Memory Subsystem Architecture Design for Multimedia Applications

Alexsandro C. Bonatto*† and Altamiro A. Susin†
*Restinga Campus, Federal Institute of Rio Grande do Sul, Porto Alegre, Brazil
†Electrical Engineering Department, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
Email: {alexsandro.bonatto; altamiro.susin}@ufrgs.br

***Abstract*—Multimedia applications for processing high resolution video, data and audio sequences are known to require a high speed and high density memory port. Several hardware modules accessing the same main memory simultaneously generate concurrent accesses and memory conflicts, which reduce the memory port bandwidth and increase data latency. This paper proposes to integrate the SoC modules using an intelligent memory controller, in a memory-centric design approach. Also, it presents a memory system design analysis for a multimedia SoC with an analytical model for latency reduction in a multi-level memory hierarchy.**

Keywords*—Memory system; Memory controller; DRAM; Multimedia applications; Hardware description languages.

I. Introduction

Memory performance is a limiting factor of computer system efficiency and is the bottleneck in current multimedia processing systems. Design engineers must accurately evaluate the tradeoff between performance, power consumption, cost and reliability in the memory system design. Properties like memory locality, technology, features, data width, among others, have to be evaluated. These design considerations refer to multi-variable equations that are used to model the memory system performance characteristics, conducting the design to levels of different types and sizes of memories. This is called the system memory hierarchy, structured in levels of local and external memories.

The optimal memory hierarchy design for a system depends on the specific latency and bandwidth requirements, system power and cost, and processes workload characteristics. Based on these characteristics, local and external memories are designed. In the case of System-on-Chip (SoC) architectures for multimedia, a single interface to an external Dynamic Random Access Memory (DRAM) is used and the memory system design challenge is to fit bandwidth requirements to this single port memory. Memory system model and design are widely explored in recent works [1-4] and show the state-of-the-art in the area of memory systems design.

This work aims to design and implement a new SoC communication infrastructure that is able to manage all module communication by the main memory. System integration is planned in a memory-centric approach, in which memory communication interfaces are tailored to the application requirements. The SoC processing modules are classified according latency tolerance and deadline requirements to design specific client interfaces to these modules. The memory system is designed to guarantee Quality of Service (QoS) for the modules. The architectural design is based in the analysis of an analytical model that combines bandwidth, area and latency variables to achieve the best memory subsystem implementation. The case study is a digital television SoC implemented in an FPGA platform using VHDL and some simulated results are compared with results obtained from an analytical model.

II. Memory Subsystem Architecture Analysis

In a multimedia SoC, most data transfers passes through the main memory. Because multimedia processing is predominantly sequential, a memory hierarchy structured in memory levels is preferred than buses or network-on-chip. Therefore, memory-centric design is the key-point to reach QoS by separating processes that need real-time data access to the external memory from processes with less critical time constraints for the memory accesses. Hierarchical interconnection networks are proposed to reduce data transfer delays in memory systems with several concurrent accesses.

In this work we propose to separate the interconnection network in small groups of processes, classified according its QoS characteristics as: real-time (RT), latency sensitive (LS) or bandwidth sensitive (BS). Real-time processes require the shortest path to access the external memory. To exemplify, a CPU running control operations has no real-time requirements

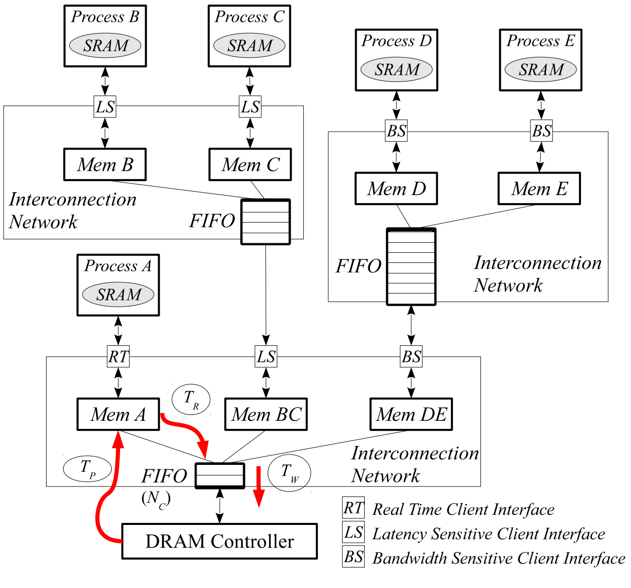
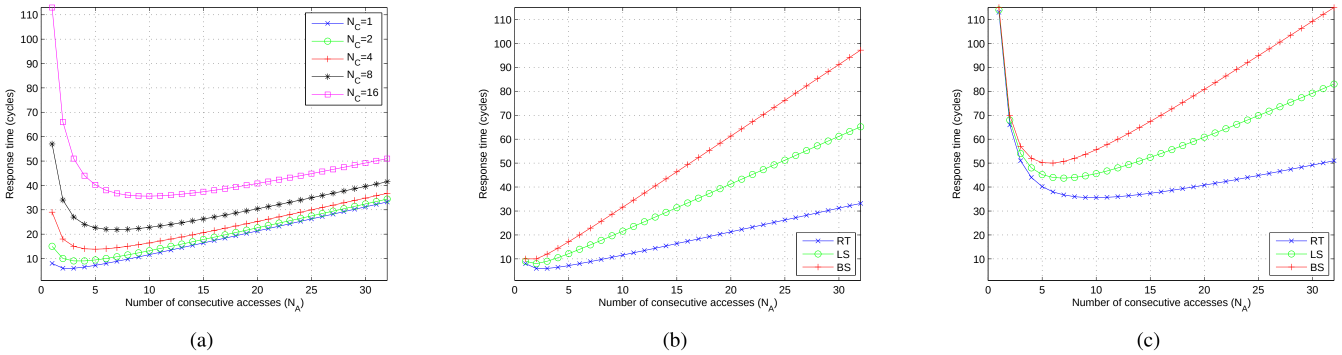


