter access time for the HMA memory with distributed crossbar function when compared with the centralized bank based multiport memories.

K. Johguchi et, al., [26] presented a proposal to improve the low access bandwidth of conventional 1-port caches by utilizing a multi-bank structure with distributed crossbar to increase port number at small additional area cost. Using the proposed HMA cache, instruction and data cache can be combined without loss in access bandwidth, and also resulting in a lower cache miss rate at the same storage capacity. On the other hand, using a bank based multi port cache the access conflict rate increased. The simulation was carried out with a SimpleScalar tool, and when tested with benchmark programs, namely, Dhrystone and SPECint95 (GCC, IJPEG, etc.).

In [27], authors presented a 4-port unified data/instruction cache with HMA structure in 200nm CMOS technology. To minimize bank conflicts, an addressing method is proposed which ensures that the words in one cache line and also the consecutive cache lines are located in different banks, so that the access performance close to an ideal multiport cache can be realized. Adopting an efficient floor plan without routing restricted the area overhead, including the upper hierarchy level circuits and the conflict manager for the 4-ports, to only 25% in comparison to a 1-port cache. A minimum clock cycle time of 3.4ns has been achieved with a dynamic CMOS circuit technology and by overlapping the external bank access with the bank internal pre-charge.

Hybrid Hierarchical Multi-port Memory Architecture(HHMA) constructed by sharing multi port memory was proposed by Caixia Liu et al., in [28] and achieved high parallelism of multi port memory assuring processor nodes to access memory at high parallel degree. Experimental results show that, the nodes parallel degree when accessing shared memory in HHMA was higher than 90%. The performance of HHMA is evaluated based on Nodes Access Parallel Degree (NAPD).

$$NAPD = \frac{N}{(L+delay)}$$

Where L is data length, measured as the number of words and whereas delay is measured as, the number of delayed cycles for submission.

NAPD apparently reflects the ratio of the time used for transferring data to total access time. The higher the ratio is, the shorter the waiting time and higher the access parallelism. Experiments showed that HHMA is better than Distributed Shared Memory (DSM) in supporting highly efficient parallel communication. Parallel communication performance of HHMA in average is higher than that of DSM by 40%.

Zhaomin Zhu et al., [14] had proposed a low power bank-based multi-port SRAM design method with bank standby mode, a novel hidden pre-charge-time access method incorporated for no loss of speed. For HMA SRAMs, at every cycle time, most banks are not in use and deactivated, which means that there is no reading or writing operation in this bank. Three aspects are proposed to realize the bank-standby mode. First aspect is to employ bank selector signal concept to transfer the unused banks into a standby status to achieve much less power dissipation for these inactive banks. If the bank is inactive, then bank-selector signal is set to be zero. The second aspect for standby-mode realization is, to set the bank-internal clock signal as zero when the bank is not in use. For the conventional structure, the bank-internal clock signal is just taken from the external clock signal. Now, the input clock signal CLK-IN is controlled by Bank selector signal and global clock signal. External clock rising edge signal occurs slower than the bank-selector signal by accessing the inactive banks and it does not hazard the speed of normal operation in the active banks. The third aspect of standby mode realization, is to design a dynamic pseudo-NOR type word-line decoder controlled by CLK-IN signal. When the bank is inactive, CLK-IN is set to zero, causing all word lines to be connected to ground. This circuit has the advantage of low power consumption when compared to a static pseudo-NOR can type word-line decoder in [28] and it achieve faster speed than NAND-type decoder by reducing the PMOS transistor capacitance. The proposed structure has much better performance in terms of power dissipation when compared to conventional architecture. The power dissipation was 18.8% and 56% respectively for bank numbers 4 and 64. The feasibility of the new architecture is verified by the fabrication of a test chip in 0.18µm CMOS technology. More than 56% power dissipation decrease can be achieved when 64 banks are used while the chip size is the same.

Bank-based unified data/instruction cache with multiple ports was proposed by Koh Johguchi et al., [29]. To minimize bank conflicts, an addressing method ensures that the words in one cache-line and consecutive cache-lines are located in different banks to resolve the conflict issues. Using the proposed HMA cache, instruction and data cache can be unified without loss in access bandwidth, but with the advantage of a lower miss rate at the same storage capacity. On the other hand, using a bank-based multi-port cache access to one bank is restricted to 1-port which in turn increases the access-conflict rate. The simulation is carried out with a Simple Scalar tool for Dhrystone and SPEC95 (gcc, ijpeg, etc.,) benchmark programs. A test chip design of a 4-port bank-based cache in 0.18µm CMOS technology showed, that the area-overhead for the 4 ports is about 25%. A minimum clock cycle time of 3.8ns was achieved with a dynamic CMOS circuit technology and by overlapping the external bank access with the bank-internal pre-charge. The miss rate of unified cache with data

and instruction cache is shown in Fig.6 and Fig.7.

