IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, July 2022
Impact Factor: 6.252

Optimized DMA Controller for Soc Interconnections using AMBA

Naval Kishor Sharma[1] and Ritu Juneja[2]
Research Scholar, Department of Electronics & Communication Engineering[1]
Assistant Professor, Department of Electronics & Communication Engineering[2]
School of Engineering & Technology, Bahadurgarh, Haryana, India

Abstract: *In this work, the design and implementation of an AMBA based advanced DMA controller is proposed. The DMAC has 6 channels which support hardware triggers, linking operation and channel chaining transfer and provides three dimensions transmission by parameter sets to improve the real-time processing capability, so as to perform data block moving, data sorting and sub-frame extraction of various data structures. All channels can also be used for channel chaining transfer triggered automatically by Interrupt and Error module. The channel chaining capability for the DMAC allows the completion of a DMAC channel transfer to trigger another DMAC channel transfer.*

Keywords: AMBA, BUS, ARB BUS, TX-FIFO, RX-FIFO, IP INTERFACE

I. INTRODUCTION

Amba stands for advanced microcontroller bus architecture.it are widely used as the on chip bus in soc (system on chip) design. Three distinct buses are defined within the amba specification:

- Advanced high-performance bus (ahb)
- Advanced system bus (asb)
- Advanced peripheral bus (apb)

Among which ahb used for high speed low latency and high frequency operation.
ahb supports 32,64 and 128 bit data bus implementations with a fixed 32 bit address bus. it is a sycvhronous bus that supports and pipelining of accesses to improve throughput ahb support multiple masters. the arbiter has the task of determining which master gets to do on access. every transfer has an address/control phase and seperate data phase. they are both pipelined (able to start the next transfer's arbitration and address phase while finishing the current transfer.
A typical AMBA AHB system design contains the following components:

1. AHB master: A bus master is able to initiate read and write operations by providing an address and control information. Only one bus master is allowed to actively use the bus at any one time.
2. AHB slave: A bus slave responds to a read or write operation within a given address-space range. The bus slave signals back to the active master the success, failure or waiting of the data transfer.
3. AHB arbiter: The bus arbiter ensures that only one bus master at a time is allowed to initiate data transfers. Even though the arbitration protocol is fixed, any arbitration algorithm, such as highest priority or fair access can be implemented depending on the application requirements. An AHB would include only one arbiter, although this would be trivial in single bus master.
4. AHB decoder: The AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer. A single centralized decoder is required in all AHB implementations.

II. METHODOLOGY

When DMAC transfers data, it is as a master on ARB bus. When realizing data transfer between ARB slave and APB peripheral, DMAC must buffer data and apply to APB Bridge for visiting APB peripheral. The buffer mode leads to transfer rate not high. We have proposed a new DMAC architecture which lies in between ARB bus and APB bus with APB Bridge function, that is, DMAC controls directly data, address and control signals on APB bus. So it could

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, July 2022
Impact Factor: 6.252

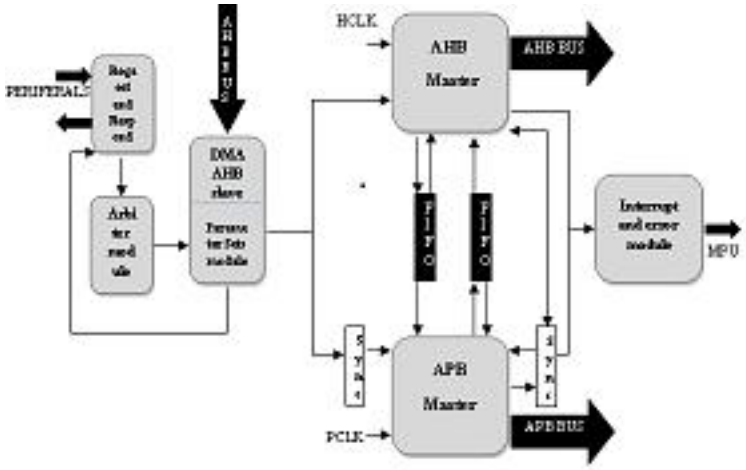
achieve ARB operation and APB operation run in parallel. And data transfer mode can be buffer and non-buffer mode according to practical application by setting control register.