Fig. 1. Proposed memory system for SoC integration using hierarchical interconnect networks and processes classified by QoS requirements.



(a)

(b)

(c)

Fig. 2. Response time analysis (T_D) for read data in the external memory for different scenarios: (a) presents a comparison between access granularity (N_A) and buffer size (N_C) for the RT client interface; (b) and (c) show a comparison between RT, LS and BS client interaces for 1 and 16 positions buffer size.

but if this CPU runs a video composition algorithm, real-time access to external memory is required. In contrast to CPU, the video processing modules can support small amounts of latency with the use of input and output buffer memories, because of their sequential behavior to process data pixels.

The proposed memory system hierarchy is structured into interconnection networks and client interfaces to multiplex the memory port, as is presented in Fig. 1. Processes can access the external memory through an intelligent interconnection network which manages memory accesses. Buffer memories are used to hold temporarily commands before access the memory port, improving memory bandwidth by the use of bursts transfers. Each interconnection network contains an arbiter that manages command requests and data flow. Processes are classified according its workload characteristics and are connected to the appropriate client interface (RT, LS or BS).

An analytical model is used to estimate the worst-case latency across levels in the memory hierarchy. In this analysis the worst case is achieved for several modules accessing the memory at same time. Therefore, the command queue is full and the current command is the last one in the command FIFO. The QoS for different workloads in the memory controller is adjusted by buffers and FIFO sizes and accesses granularities, which impacts on latency and bandwidth.

The proposed memory system model allows quantifying latency for an RT client by calculating latency to access the memory port. The response time $T_D(c)$ for a client c is proportional to the number of clock cycles to pass commands through the memory hierarchy. $T_D(c)$ is formed by three parts as is shown in (1) and (2) [5]: request cycles $T_R(c)$, wait cycles $T_W(c)$ and processing cycles T_P, which are related to the system memory levels inside each interconnection network (Fig. 1). N_A is the number of consecutive accesses in the external DRAM, N_C is the command buffer size placed in the input of the DRAM port and N_{RC} is the number of clock cycles to make a row/bank change.

$$T_D(c) = T_R(c) + T_W(c) + T_P \tag{1}$$

$$T_D(c) = \sum_{i=0}^{c} N_A(i) + N_C + N_{RC} \cdot \frac{N_C}{N_A(c)} + T_P \tag{2}$$

Results obtained from the analytical model for worst case delays are presented in Fig. 2. There is a tradeoff between the granularity of accesses and the command buffer size. Memory system variables N_A and N_C can be adjusted to limit the worst-case for data latency in client interfaces. The time spent by the DRAM controller to process stored commands in the input queue is reduced with the increase of the granularity (N_A). However, the increase of N_A causes a reduction of the clients switching activity, thus improving the memory system performance. Also, larger command buffer size increases the time that a buffered command waits until being processed by the memory controller. Some simulation results obtained using the FPGA model were compared with results presented in Fig. 2 but the entire simulation is still under analysis.

III. Conclusions

This work presents a memory-centric design strategy to interconnect clients with different data access requirements to the memory port guaranteeing quality of service. The proposed memory system is architected as an interconnection network which presents a predictable behavior, designed to target latency reduction maintaining the throughput. Clients are classified according their access granularities and buffer sizes in the memory levels. The proposed memory system is suitable for systems that handle data structures stored in external memories.

References

[1] P. van der Wolf and J. Geuzebroek, “Soc infrastructures for predictable system integration,” in Design, Automation Test in Europe Conference Exhibition (DATE), 2011, pp. 1–6.

[2] M. D. Gomony, C. Weis, B. Akesson, N. Wehn and K. Goossens, “DRAM Selection and Configuration for Real-Time Mobile Systems,” in: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012 , pp.51–56.

[3] B. Akesson, K. Goossens, “Architectures and modeling of predictable memory controllers for improved system integration,” in: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011, pp.1–6.

[4] M. K. Jeong; E. M. Sudanthi, N. C. Paver, “A QoS-aware memory controller for dynamically balancing GPU and CPU bandwidth use in an MPSoC,” in Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE, 2012, pp.850–855.

[5] A. C. Bonatto, M. Negreiros, A. B. Soares, A. A. Susin, “Towards an Efficient Memory Architecture for Video Decoding Systems,” Computing System Engineering (SBESC), 2012 Brazilian Symposium on, 2012, pp.198–203.