Johguchi et al., [30] proposed a method of bank-based unified data/instruction cache with a hidden pre-charge pipeline. A 2-stage synchronous cache-access scheme is applied, which overlaps the pre-charge phase of one stage with the other stages access phase. Thus the clock cycle time and the access time of the unified data/instruction cache become equal. A unified data/instruction cache can provide lower miss rate, because dynamic allocation of the effective storage capacity for data and instructions becomes possible, but 2-ports are needed to obtain the same access bandwidth possible with split caches. The disadvantage of increased access-conflict rate for a bank-based multi-port cache is mitigated, because locality conflicts are avoided by interleaved cache lines and cache-line words. Furthermore, the performance decrease due to access conflicts which can be resolved in 1 clock cycle, is much smaller than the performance decrease from cache misses. Simulation results reveal that the miss rate of the unified cache is reduced by 25% storage capacity as shown in Fig.8.

The authors proposed to exploit the hierarchical structure of the multi-bank cache for a 2-stage synchronous access scheme, which hides the pre-charge by overlapping the access phase of one stage with the pre-charge phase of the other stage. Bank-conflict management, bank selection, port conversion and bank-internal word line decoding constitute the first stage of the access path and are carried out simultaneously with the bank pre-charge when the clock is set to zero. When the clock changes from $0 \rightarrow 1$, the second stage of the access path, starting with the word line-driver activation is executed, while the first stage of the access path is pre-charged. Dynamic CMOS technology is used for the design of the cache circuits in order to reduce in particular the capacitive loads of the global routing to the banks. By adopting an efficient floor plan without routing-only areas restricts the area-overhead including second level circuits and the conflict manager for the 4-ports to only 25% in comparison to a 1-port cache. A minimum clock cycle time of 3.4ns could be achieved with a dynamic CMOS circuit technology and by overlapping the external bank access with the bank-internal pre-charge. The proposed bank-based multi-port cache is also very attractive for low power dissipation because of the maximum number of active banks. Determining the power dissipation is decided by the access-band width. Therefore, at most 10 data/instruction banks and 4 tag banks are simultaneously active for the test chip. The number of simultaneously active banks is independent of the total number of banks in the unified cache. The comparison between the splitted and unified cache for both data and instruction caches is shown in Fig 8.

Y. Mukuda et al., [31], the authors proposed a processor structure through the design of a multi-bank register file in 180nm CMOS technology. Register access queues were used as a part of the scheduler implemented in the multi-bank register file. The register access scheduler avoids the access conflicts by introducing a register access queue for each bank to store the bank accesses. The register file that implements the queues with 2-port banks enables to read, and to write at the same time. The register access queue consists of read queues and writes queues in each bank. The designed 12-port multi-bank register file with register-access queues has an area of $0.5mm^2$ and operates at up to 540 MHz, whereas, the conventional Multi-port-cell register file has an area of $1.43mm^2$ and operates at up to 330 MHz only. The proposed architecture achieves 65% smaller area and 64% higher operation frequency than a conventional multi-port-cell register file.

Tetsuo Hironaka et al., [32] introduced a multi-bank register file with high speed, low power consumption and area. Synthesizable Verilog-HDL has been developed with a fully custom designed superscalar processor with the ability to utilize a multi-bank register file. This was done to achieve higher clock rate without having large performance drop in the cycle based performance.

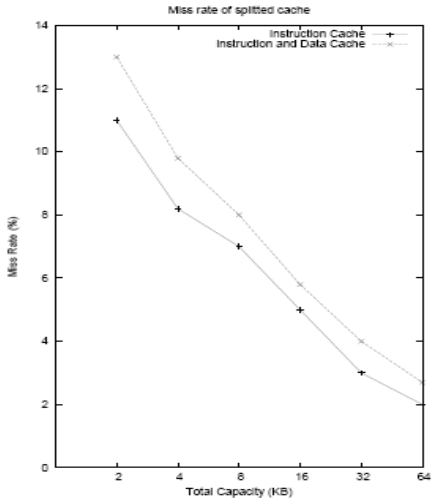

Fig. 6. Miss rate comparison of Splitted Cache Instruction Vs combined Instruction and Data

