Implementation of a SDRAM Controller for System on Chip

[1] RAKESH MEHTA, [2] JAYDEEP BHATT, [3] RUTVA PATHAK,

[1, 2,3] M.E.(VLSI & ESD),GTU PG SCHOOL , Gandhinagar, Gujarat, India.

rakeshmehta7790@rediffmail.com , jaydeep.bhatt11@gmail.com

ABSTRACT : The most significant factors in the success of today's system-on-chip (SoC) designs are the ability to deliver efficient access to off-chip high speed memory. This paper introduces a intellectual Property for external SDRAM interface for the high performance and high flexibility memory interface. Several techniques and methods, i.e. data and command buffers based on asynchronous FIFO, multiple bus width adaptation, and improved WISHBONE bus interface are proposed in this paper for fulfilling the requirements mentioned above. And the performance improvement by using techniques is analyzed. At the end of the paper, the simulation and tape-out results are provided. The whole implementation of architecture is proven to be not only functional and efficient, but also flexible, programmable and reusable in general purpose SoC.

.

KEYWORDS: system-on-chip, SDRAM, WISHBONE, intellectual Property.

1. INTRODUCTION

In normal computer system, the memory interfaces are always implemented by specific bridge chips, which are used to transfer data and instructions with external memory. While in embedded system, all of these, including the microprocessor and other co processing units, have to integrate into one chip. So an excellent external memory access interface in the system-on-chip should exploit more performance as much as possible in order to meet the increasing band-width requirement. Many architecture researches and implementations have been done for the improvement of the performance and flexibility of external memory interface. EMIF [2] is high compatible interface architecture. But its structure is not suitable for the physical implementation and also has to bring many redundant cycles during the memory accessing. A External SDRAM Interface IP(ESIIP) is proposed for system on chip in this paper. Fig.1 shows a typical integrated system-on-chip, which uses external SDRAM interface as the main expressway of memory data transmission. Obviously, this interface can increase the integrity and flexibility of system and also reusability.by using such functional tested IP time to market can be minimized.

Keywords: Memory interface, wishbone, ESIIP

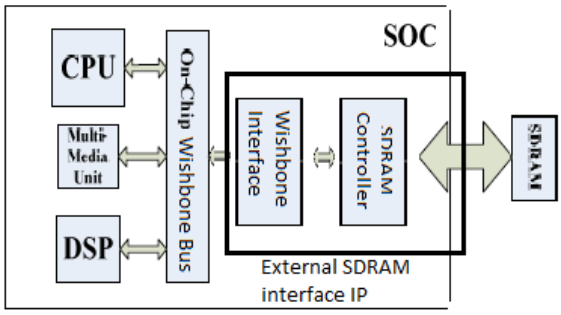

Figure 1 A System-On-Chip integrated with ESIIP

The content of paper is organized as follows: Section 2 introduces the whole architecture of ESIIP. Section 3 provides details of design and some important techniques. Section 4 provides some detail information on the simulation and synthesis with 90nm technology library. Section 5 gives the conclusion.

2. ARCHITECTURE

2.1. Function feature

The ESIIP architecture proposed has the following Special function features:

- Support 8-bit, 16-bit, 32-bit off-chip bus access with the automatic assembly and disassembly ability in order to match different bus-widths.
- Improved high-throughput WISHBONE on-chip bus interface.
- Support any given operation frequency under the maximum limit by using data and command buffer for clock domains crossing.
- Ready to use in System On Chip

2.2. Architecture

The block structure diagram of ESIIP proposed in this paper is shown in Fig.2. Each function unit is introduced below.