III. ALGORITHM

The purpose is to allow the ability to chain several transfers through one transfer occurrence. dmac combines hardware and software-programmable arbitration mechanism. In hardware channel priority, the lowest-numberedchannel has the higher priority and software-programmable arbitration mechanism adopts weighted priority rotational algorithm proposed named"weighted priority rotational algorithm on pci bus arbitration".

IV. PROPOSED DMAC ARCHITECTURE

This section provides work process of the DMA controller. Fig shows the DMAC architecture. firstly, mpu programs the parameter set associated with the channel by dmaahb slave interface. Request and respond module accepts the request of data transfer from peripherals, software transfer request, or chaining transfer. Those requests enter arbiter module. The selected request finds its corresponding parameter set in parameter sets module which submits transfer parameters to ahb master module and apb master module pasting synchronizer.



AHB master module asserts bus request signal to get access to the ahb according to parameter set, and completes data transfer between AHB and FIFO. APB master module asserts bus request signal to gain the control of APB after arbitration with APBbridge, and completes data transfer between APB and FIFO. AHB operation and APB operation are independent, and DMAC could achieve AHBoperation is in parallel with APB operation. When data transfer completes or error occurs in the process of data transfer, interrupt and error module asserts interrupt signal or error signal to the mpu.

V. AHB TO APB OPERATION

DMAC as AHB master enables four transfer types: AHB slave-to-AHB slave, AHB slave-to-APB peripheral, APB peripheral-to-AHB salve, and APB peripheral-to-APB peripheral. And DMAC has incrementing and wrapping address modes for source and destination and supports for 8, 16 and 32 bit wide transactions

5.1 AHB Slave –to – AHB Slave

In AHB slave-to-AHB slave, in order to meet the requirements of real-time, DMAC reads data from AHB slave for one burst, writing into FIFO, and asserts request bus signal for write operation. When one burst data is read completely and DMAC occupies bus again, DMAC writes data to AHB slave. Transfer operations carry out in order, until task terminates.

5.2 AHB Slave –to-APB Peripheral

In AHB slave-to-APB peripheral, if the APB peripheral is slow equipment, user could adopt the way of transfer with FIFO buffer by set transfer mode register in parameter set, and the process is like AHB slave-to-AHB slave.

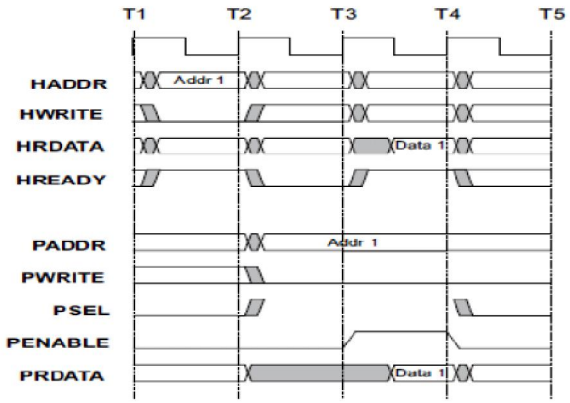
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, July 2022
Impact Factor: 6.252

If the frequency of APB peripheral is match to AHB bus frequency, DMAC can implement transfer without FIFO buffer. AHB read operation is in parallel with APB write operation, forming a two-stage pipeline, thus transfer speed is increased greatly.

