EXPLORING ON-CHIP MEMORY ARCHITECTURES: DESIGN AND PERFORMANCE ANALYSIS FOR EMBEDDED SYSTEMS

Dr. J. Ravi Shankar, M.Tech., Ph.D.[1], Dr. T. Meera Kumari, M.Tech., Ph.D.[2] & Dr. K. Rajendra, M.Tech., Ph.D.[3]

Professor, Department of Electronics & Communication Engineering, St. PETER'S ENGINEERING COLLEGE, Kompally, Hyderabad, India

Professor, Department of Electronics & Communication Engineering, Visvesvaraya College of Engineering & Technolohy, Patelguda Hyderabad, India

ABSTRACT

Today's feature-rich multimedia products require embedded system solution with complex System-on-Chip (SoC) to meet market expectations of high performance at low cost and lower energy consumption. SoCs are complex designs with multiple embedded processors, memory subsystems, and application specific peripherals. The memory architecture of embedded SoCs strongly influences the area, power and performance of the entire system. Further, the memory subsystem constitutes a major part (typically up to 70%) of the silicon area for the current day SoC.

The on-chip memory organization of embedded processors varies widely from one SoC to another, depending on the application and market segment for which the SoC is deployed. There is a wide variety of choices available for the embedded designers, starting from simple on-chip SPRAM based architecture to more complex cache-SPRAM based hybrid architecture. The performance of a memory architecture also depends on how the data variables of the application are placed in the memory. There are multiple data layouts for each memory architecture that are efficient from a power and performance viewpoint. Further, the designer would be interested in multiple optimal design points to address various market segments. Hence a memory architecture exploration for an embedded system involves evaluating a large design space in the order of 100,000 of design points and each design points having several tens of thousands of data layouts. Due to its large impact on system performance parameters, the memory architecture is often hand-crafted by experienced designers exploring a very small subset of this design space. The vast memory design space prohibits any possibility for a manual analysis.

In this work, we propose an automated framework for on-chip memory architecture exploration. Our proposed framework integrates memory architecture exploration and data layout to search the design space efficiently. While the memory exploration selects specific memory architectures, the data layout efficiently maps the given application on to the memory architecture under consideration and thus helps in evaluating the memory architecture. The proposed memory exploration framework works at both logical and physical memory architecture level. Our work addresses on-chip memory architecture for DSP processors that is organized as multiple memory banks, with each back can be a single/dual port banks and with non-uniform bank sizes. Further, our work also address memory architecture exploration for on-chip memory architectures that is SPRAM and cache based. Our proposed method is based on multi-objective Genetic Algorithm based and outputs several hundred Pareto-optimal design solutions that are interesting from a area, power and performance viewpoints within a few hours of running on a standard desktop configuration.

KEYWORDS: SoC, On-Chip Memory Organizer, DSP Proessor and multiple memory banks

I. INTRODUCTION

Today's Embedded Systems and VLSI technology allows us to integrate tens of processor cores on the same chip along with embedded memories, application specific circuits, and interconnect infrastruc- ture. As a result, it is possible to integrate an entire system onto a single chip. The single chip phone, which has been introduced by several semiconductor vendors, is an example of such a system-on-chip; it includes the modem, radio transceiver, power management func- tionality, a multimedia engine and security features, all on the same chip. An embedded system is an application-specific system which is optimized to perform a single function or a small set of functions [70]. We distinguish this from a general-purpose system, which is software-programmable to perform multiple functions. A personal computer is an ex- ample of a general-purpose system; depending on the software we run on the computer, it can be useful for playing games, word processing, database operations, scientific com- putation, etc. On the other hand, a digital camera is an example of an embedded system, which can perform a limited set of functions such as taking pictures, organizing them, or transferring them to another device through a suitable I/O interface. Other examples of embedded systems include mobile phones, audio/video players, videogame consoles, set- top boxes, car infotainment systems, personal digital assistants, telephone central-office switches, dedicated network routers and bridges. Note that a large number of embedded

systems are built for the consumer market. As a result, in order to be competetive, the cost of an embedded system cannot be very high. Yet, the consumers demand higher per- formance and more features from the embedded systems products. It is easy to appreciate this point if we compare the performance and feature set offered by mobile phones that cost Rs 5000/-(or 100$) today and which cost the same a few years ago. We also see that a large number of embedded systems are being built for the mobile market. This trend is not surprising - the number of mobile phone subscribers increased from 500 Million in year 2000 to 2.6 Billion in 2007 [7]. Because of such high volumes, embedded systems are extremely cost sensitive and their design demands careful silicon-area optimization. Since mobile devices use batteries as the main source of power, embedded systems must also be optimized for energy dissipation. Power, which represents the rate at which energy is con- sumed, must also be kept low to avoid heating and improving reliability. In summary, the designer of an embedded system must simultaneously consider and optimize price, perfor- mance, energy, and power dissipation. Application specific embedded systems designed today demand innovative methods to optimize these system cost functions

Memory Subsystem

On-chip Memory Organization

The memory architecture of an embedded processor core is complex and is custom de- signed to improve run-time performance and power consumption. In this section we describe only on the memory architecture of the DSP processor as this is the focus of the thesis. This is because, the memory architecture of the DSP is more complex than that of microcontrollers (MCU) due to the following reasons: (a) DSP applications are more data dominated than the control-dominated software executed on an MCU. Mem- ory bandwidth requirements for DSP applications range from 2 to 3 memory accesses per.

