International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Design and Simulation of a Multiport Memory Controller for Communication between Master and Slave Port

Tejaswini A Mahajan[1], Mahesh Neelagar[2], T C Thanuja[3]

[1]PG student in Department of VLSI design and Embedded systems, Center for PG Studies, Visvesvaraya Technological University, Belagavi, Karnataka, India

[2]Assistant Professor in Department of VLSI design and Embedded systems, Center for PG Studies, Visvesvaraya Technological University, Belagavi, Karnataka, India

[3]Head of Department of VLSI design and Embedded systems, Center for PG Studies, Visvesvaraya Technological University, Belagavi, Karnataka, India

Abstract: ***The dynamic memory controller plays an important role in system-on-a-chip (SoC) designs to provide enough memory bandwidth through external memory for DSP and multi-media processing. As the multimedia applications are growing rapidly past a decade. The applications of multi-media for processing high resolution video, data and audio sequences are known to require a high speed and high-density memory port. The memory is required for data storage in real time applications, the memory controllers support DDR3/DDR2/DDR/SDRAM memories and it can be configured according to their requirements. In spite much research on performance improvement, the external memory performance is lagging. Hence the memory controller is essential. The proposed architecture of multiport memory controller is designed for flexible communication between the master and the slave ports and also the communication speed is increased as the design contains a number of buffers for, and also embedded memory for configuration storage and an arbiter including round robin scheduling scheme for scheduling the read/write accesses. The design technique provides flexible systems and independent from other system architecture. The design is modelled in Altera and the read/write simulation results are acquired in Modelsim 6.6a using an external DDR3 SDRAM memory.***

Keywords: Altera Quartus II, Buffers, DDR3 SDRAM, Flexible communication, Modelsim 6.6a.

1. Introduction

As the more, time-to-market, high cost and complexity of the frameworks are expanding the framework on chip is turning into an outline incline for inserted frameworks, subsequently the gadgets having high limit and minimal effort are utilized as the off-chip memory frameworks which are satisfied by DRAM gadgets. As the expanding execution crevice between the preparing centers and DRAM memory, the execution bottleneck happen in SoC outlines due to off-chip memory frameworks. The memory controller controls the correspondence to and from the PC's primary memory. It is once in a while likewise called a memory chip controller (MCC) or a memory controller unit (MCU). These advances have been broadly connected in different applications, including PC designs, vast scale image grouping, 3D signal handling, design acknowledgment, and picture and video content investigation [1]. Accordingly, mixed media applications are getting more convoluted step by step, which requires the equipment framework to give higher processing power and also higher information transmission capacity.

As the multimedia applications are being developed rapidly the efficacy of memories and the processors are becoming the key to success. As concluded from the literature survey the existing memory controllers offer low speed communication between the master and the slave and also the supported slave modules are 16 MOD's. Hence the proposed multiport memory controller is required. The proficiency of processors and recollections, in this way, has turned into the way to achievement. With the significant advances in execution of processor, the significance of the memory controller has likewise expanded. John L [2]. Hennessy et al demonstrates that the framework bottleneck for the most part originates from the constraint of outer memory by measuring the distinction in time between processor memory ask (for a solitary processor or center) and the inertness of a DRAM access from 1980 to 2010. Besides, with the advancement of multicore processor, the aggregate pinnacle transmission capacity basically develops as the quantities of center develops. This was exemplified by the present-day top of the line four-center 3.2-GHz-clock-rate processor Intel Core i7 which can create two information memory references for every center each clock cycle, i.e., a pinnacle of 25.6 billion 64-bit information memory references by second, while the pinnacle transfer speed to primary memory is just 6% of this. Advancing memory get to time, in this manner, has been one of the greatest difficulties frameworks, particularly those require rapid or multi-question handling, regularly have significant issues of memory transmission capacity. This is appeared by the examination [3], where an advanced flag processor-based (DSP-based) hand signal acknowledgment framework could just accomplish continuous necessity at low determination because of the constraint of memory transmission capacity, despite the fact that a capable DSP is utilized. The issue may turn out to be more genuine on the convenient gadgets, where the equipment asset is unquestionably confined. Elite off-chip memory correspondence, in this manner, could be viewed as a key to

effective framework outline. In the paper, we present a multiport memory controller that provides the communication flexibility between the master i.e. the external memory and the slave modules with increased speed and supporting up to 32 slave modules.