The authors introduced three register access methods to avoid the bank conflicts in the access of the multi-bank register file. These are Out-of-order register access method, reduction of the number of register accesses and access reduction by forwarding the register. If the register accesses are blocked by bank conflicts for some cycles, the processor has to stall the decode and rename the subsequent instructions until the bank conflict is resolved. Even if the register bank is able to accept register access for the subsequent instructions the access is stalled. To remove this problem, they proposed a method for register access per operand, not register access per instructions by decomposing the instruction to operation and register access operands. The out-of-order register access by the decomposed register access operands is realized by adding a register access queue on each register bank, which buffers the bank accesses. All register access operands are queued in

either one of the access queue in order, and delayed until the bank gets accessible. The cycle based performance evaluation was achieved by implementing a software simulator, whereas, the gate level performance evaluation was achieved by designing the full processor with synthesizable Verilog-HDL description. From the cycle based performance evaluation, the register file architecture has only 2% performance drop in harmonic mean with 4-bank register file, each with a single-port bank, compared to the superscalar processor with the ideal multi-port register file. Gate level performance evaluation shows that the functions added to support the multi-bank register file increase the number of Gates needed to implement the register rename unit and the reservation unit by 30%. But, on the other hand, it reduces the size of the register file to 30% and improved the register access speed by 49%. This means superscalar processor can achieve two times better performance than conventional method.

Tomohiro Inoue et al., [33], authors introduced a novel concept of bank-based multiport memory architecture using a blocking network instead of a crossbar network. The blocking network achieves high access bandwidth with minimum hardware resources than the conventional approach. With this approach, the number of transistors of the bank-based multiport memory with a blocking network is 50% less than the crossbar method and 30% smaller than the HMA for 512 ports and 512 banks. Furthermore, as compared with the conventional bank-based multiport memory, the proposed method decreases the random access bandwidth to 5% or less and achieves nearly the same performance.

In [33], authors proposed EBMA (EBSF-based Multiport Memory Architecture) wherein the EBSF (Expanded Banyan Switching Fabrics) blocking net-work is used instead of the crossbar system. EBSF is one of the blocking networks which can reduce access conflicts in the network, because it expands and multiplies the Banyan network. The EBSF consists of MK/2 cross points per stage (vertical direction), and $\log_2 M$ stages. The EBSF cross point consists of a control circuit and two switches. The two switches consist of an address switch and a data switch. The address switch is a one-direction switch which transmits memory addresses from each port to one of the banks. The data switch is a bidirectional switch which transmits data between each port and one of the banks. The number of transistors used in the interconnection network in EBMA is smaller than those in the crossbar and HMA systems. The transistor ratio for EBMA is only 5% while those of the cross bar and HMA systems are 50% and 32%, respectively, for 512 ports and 512 banks. The ratio of the number of transistors used in the data switch in crossbar and HMA systems are about 70% to 80% of all memory, while the number of transistors of the data switch in EBMA is about 60% of the total.

In [34] [35], a novel intelligent Multi-Port Memory design has been proposed. with the help of port-priority and read/write-priority, this multi-port memory can resolve both access conflicts and Extend Port Importance Hierarchy (EPIH) algorithm is proposed for basic conflict handling, while Block Access Control (BAC) algorithm is proposed to reduce conflicts when processors carry block read/write operations. Experimental results on Xilinx Virtex-II shows that, when compared to implementation of N-ports in each cell, this design saves 88% LUT resources and also shows that, as port number N increases, the cell cost increases significantly.

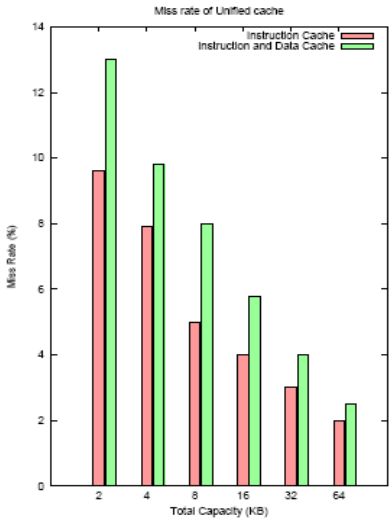

Fig. 7. Miss rate comparison of Unified Cache Instruction Vs combined Instruction and Data