5.3 APB Peripheral –to-AHB Slave

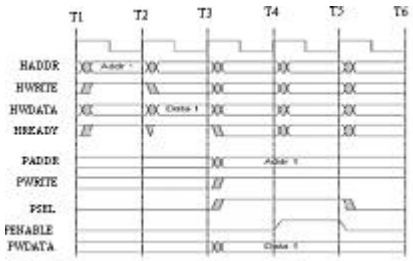
Interfacing the AMBA APB to the AHB is described in:

- Read transfers
- Write transfers
- Read transfers



The transfer starts on the AHB at time T1 and the address is sampled by the APB Bridge at T2. If the transfer is for the peripheral bus then this address is broadcast and the appropriate peripheral select signal is generated. This first cycle on the peripheral bus is called the SETUP cycle, this is followed by the ENABLE cycle, when the PENABLE signal is asserted .During the ENABLE cycle the peripheral must provide the read data. Normally it will be possible to route this read data directly back to the AHB, where bus master can sample it on the rising edge of the clock at the end ofENABLE cycle, which is at time T4 in Figure.

5.4 Write Tranfer



Single write transfers to the APB can occur with zero wait states. The bridge is responsible for sampling the address and data of the transfer and then holding these values for the duration of the write transfer on the APB

5.5 APB Peripheral to APB Peripheral

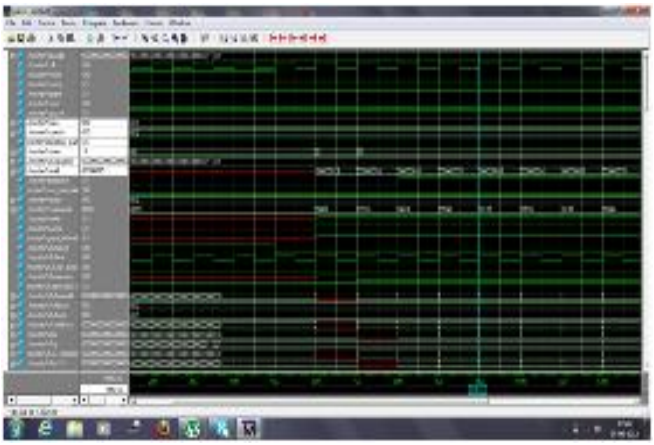
When using a combination of peripherals, some designed to the revision specification and others designed to previous revisions, it is recommended that a revision bridge is used and the earlier version peripherals are converted for use with the new bridge. There are two fundamental differences between there v D and rev APB specifications.

- The timing of the strobe signal compared to the enable signal.
- The point at which read data is sampled.

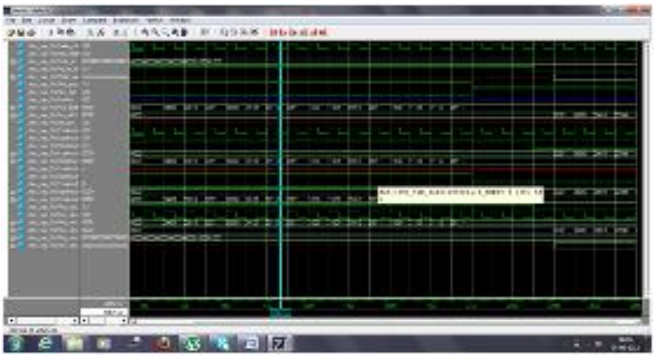
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, July 2022
Impact Factor: 6.252

VI. IMPLEMENTATION AND SIMULATION

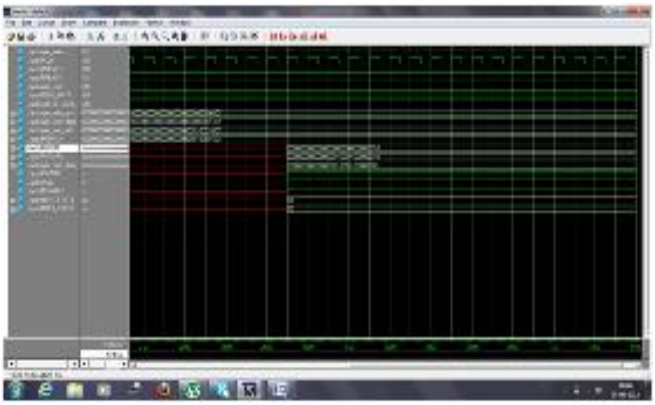
Simulation results for AHB slave_to AHB master