2. Proposed Block Diagram

The proposed technique involves three main modules as follows:
1. A-MCU
2. Multiport Front End

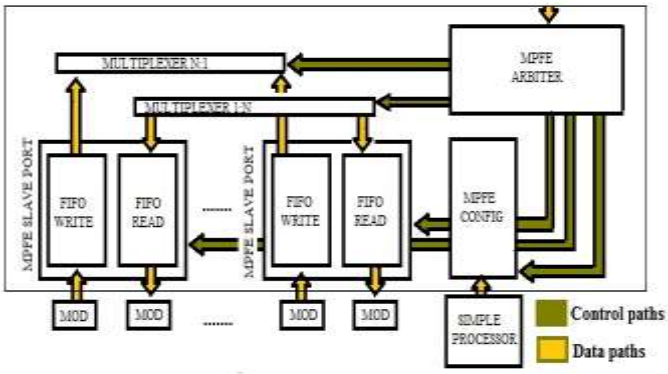


Fig 1. Proposed block diagram for MPMC

The proposed block diagram of the system is as shown in Fig.1. Initially the configuration parameters of an MPFE are loaded from an external module such as a simple processor and are stored in the MPFE CONFIG. After completing, the MPFE ARBITER checks all parameters and starts the communication between the MPFE and other MODs. Subsequently, MODs access off-chip memory by sending the request signals to the MPFE SLAVE PORT. The MPFE ARBITER checks over the status registers continuously and grants the permission to others components correspondently. The multiplexers, at the same time, take charge of the data paths to and from the A-MCU. The system consists of MPFE Slave port (FIFO Read/Write), MPFE Arbiter, MPFE Configuration, Multiplexer.

A. MPFE Slave port

The MPFE SLAVE PORT is exploited as data buffer so as to increase transfer performance. In fact, each MPFE SLAVE PORT contains two distinct dual-clock dual width first-in first-out modules (DCDWFFs) for both read and write processes. The DCDWFFs ensure that two separate components operating in different clock domains can communicate with each other properly. The divergence of MOD clock and internal MPFE clock is a case in point. Besides, these DCDWFFs supports different port widths for input and output data, which is advantageous to various applications. The DCDWFF full, almost-full, empty, and almost-empty signals indicate the availability of a certain MPFE SLAVE PORT. For instance, unless full signal is de-asserted, an MOD cannot write any data to the given port. The burst length, used by MODs to indicate the number of transfers in each burst, and the port categories have an effect on the depth of DCDWFF. This is shown by the following example, if an MPFE SLAVE PORT is time-critical category with burst length of 32, the DCDWFF depth should be a multiple of 32 in order to keep temporary data as much as possible. In a word, depend on the requirements, the MPFE SLAVE PORT clock, width, and depth are adjusted appropriately; this is definitely considered as an advantage over the A-MPFE.

B. MPFE Configuration Port

This port stores whole configuration parameters of the MPFE including number of used ports(NUP), burst length for write and read process (wBL, rBL), starting address (SA) and transaction length (TL). It is noticed that ending addresses for write and read process (wEA, rEA) of one transaction are derived from SA, wBL, rBL, and TL. The MPFE CONFIG possesses two operating modes, configuration mode (CFG) and active mode (ACT). As soon as the iCFG enable signal is asserted, the MPFE CONFIG switches to CFG, receives configuration parameters from external module and then stores them on internal memories. As can be seen in the figure, there are five single-clock dual-port memory components (SCDPMs) which keep the information of NUP, rBL and wBL, SA, and rEA and wEA. Two high-order and five low-order bits of an iCFG address register select the SCDPM and SCDPM address, respectively figure demonstrates the structure of iCFG data register corresponding with two high-order bits of iCFG address, iCFG address [6:5]. The first row is the arrangement of wBL, rBL, and NUP during iCFG address [6:5] =002. Similarly, the second and third row are the organization of SA and TL during iCFG address [6:5] =012 and 102, respectively. The wEA and rEA are started computing as soon as the first TL is obtained. During CFG, the MPFE CONFIG also sends SA to MPFE ARBITER for several initializations.