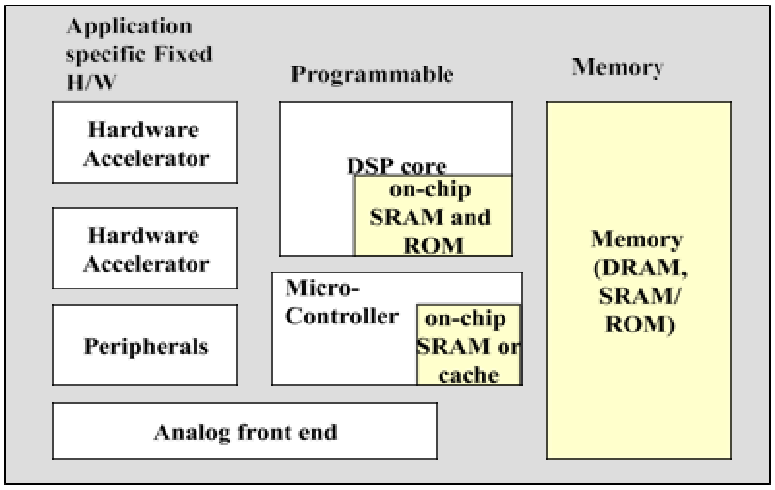


Figure 1.1: Architecture of an Embedded SoC

processor clock cycle. For an MCU, this figure is, at best, one memory access per cycle.
(b) It is critical in DSP application to extract maximum performance from the memory subsystem in order to meet the real-time constraints of the embedded application. As a consequence, the DSP software for critical kernels is developed mostly as hand optimized assembly code. In contrast, the software for MCU is typically developed in high-level languages. The memory architecture for a DSP is unique since the DSP has multiple on-chip buses and multiple address generation units to service higher bandwidth needs. The on-chip memory of embedded processors can include (a) only Level-1 cache (L1-cache) (e.g., [1]), (b) only scratch-pad RAM (SPRAM) (e.g., [75, 76], or (c) a combination of L1-cache and SPRAM

Data Layout

To efficiently use the on-chip memory, critical data variables of the application need to be identified and mapped to the on-chip RAM. The memory architecture may contain both on-chip cache and SPRAM. In such a case it is important to partition the data section and assign them appropriately to on-chip cache and SPRAM such that memory performance of the application is optimized. Further, among the data sections assigned to on-chip cache and SPRAM, a proper placement of the data sections on the cache and SPRAM is required to ensure that the cache misses are reduced and the multiple memory banks of the SPRAM and the dual ported SPRAMs are efficiently utilized. Identifying such a data placement for data sections, referred to as the data layout problem, is complex and critical step [10, 53]. This task is typically performed manually as the compiler cannot assume that the code under compilation represents the entire system.

Memory Architecture Exploration
In modern embedded systems, the area and power consumed by the memory subsystem is up to 10 times that of the data path, making memory a critical component of the design [11]. Further, the memory subsystem constitutes a large part (typically up to 70%) of the silicon area for the current day SoC and it is expected to go up to 94% in 2014 as shown in the Figure 1.3 [6]. The main reason for this is that embedded memory has a relatively smallsubsystem per-area design cost in terms of both man-power, time-to- market and power consumption [60]. Hence the memory plays an important role in the design of embedded SoCs. Further the memory architecture strongly influences the cost, performance and power dissipation of an embedded SoC.