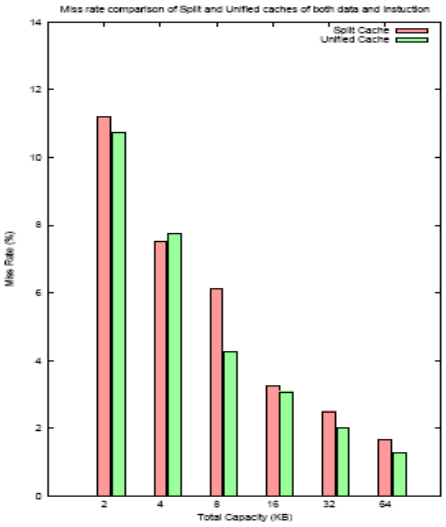

Fig. 8. Miss rate comparison of Splitted and Unified Cache

VI. CONCLUSION

In this paper, we have discussed the various conventional memory architectures. Multiport Memory concept is explained as basis of HMA model. The basic structure of HMA has been discussed with the developments. The reported

works in literature related to HMA indicate that high bandwidth, area efficiency, low power dissipation and low access conflicts have been achieved. The same technologies can be upgraded with present day CMOS technologies like 16nm, 18nm, 22nm, and 26nm to achieve better area efficiency, low power and high bandwidth. (2) The Reconfigurable architecture, System-on-Chip (SoC) and Intellectual Property (IP) technologies can be used to get better performance and compatibility in memory based systems.

REFERENCES

[1] M. Forsell, "Are multiport memories physically feasible?" ACM SIGARCH Computer Architecture News, vol. 22, no. 4, pp. 47– 54, 1994.

[2] R. Corvino, A. Gamatie,´ and P. Boulet, "Architecture explo-ration for efficient data transfer and storage in data-parallel ap-plications," Proceedings of the Euro-Par conference on Parallel Processing, pp. 101–116, 2010.

[3] D. Culler, J. Singh, and A. Gupta, Parallel computer archi-tecture: a hardware/software approach. Morgan Kaufmann, 1999.

[4] J. Hennessy and D. Patterson, Computer Architecture: A Quan-titative Approach. Morgan Kaufmann Pub, 2011.

[5] B. J.L., "A survey of some theoretical aspects of multiprocess-ing," Computing Surveys, vol. 5, no. 1, pp. 31–80, 1973.

[6] P. Enslow, Multiprocessors and Parallel Processing. Wiley-Interscience, 1974.

[7] A. G.A. and J. E.D., "Computer interconnection structures: Taxonomy, characteristics, and examples," Computing Surveys, vol. 7, no. 4, pp. 197–213, 1975.

[8] S. D. Haynes L.S., Lau R.L. and M. D.W., "A survey of highly parallel computing," Computer, vol. 15, no. 1, pp. 9–24, 1982.

[9] A. Wilson Jr, "Hierarchical cache/bus architecture for shared memory multiprocessors," pp. 244–252, 1987.

[10] G. Burnett and E. Coffman Jr, "A study of interleaved memory systems," in Proceedings of Computer Conference. ACM, 1970, pp. 467–474.

[11] M. Mano et al., Computer system architecture. Prentice-Hall, 1993.

[12] L. Glasser and R. Rivest, "A fast multiport memory based on single-port memory cells," 1991.

[13] Y. Tatsumi and H. Mattausch, "Fast quadratic increase of multiport-storage-cell area with port number," Electronics Let-ters, vol. 35, no. 25, pp. 2185–2187, 1999.

[14] Z. Zhu, K. Johguchi, H. Mattausch, T. Koide, and T. Hironaka, "Low power bank-based multi-port SRAM design due to bank standby mode," in proceedings of 47th Midwest Symposium on Circuits and Systems, vol. 1. IEEE, 2004, pp. I–569.

[15] K. Johguchi, Y. Mukuda, K. Aoyama, H. Mattausch, and T. Koide, "A 2-stage-pipelined 16 port SRAM with 590Gbps random access bandwidth and large noise margin," IEICE Electronics Express, vol. 4, no. 2, pp. 21–25, 2007.

[16] H. Mattausch, "Hierarchical n-port memory architecture based on 1-port memory cells," in Proceedings of the 23rd European Solid-State Circuits Conference. IEEE, 1997, pp. 348–351.

[17] K. Johguchi, K. Aoyama, T. Sueyoshi, H. Mattausch, T. Koide, M. Maeda, T. Hironaka, and K. Tanigawa, "Multi-Bank register file for increased performance of highly-parallel processors," in Proceedings of the 32nd European conference on Solid-State Circuits Conference. IEEE, 2006, pp. 154–157.

[18] H. Mattausch, "Hierarchical architecture for area-efficient in-tegrated n-port memories with latency-free multi-gigabit per second access bandwidth," Electronics Letters, vol. 35, no. 17, pp. 1441–1443, 1999.

[19] K. Johguchi, Y. Mukuda, S. Izumi, H. Mattausch, and T. Koide, "A 0.6-Tbps, 16-port SRAM design with 2-stage-pipeline and multi-stage-sensing scheme," in Proceedings of International Conference on Solid State Circuits. IEEE, 2007, pp. 320–323.