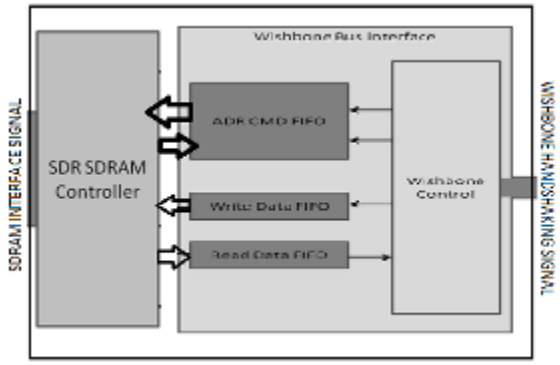

Figure 2 ESIIP block Diagram

The ESIIP architecture adopts 3 different depth FIFOs for crossing two clock domains. These FIFOs buffer all requests and data which are sent by master side from high frequency domain and feedback data from external memory which is low frequency domain.

The 3 FIFOs are read-data buffer, write-data buffer, address and command buffer. All configuration registers accesses are processed at high frequency clock domain.

SDRAM controller is the heart of whole architecture, which is integrated four sub block Bus convertor, Request Generator, Bank Controller, Transfer Controller. This controller module also provides a standard interface for SDRAM. It sends out multiplexed control signals to the external memory. It also controls the logic states of data path and address path. And it connects with all other accessory modules to manage and track every step in transaction process. The data path module and address multiplex module are the main way for data transmission. The Bus converter module within SDRAM Controller automatically pack the data when the onchip bus-width doesn't match the off-chip external memory bus-width. The assembly and disassembly scheme is based on the big or little endian format which is configured at the power-on moment.

Request generator module generate bus width, internal address and burst length Based on the SDRAM request from application.Bank controller module takes requests from request generator, checks for page hit/miss and issues precharge/activate commands and then passes the request to Transfer Controller. Transfer Controller module takes requests from Bank controller, runs the transfer and controls data flow to/from the app. At the end of the transfer it issues a burst terminate if not at the end of a burst and another command to this bank is not available.

Wishbone Interface handles the Protocol handshake between wish bone master and custom SDRAM controller. This module also takes care of necessary clock domain change over. This block include ; Command Async FIFO, Write Data Async FIFO, Read Data Async FIFO.

3. Design and optimization technology

3.1. Use of FIFO ForClock domains crossing

In order to make data processing unit like Wishbone Master get higher performance, the ESIIP and other units on the chip should run in two different clock domains. Using Buffers based on clock domains crossing FIFO is one of the reliable and efficient techniques used for the transmission between two different clock domains. ESIIP adopts 3 data and command buffers based on this technique. The basic structure of this buffer is shown as Fig.3.

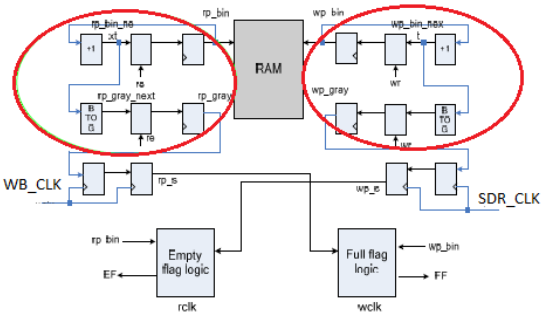

Figure 3 FIFO Block Dia

Two fields which are marked by Red circle are in two different clock domains. The reading clock domain structure on the left side in Fig.3 is introduced as an example. The rp_bin signal is the read pointer of FIFO, which is connected to the address bus of the Dual-port RAM. The Dual-port RAM is used for storing data writing into FIFO. When the logic unit in

the reading clock domain which is on the left in Fig.3 sends its read request signal re, the rp_bin signal, as the reading pointer of FIFO, will increment one automatically. At the same time, reading pointer is translated into Gray code which is only 1 bit different between two neighboring number coded. And this Gray code is going to synchronize to writing clock domain as the indicator of FIFO full flag. The synchronizing mechanism is realized by latching reading pointer's Gray code with two registers controlled by writing domain clock. As mention above, Gray code is a discontinuous binary code [4], and has the important property that there is only one different bit between adjacent Gray codes. This property of Gray code dramatically diminishes the probability of metastability which often appears in crossing clock domain transmission. The reliability of transmission is guaranteed by this technique.