Simulation results for APB to AHB



Simulation results for APB MASTER



Simulation results for AHB to APB

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, July 2022
Impact Factor: 6.252

VII. COMPARISON

The performance of this DMAC is better. Each data transfer rate is increasedby about 50%. Between AHB slave and APB peripheral, our DMAC adoptsnon-buffer mode to transfer data, and the rate is increased by 67%

	AN2548 DMA	PL081 DMA	PDMA (buffer)	PDMA (non-buffer)
AHB to AHB	1920	989	989	-
AHB to APB	3072	1320	1564	1012
APB to AHB	3072	1320	1564	1012
APB to APB	3840	1728	1883	-

From Table, if this DMAC uses buffer mode, the performance of PL081is better than ours. But the time spent more than PL081 is used in signals synchronization. This DMAC adopts dual-clock domain design, and PL081 only has one clock RCLK. When this DMAC uses non-buffer, even though signal synchronization will occupy much time, the performance of this DMAC is better than PL081, and the data transfer rate is increased by 23.3%.And this DMAC has more features than PL081.

VIII. CONCLUSION

In this project, a design and implementation of an AMBA based advanced DMAC controller is proposed. The DMAC has 8 channels which support hardware and software triggers, linking operation and channel chaining transfer to improve the real-time processing capability and provides three dimensions transmission so as to perform data block moving, data sorting and sub-frame extraction of various data structures. Channel arbitration mechanism adopts hardware priority combined with weighted priority rotational algorithm, so that meet the different requirements of fairness and the priority in different systems. And the DMAC supports incrementing and wrapping address modes and completes data transfer which the data-width of read and write is different by asymmetric asynchronous FIFO. Moreover the DMAC adopts dual-clock domain design so as to decrease the power consumption. Furthermore the DMAC has the function of APB Bridge, and it can control address, data and control signals independently and achieve ARB bus and APB bus to run in parallel. And the DMAC could adopt buffer and non-buffer data transfer mode according to the speed of equipment. Non-buffer mode can enhance the data transfer rate significantly.

This new DMAC architecture which lies in between AHB bus and APB bus with APB Bridge function, that is, DMAC controls directly data, address and control signals on APB bus. So it could achieve AHB operation and APB operation run in parallel. And data transfer mode can be buffer and non-buffer mode according to practical application by setting control register. Experimental results show that the DMAC has the advantage of high speed transfer rate and is much suitable to various application fields, such as multimedia processing.

REFERENCES

[1]. Guoliang Ma, Hu He , "Design And Implementation Of And Advanced DMA controller On Amba-Based SoC", ASIC, 2009. ASICON '09. IEEE 8th International Conference on Digital Object, Page(s): 419 – 422

[2]. A. Olugbon, S. Khawam, T. Arslan, I. Nousias, I. Lindsay, "An AMBAAHB-based reconfigurable SoC architecture using multiplicity of dedicated flybyDMA blocks", Design Automation Conference, 2005. Proceedings of the ASPDAC2005. Asia and South Pacific, Volume: 2,Year: 2005 , Page(s).1256 – 1259

[3]. S. Hessel, D. Szczesny, F. Bruns, A. Bilgic, Hausner, J. Vehicular,"Architectural analysis of a Smart DMA Controller for Protocol StackAcceleration in LTE terminals",Technology Conference Fall (VTC 2010-Fall),2010 IEEE 72nd Digital Object, Page(s). 1 – 5