C. MPFE Arbiter Port

The access to external memory is shared by the multiport front-end design between different accesses that are present on the slave ports. The following functions are performed by the arbiter in MPFE design.

1) Dividing the traffic in two classes—critical and non-critical—for protecting the time-critical accesses.
2) Using user specified bandwidth the available bandwidth is shared between the slave ports.

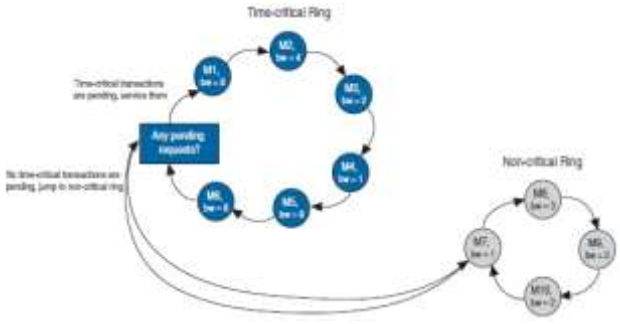


Figure 2: Scheme of Arbitation

In this scheme of arbitration, based on the specified settings each data master contains a share of available bandwidth on the time-critical ring. The arbiter cycles through the data ports, thus grants each the ability of issuing a read/write burst up to the largest supported burst size i.e. 64 bits to the

memory controller. The cycle of arbitration continues around this ring, servicing slave ports whose bandwidth allowance is not exceeded. Whenever the outstanding time-critical requests are nil, the arbiter executes non-critical ring pending transaction. The service of non-critical ring transaction is done, the arbiter checks for new requests if any on the time-critical ring and returns for servicing the request. If request is more than one present on the time-critical ring, the arbiter continues to service them. After all the pending time-critical requests are serviced by the arbiter, it switches back to servicing the non-critical ring. The system shown in Figure 2 contain masters M1, M5, and M6 and each master get 8 shares of the total available bandwidth. The time-critical ring adds up to 31 of the total number of bandwidth shares, and hence the allocation for master M1 is 25% (8/31) of memory bandwidth. Master M2 is allocated 4 shares of the bandwidth available, or one-half of the bandwidth that is allocated to master M1. When the first 6 masters are idle and do not request any memory access, then the masters M7 through M10 have the chance to access the external memory.