As discussed earlier, the on-chip memory organization of embedded processors varies widely from one SoC to another, depending on the application and market segment for which the SoC is deployed. There is a wide variety of choices available for the embed- ded designers, starting from simple on-chip SPRAM based architecture to more complex cache-SPRAM based hybrid architecture. To begin with, the system designer needs to decide if the SoC requires cache and what is the right size of on-chip RAM. Once the high level memory organization is decided, the finer parameters need to be defined to complete the memory architecture definition. For the on-chip SPRAM based architecture, the pa- rameters, namely, size, latency, number of memory banks, number of read/write ports per memory bank and connectivity, collectively define the memory organization and strongly influence the performance, cost, and power consumption. For cache based on-chip RAM

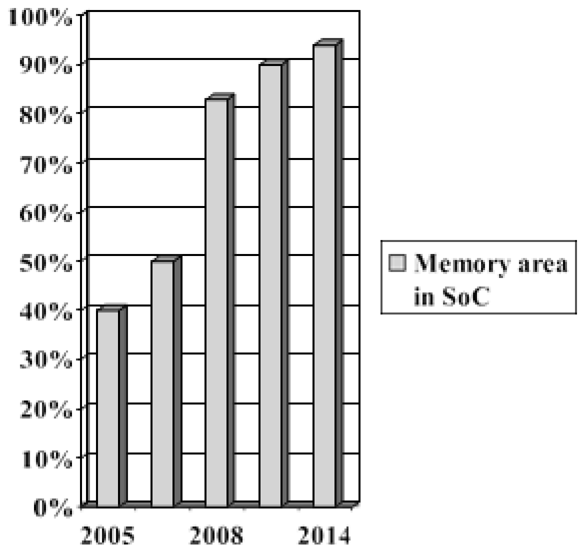
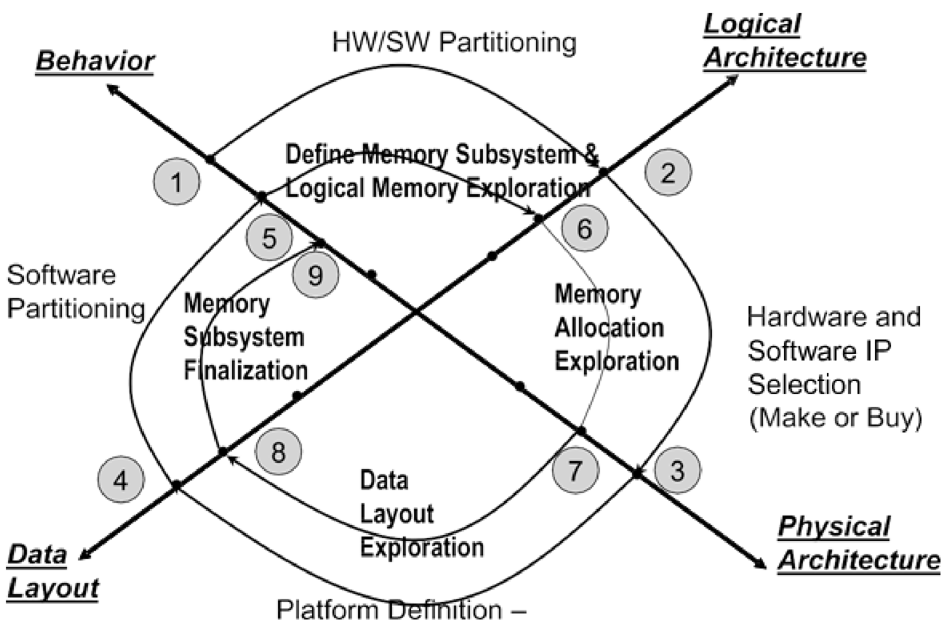


Figure 1.3: Memory Trends in SoC

the finer parameters are the size of cache, associativity, line size, miss latency and write policy. Due to its large impact on system performance parameters, the memory architec- ture is often hand-crafted by the designer based on the targeted applications. However, with the combination of on-chip SPRAM and cache, the memory design space is too large for a manual analysis [31]. Also, with the projected growth in the complexity of embed- ded systems and the vast design space in memory architecture, hand optimization of the memory architecture will soon become impossible. This warrants an automated frame- work which can explore the memory architecture design space and identify interesting design points that are optimal from a performance, power consumption and VLSI area (and hence cost) perspective. As the memory architecture design space itself is vast, a brute force design space exploration tool may take large computation time and hence is unlikely to be useful in meeting the tight time-to-market constraint. Further, for each given memory architecture, there are several possible data section layouts which are opti- mal in terms of performance and power. This further compounds the memory architecture exploration problem.


Behavior
HW/SW Partitioning
Logical Architecture
Define Memory Subsystem & Logical Memory Exploration
1
2
5
6
9
Software Partitioning
Memory Subsystem Finalization
Memory Allocation Exploration
Hardware and Software IP Selection (Make or Buy)
4
8
7
3
Data Layout Exploration
Data Layout
Physical Architecture
Platform Definition –

II. ON-CHIP MEMORY ARCHITECTURE OF EMBEDDED PROCESSORSDSP

On-chip SPRAM Architecture

DSP processor based embedded systems have an *on-chip* memory which typically has a single cycle access time [49]. The on-chip memory, also referred to as *scratch pad memory*, is mapped into an address space disjoint from the off-chip memory but connected to the same address and data buses. [1]Typically the scratch-pad memory is organized into multiple memory banks to facilitate multiple simultaneous data accesses. DSP Processors typically have 2 or more address generation units and multiple on-chip buses to facilitate multiple memory accesses.