[20] W. Ji, F. Shi, B. Qiao, and H. Song, "Multi-port memory design methodology based on block read and write," in Proceedings of International Conference on Control and Automation. IEEE, 2007, pp. 256–259.

[21] Z. Zhu, K. Johguchi, H. Mattausch, T. Koide, T. Hirakawa, and T. Hironaka, "A novel hierarchical multi-port cache," in Pro-ceedings of the 29th European Solid-State Circuits Conference. IEEE, 2003, pp. 405–408.

[22] N. Kojii, M. Hideshi, O. Toduya, I. Shuuhei, K. Shinpei, and S. Hiroshi, "A novel Memory Cell for Multiport RAM on 0.5 um CMOS Sea-of-Gates," in Proceedings of International conference on Solid-State Circuits, vol.

30, no. 3. IEEE, 1995,

pp. 316–320.

[23] N. Omori and H. Mattausch, "Compact central arbitrers for Memories with multiple read/write ports," Electronics Letters, vol. 37, no. 13, pp. 811–813, 2001.

[24] S. Tetsuya, U. Hiroshi, H. Mattausch, K. Tetsushi, M. Yosuke, and H. Tetsuo, "Compact 12-Port Multi-Bank Register File Test-Chip in 0.35 um CMOS for Highly Parallel Processors," in
Proceedings of Design Automation Conference. IEEE, 2004,

pp. 551–552.

[25] S. Fukae, T. Inoue, H. Mattausch, T. Koide, and T. Hironaka, "Distributed against centralized crossbar function for realizing bank-based multiport memories," Electronics Letters, vol. 40, no. 2, pp. 101–103, 2004.

[26] K. Johguchi, Z. Zhu, T. Hirakawa, T. Koide, T. Hironaka, and H. Mattausch, "Distributed crossbar architecture for area-efficient combined data/instruction caches with multiple ports," IET Electronics Letters, vol. 40, no. 3, pp. 160–162, 2004.

[27] K. Johguchi, H. Mattausch, T. Koide, and T. Hironaka, "4-Port Unified Data/Instruction Cache Design with Distributed Cross-bar and Interleaved Cache-Line Words," IEICE transactions on Electronics, vol. 90, no. 11, pp. 2157–2160, 2007.

[28] C. Liu, J. Li, H. Zhang, and Q. Zuo, "HHMA: A Hierarchical Hybrid Memory Architecture Sharing Multi-Port Memory," in proceedings of 9th International Conference for Young Com-puter Scientists,. IEEE, 2008, pp. 1320–1325.

[29] K. Johguchi, Z. Zhu, H. Mattausch, T. Koide, and T. Hironaka, "Unified Data/Instruction Cache with Bank-based Multi-Port architecture," in Proceedings of of Asia Pacific Conference on Circuits and Systems. IEEE, 2006, pp. 1–3.

[30] K. Johguchi, Z. Zhu, H. Mattausch, T. Koide, T. Hironaka, and K. Tanigawa, "Unified Data/Instruction Cache with Hierarchical Multi-port Architecture and Hidden precharge pipeline," in Pro-ceedings of Asia Pacific Conference on Circuits and Systems. IEEE, 2006, pp. 1297–1300.

[31] Y. Mukuda, K. Aoyama, K. Johguchi, H. Mattausch, and T. Koide, "Access Queues for Multi-Bank Register Files En-abling Enhanced Performance of Highly Parallel Processors," in IEEE Region 10 Conference. IEEE, 2006, pp. 1–4.

[32] T. Hironaka, M. Maeda, K. Tanigawa, T. Sueyoshi, K. Aoyama, T. Koide, H. Mattausch, and T. Saito, "Superscalar Processor with multi-bank register file," in proceddings of IEEE confer-ence on Innovative Architecture for Future Generation High-Performance Processors and Systems. IEEE, 2005, pp. 1–10.

[33] T. Inoue, T. Hironaka, T. Sasaki, S. Fukae, T. Koide, and H. Mattausch, "Evaluation of bank-based multiport memory architecture with blocking network," Electronics and Commu-nications in Japan (Part III: Fundamental Electronic Science), vol. 89, no. 6, pp. 22–33, 2006.

[34] W. Zuo, Z. Qi, and L. Jiaxing, "An intelligent multi-port mem-ory," in proceedings of International Symposium on Intelligent Information Technology Application Workshop. IEEE, 2008, pp. 251–254. W. Zuo, "An intelligent multi-port memory," Journal of Computers, vol. 5, no. 3, pp. 471–478, 2010.