[4]. Jaehoon Song, Piljae Min, Hyunbean Yi, "Design of Test AccessMechanism for AMBA-Based System-on-a-Chip",Sungju Park VLSI TestSymposium, 2007. 25th IEEE , Page(s).375 – 380

[5]. Hang Yuan, Hongyi Chen, "An improved DMA controller for high speeddata transfer in MPU based SOC",GuoqiangBai Solid-State and Integrated Circuits Technology, 2004. Proceedings. 7th International Conference onVolume: 2 Digital Object, Page(s). 1372 – 1375

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, July 2022
Impact Factor: 6.252

[6]. LufengQiao, "Design of DMA controller for multichannel PCI bus frame engine and data link manager",Zhigong Wang Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conferenceon Volume: 2 Digital Object,Page(s). 1481 – 1485.

[7]. D, Szczesny, S. Hessel, S. Traboulsi, "Optimizing the Processing Performance of a Smart DMA Controller for LTE Terminals",Bilgic, A. Embedded and Real-Time Computing Systems and Applications (RTCSA),2010 IEEE 16th International Conference on Digital Object , Page(s). 309-315

[8]. Chia-Hao Yu, Chung-Kai Liu, Chih-Heng Kang, Tsun-Hsien Wang, Chih- ChienShen, "An Efficient DMA Controller for Multimedia Application in MPU Based SOC", Shau-Yin Tseng Multimedia and Expo, 2007 IEEEInternational Conference on, Page(s). 80 – 83.

[9]. Prokin, "DMA transfer method for wide-range speed and frequency measurement", M.Instrumentation and Measurement, IEEE Transactions Volume: 42 , Issue: 4 , Page(s). 842 – 846.

[10]. Tai-Yi Huang. Liu, J.W.-S., "A method for bounding the effect of DMA I/O interference on program execution time", Hull, D.Real-Time Systems Symposium, 1996., 17th IEEE , Page(s). 275 – 285

[11]. S. Osborne, A.T. Erdogan, T. Arslan, "Bus encoding architecture for low-power implementation of an AMBA-based SoCplatform",Robinson, .Computers and Digital Techniques, IEEE Proceedings- Volume: 149 , Issue: 4 , Page(s). 152 – 156

[12]. M. Pockrandt, P. Herber, "Model checking a System C/TLM design of the AMBA AHB protocol", Glesner, S.Embedded Systems for Real-Time Multimedia (ESTIMedia), 2011 9th IEEE Symposium on, Page(s). 66 – 75

[13]. Yi-Ting Lin, Chien-Chou Wang, "AMBA AHB bus potocol checker with efficient debugging mechanism",Ing-Jer Huang Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on , Page(s). 928 – 93

[14]. M. Dubois, Y. Savaria, "A generic AHB bus for implementing high- speed locally synchronous islands",Bois, G.Southeast Con, 2005. Proceedings. IEEE , Page(s). 11 – 16

[15]. C. Toal, S. Sezer, "A pipelined SoPC architecture for 2.5 Gbps network processing",Xing Yu Field-Programmable Custom Computing Machines, 2003. FCCM 2003. 11th Annual IEEE Symposium on , Page(s), 271 – 272

[16]. S. Sezer, C. Toal, "A pipelined SoPC architecture for data link layer protocol processing",Xing Yu SOC Conference, 2003. Proceedings. IEEE International [Systems-on-Chip] , Page(s). 277 – 278

[17]. C. Toal, "A 32-bit SoPC implementation of a P5,Sezer", S.Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on ,Volume 1, Page(s). 504 – 507

[18]. M. Prokin, "DMA transfer method for wide-range speed and frequency measurementt", Instrumentation and Measurement,IEEETransactions,Volume: 42 , Issue: 4 1993 , Page(s).842 – 846

[19]. AN2548 Application note. http://www.st.com.

[20]. AMBA Specification (Rev 2.0). http://www.arm.com.

[21]. TMS320DM643x DMP EDMA3 User's Guide. SPRU987, January 2007.