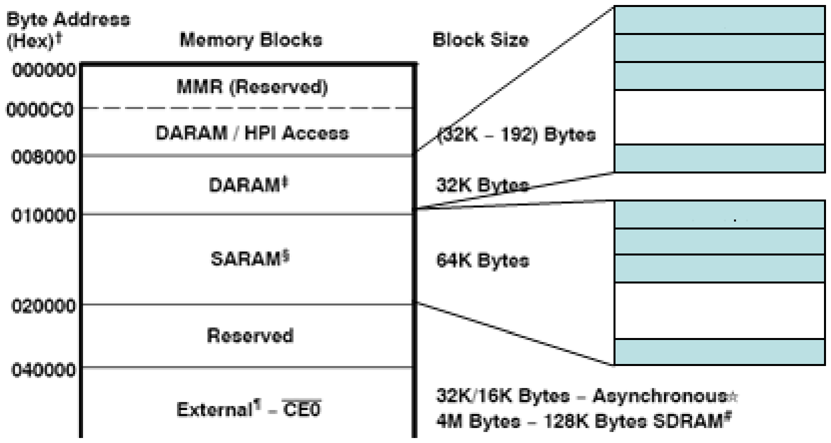


Figure 2.1: Example DSP Memory Map

Further, each on-chip memory bank can be organized either as a single-access RAM (SARAM) or as a dual-access RAM (DARAM), to provide single or dual accesses to the same memory bank in a single cycle. For example, Texas Instruments TMS320C54X digital signal processor has two data read buses and one data write bus [75]. and, Texas In- struments TMS320C55X processor has three data read busses and two data write busses, since concurrent access to the same array are common in DSP applications [76]. Fig- ure 2.1 presents memory map of C55X DSPs, where multiple memory banks of SARAM and DARAM memory banks form a part of memory map, and MMR represents mem- ory mapped registers which typically contain control registers, status registers and stack pointers. The

DARAM and SARAM regions can be recognized using multiple memory bank to enable two concurrent accesses

III. BIBLIOGRAPHY

1. *ARM920T and ARM922T: ARM9 Family of Embedded Processors.* http://www.arm.com/products/CPUs/families/ARM9Family.html.
2. *ARM926EJ-S and ARM926E-S: ARM9E Family of Embedded Processors* http://www.arm.com/products/CPUs/families/ARM9Family.html.
3. *lp solve*.http://lpsolve.sourceforge.net/5.5/.
4. *SystemC – Language for System-Level Modeling, Design and Verification.* http://www.systemc.org/home.
5. *Verilog Hardware Description Language.* http://www.verilog.com/index.html.
6. *International Technology Roadmap for Semiconductors*, SEMATECH, 3101, Indus- trial Terrace Suite, 106 Austin TX 78758., 2001.
7. 2007 global mobile communications - statistics, trends and forecasts Technical report, http://www.reportbuyer.com/telecoms/mobile/2007 global mobile trends.html, 2007.
8. 1st IEEE Inter. Symposium on Industrial Embedded Systems. *Panel Discussion. Open Issues in SoC Design.*, http://www.iestcfa.org/panel discussions.htm, 2006

IV. REFERENCES

[1] P. Mishra, P. Grun, N. Dutt, and A. Nicolau. Processor-memory co-exploration driven by a memory-aware architecture description language. In Proceedings of the International Conference on VLSI Design, 2001.

[2] B. Monien and R. Diekmann. A local graph partitioning heuristic meeting bisection bounds. In 8th SIAM Conference on Parallel Processing for Scientific Computing, 1997.

[3] H. Orsila, T. Kangas, E. Salminen, T. D. Hamalainen, and M. Hannikainen. Au- tomated memory-aware application distribution for multi-processor system-on-chips. Journal of System Architectcure: the EUROMICRO Journal, 53(11), November 2007.

[4] R. Oshana. DSP Software Development Techniques for Embedded and Real-Time Systems. Embedded computer systems, 2006.

[5] K. V. Palem, R. M. Rabbah, V. J. Mooney III, P. Korkmaz, and K. Puttaswamy. Design space optimization of embedded memory systems via data remapping. ACM Conference on Languages, Compilers and Tools for Embedded Systems (LCTES), June 2002.

[6] G. Palermo, C. Silvano, and V. Zaccaria. Multi-objective design space exploration of embedded systems. Journal of Embedded Computing, 1(3), August 2005.

[7] M. Palesi and T. Givargis. Multi-objective design space exploration using genetic algorithms. In International Workshop on Hardware/Software Codesign (CODES), May 2002