3.2. Multiple bus width Support

ESIIP supports memory widths of 8 bits, 16 bits, and 32 bits, including reads and writes of both big- and little-endian devices. The data packing and unpacking is automatically performed by the GEMI for accesses to external memories of less than 32 bits. For a 32-bit write to an 8-bit memory, the data is automatically unpacked into bytes such that the bytes are written to byte address N, N+1, N+2, and then N+3. Likewise for 32-bit reads from a 16-bit memory, data is taken from half-word address N then N+1, packed into a 32-bit word, and then written to its destination. The byte lane used on GEMI as shown in Fig.5.

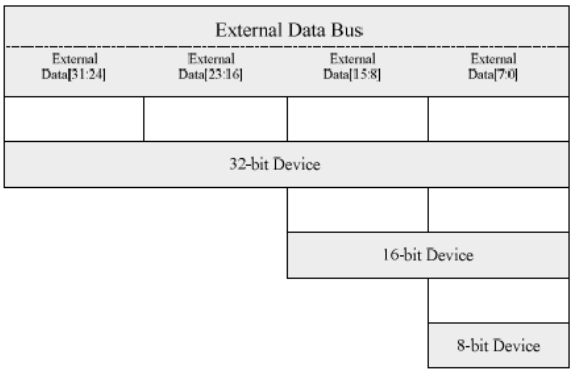

Figure 4 Byte alignment

The external memory is always right aligned to the ED[7:0] side of the bus. The endianness mode determines whether byte lane 0 (ED[7:0]) is accessed as byte address 0 (little endian) or as byte address N (big endian), where 2N is memory width in bytes. This style of interface design provides convenience to the board design engineers and enhances the compatibility of the chip.

3.3. Improved WISHBONE interface

The improved WISHBONE interface is shown as Fig.5. The new extension set of the signals includes ADDR_TAG, ID_TAG_O, ACK_TAG and ID_TAG_I. STB and CYC are similar to the classical WISHBONE protocol [7], which is used to indicate one transaction on the bus. ADDR_TAG is used to indicate which burst style is requested and corresponding burst length. ID_TAG_O is used for indicating the ID number of the transaction request. This is one of the key signals for outstanding read/write and out-of-order transfer. The maximum number of outstanding read/write allowed by improved WISHBONE interface depends on the width of ID_TAG signal. Some extra features added in WB_ACK function. WB_ACK combined with ACK_TAG indicates what kind of acknowledge it is for every acknowledge behavior. There are 4 types of acknowledge, normal ack, reading data ack, last data ack and atomic access ack. ID_TAG_I is used to indicate which transaction request the feedback information from slave belongs to. Fig.5.

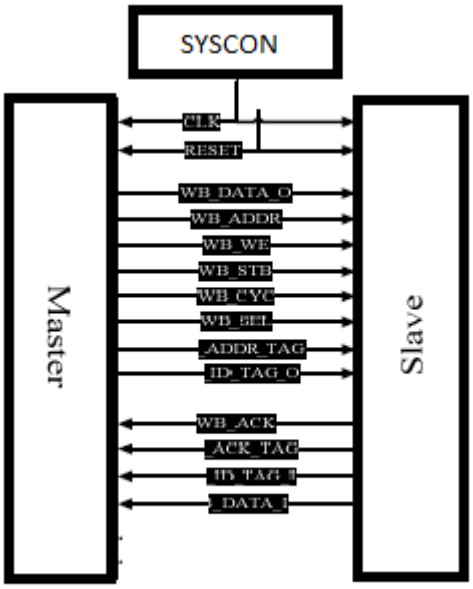

Figure 5 Wishbone Interface

Connection of improved WISHBONE interface We analyze the performance improvement of this on-chip bus interface by taking the SDRAM access as an example. Provided every access address hits in bank management pages, the accessing timings both on chip side interface and off-chip side interface are shown in Fig.6. two times write only take two cycles in the view of the master side, while classical

interface has to take 4 cycles. Similarly, if we define the read operation time is the interval time between the first request sending out and the last data arriving, then the improved interface proposed in this paper only takes 5 cycles, while the classical takes 8 cycles. The speedup is growing with the number of launched outstanding access requests, which is shown in Table 1. Another important advantage of this improved interface is that it cuts off the combinational logic feedback path, which starts from the master, runs through the slave and finally feedback the master. It brings a great benefit to the timing closure in synthesis phase, which can achieve higher operation frequency. And the improved interface provides an excellent solution for clock domains crossing transfer which is impossible to be solved by the classical WISHBONE interface.