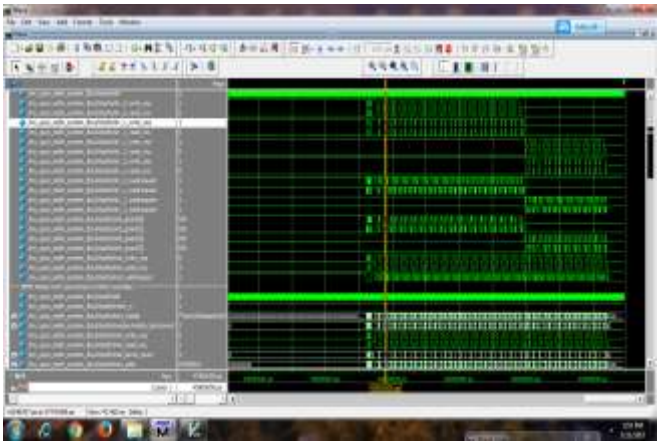
D. Multiplexer

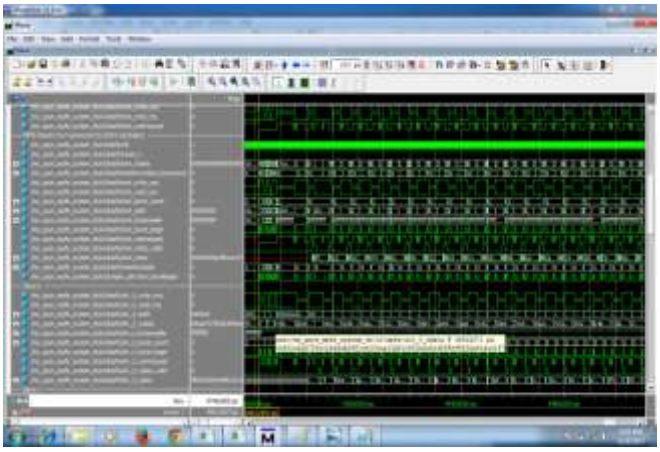
The FSM's used in the system control the data flow of MPFE by utilization two multiplexers, write and read multiplexer. In write state, the PFSM grants the permission to MPFE SLAVE PORT by port index getting from APC, i.e., data from a certain port are pushed to write data properly. The SFSM, conversely, permits read data to connect with a given MPFE SLAVE PORT. The multiplexer size could be expanded with the increasing of NUP.

3. Results and Discussion

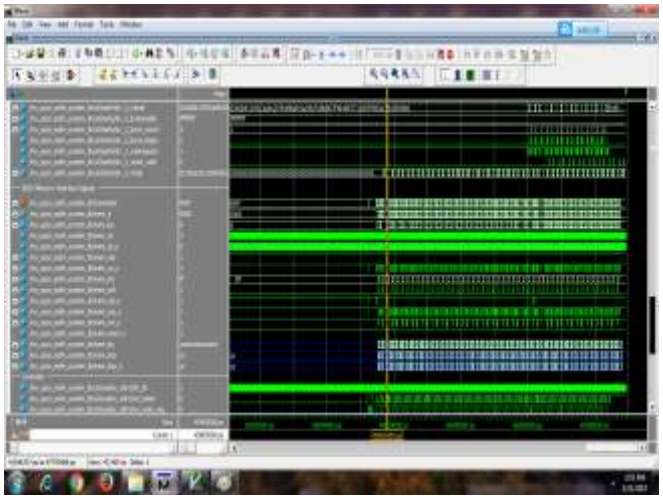
The proposed technique was designed using the Altera Quartus II and Modelsim 6.6a and the language used is Verilog. The various submodules are designed using the intellectual properties for memory that are available in the Altera software. Finally, all the submodules are integrated using the Quad system similar to the SOPC builder that automatically integrates the system modules without manual procedures. The simulated results obtained are as follows,

1. Read and Write results for Master and Slave Ports





2. DDR3 Memory Interfacing Signals



4. Conclusion

The paper concludes with simulation results for the proposed multiport memory controller that uses an external DDR3 SDRAM memory as the master port and a number of slave ports, where a communication is set up between the master and a slave port with increased speed as the usage of DDR3 RAM is done.

References

[1] O'Hare Neil and Kompatsiaris Yiannis, The Multimedia Grand Challenge 2013, IEEE Multimedia, Vol.21, Iss 1, 2014.
[2] John L. Hennessy and David A. Patterson, Memory Hierarchy Design, 2012, http://www.edn.com/design/systemsdesign/4397051/2/Memory-Hierarchy-Design-part-1
[3] Xuan-Thuan Nguyen, Lam-Hoai-Phong Nguyen, Trong-Tu Bui and Huu- Thuan Huynh, A real-time DSP based hand gesture recognition system, IEEE Int. Symp. Signal Proc. Info. Tech. (ISSPIT), Vietnam, 2012.
[4] Altera Handbook, External Memory Interface Handbook, 2013, http://www.altera.com/literature/hb/external-memory/emi.pdf

[5] Sharing External Memory Bandwidth Using the Multi-Port Front-End https://www.altera.com/content/dam/alterawww/global/en_US/pdfs/.../an637.pdf
[6] Chih-Da Chien, Chih-Wei Wang, Chiun-Chau Lin, Tien-Wei Hsieh, Yuan-Hwa Chu, Jiun-In Guo, A Low Latency Memory Controller for Video Coding Systems, IEEE Int. Conf. Multimedia and Expo, China, 2007.
[7] LogiCORE IP Multi-Port Memory Controller, MPMC (v6.03. a) – Xilinx https://www.xilinx.com/support/documentation/ip_documentation/mpmc.pdf
[8] An Efficient Multiport Memory Controller for Multimedia Applications. ieeexplore.ieee.org/iel7/7050531/7058915/07058922.pdf