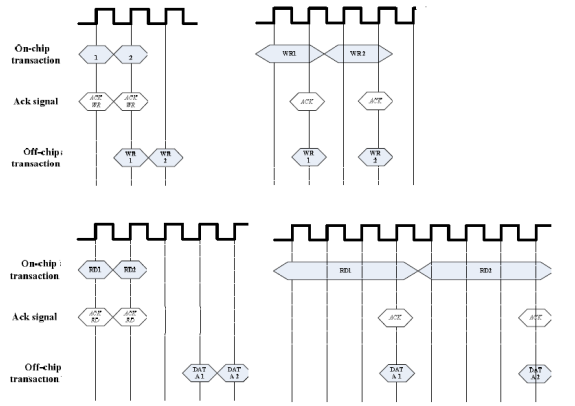

Figure 6 Improved Timing

No of Write/read	CLASSICAL (cycles)	IMPROVED (cycles)	SPEEDUP
2 Writes	4	2	50%
4 Writes	8	4	50%
2 Reads	8	5	37.5%
4 Reads	16	7	56.25%
8 Reads	32	11	65.625%

Table 1 Performance Comparison

4. Simulation and verification

In order to fully verify this ESIIP implementation, a large simulation environment for functional verification has been built. This environment includes the ESIIP RTL model, as many as different kinds of memory simulation like **mt48lc2m32b2, mt48lc4m16, mt48lc8m8a2.** models which are provided by Micron, all kinds of onchip bus and off-chip protocol monitors and unified test vectors' interface. The FPGA prototype verification has also been done with 59.144MHz SDRAM Clock and Wishbone clock of 92.33MHz clock frequency. It is synthesized and taped out in terms of 90nm SAED standard cell library. Reports shows that the ESIIP architecture proposed in this paper can operate stably at 100 MHz.

4. CONCLUSION

In this paper, a high performance and high flexibility external SDRAM interface intelactual property called ESIIP is proposed. In order to fulfill the requirement of mass data processing and complex background applications, several techniques are described and applied in the ESIIP architecture proposed in the paper. All of the new techniques can co-operate very well and make the whole architecture achieve high throughput and high compatibility. The simulation and silicon verification reports show that ESIIP is proven to be not only functional and efficient, but also flexible, programmable and reusable.

REFERENCES

[1] Carter, J., Hsieh, W., Stoller, L., et al., "Impulse: Building a Smarter Memory Controller", *Fifth International Symposium on High-Performance Computer Architecture*, 1999, pp. 70-79.

[2] Dong Wang, J. Ma, S. Chen, Y. Guo, "The Design and Analysis of a High Performance Embedded External Memory Interface", *Proceedings of the Second International Conference on Embedded Software and Systems*, 2005.

[3] Lee, K-B., Lin, T-C., Jen, C-W., "An efficient qualityaware memory controller for multimedia platform SoC", *IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 5*, 2005, p.620-633.

[4] Clifford E. Cummings, Peter Alfke, "Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons", *SNUG-2002*, San Jose, CA, 2002.

[5] Hansoo Kim, and In-Cheol Park, "High-Performance and Low-Power Memory-Interface Architecture for Video Processing Applications", *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, vol. 11, no. 11*, 2001.

[6] Takizawa T., Hirasawa M., "An efficient memory arbitration algorithm for a single chip MPEG2 AV decoder",*International Conference on Consumer Electronics,* 2001, pp. 182-183.

[7] *Specification for the WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores. Rev.B3*,
Opencores Organization, September 7. 2002.