IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 2, September 2015)

Design & Implementation of AHB Interface for SOC Application

Sangeeta Mangal
M. Tech. Scholar
Department of Electronics & Communication
Pacific University, Udaipur (India)
enggsangeetajain@gmail.com

Nakul Mangal
Asst. Professor
Department of Electronics & Communication, USB Group of Colleges of Engineering and Technology, Abu- Road, Rajasthan (India)
nakulmangal@yahoo.com

Govind Singh
M. Tech. Scholar
Geetanjali institute of technical studies Udaipur R.T.U. Kota
ec.govindsingh@gmail.com

***Abstract*— In The AMBA™ on-chip interconnect system is an established open specification that details a strategy on the interconnection and management of functional blocks that makes up a System-on-Chip (SoC).**

In this work, the design and implementation of an AMBA based AHB Master and AHB Slave with Memory controller interface is proposed. It is majorly categorized in two dedicated feature i.e. decision (AHB MASTER) and response (AHB SLAVE).

Moreover AHB master enables transfer types i.e. burst mode and AHB Master-to-AHB slave supports incrementing and wrapping addressing modes and completes data transfer which the data width of read and write is different by asymmetric asynchronous FIFO .A bridge between AHB Master and AHB slave with application of memory controller will been shown and there digital efficiency in terms of area and speed will be discussed. Control structure will be designed with finite state machine. The IP of AHB Master and AHB Slave will been implemented in Xilinx Spartan-3 3s50pq208-5.

***Keywords* — AMBA, AHB Master, AHB Slave, SOC.**

I. INTRODUCTION

Motivation

Systems-on-Chip (SoC) and in particular embedded real-time systems typically consist of several computational elements. These elements full-fill different tasks for processing an overall solution. Let's take a set-top box for TV sets as an example. A set-topmost generate a TV-signal for a particular TV channel from a digital satellite signal. This process takes different tasks. One task is to split the incoming digital signal into data streams, such as video and audio. Another task is to convert the video stream into an actual TV-signal. One more conversion has to be made to turn the audio stream into an audio signal for the TV set. Meanwhile, another task handles the user input such as changing the channel when the remote control is pressed. All these tasks have to be done in parallel and are bound by real-time deadlines. The cost of missing these deadlines is visible as black boxes on the screen or audible as noise. This is unacceptable and therefore it is necessary to always deliver this data within hard real-time deadlines. These computational elements are either general-purpose processors or digital signal processors. Nowadays, multiple of them are integrated into a System-on-Chip solution. A processor needs to interact with other processors, memories or I/O devices to complete a task. Currently busses are used to interconnect these IP blocks. The current research in the field suggests using Networks-on-Chip (NoC) to interconnect IP blocks, because NoCs allow more flexibility than busses. However, to get NoCs accepted as communication paradigm in SoCs there are still left open research questions according to. An example is how to deal with voluminous storage. High volume storage is usually put off-chip as dynamic memory. The separation is necessary because the manufacturing process is a different one for memories to that for standard logic. Dynamic memories provide high data rates and high storage capabilities. Nevertheless, dynamic memories lack flexibility in accessing random memory locations efficiently and require refresh cycles to keep the content. Allowing efficient communication between IP blocks and a shared memory requires a memory controller. IP blocks, which communicate with the memory, are named hereinafter requestors. The memory controller arbitrates between the requestors and manages the memory accesses. The goal of this paper is to design a memory controller to provide hard real-time guarantees, and provides an efficient communication between IP blocks and the memory. The

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 2, September 2015)

memory controller should also be easily/efficiently combined with the interconnection.

Problem Definition

In SOC biggest challenges are Integrity, Microprocessor performance has improved rapidly these years. In contrast, memory latencies and bandwidths have improved little. The result is that the memory access time has been a bottleneck which limits the system performance. Memory controller (MC) is designed.

Even master operates on different protocol and it is interaction with memory then it will send or receives its data with helps of its slave Interface.

Objective

As increasing numbers of companies adopting the AMBA system, it has rapidly emerged as the de-facto standard for SoC interconnection and IP library development. AMBA enhances a reusable design methodology by defining a common backbone for SoC modules.

In this work, the design and implementation of an AMBA based Memory controller is proposed .The AMBA based Memory controller gives an ease of integration for sub-frame extraction of various data structures in SOC.AMBA based interaction deals with role specific operation. It is majorly categorized in two dedicated feature i.e. decision (AHB MASTER) and response (AHB SLAVE).

Moreover AHB master enables transfer types i.e. burst mode and AHB Master-to-AHB Slave supports incrementing and wrapping addressing modes and completes data transfer which the data width of read and write is different by asymmetric asynchronous FIFO.

A bridge between AHB with application of memory controller will be shown and there digital efficiency in terms of area and speed will be discussed. Control structure will be designed with finite state machine. The IP of AHB Master and AHB Slave will been implemented and its interface with memory controller will be designed and tested.

II. Our Design

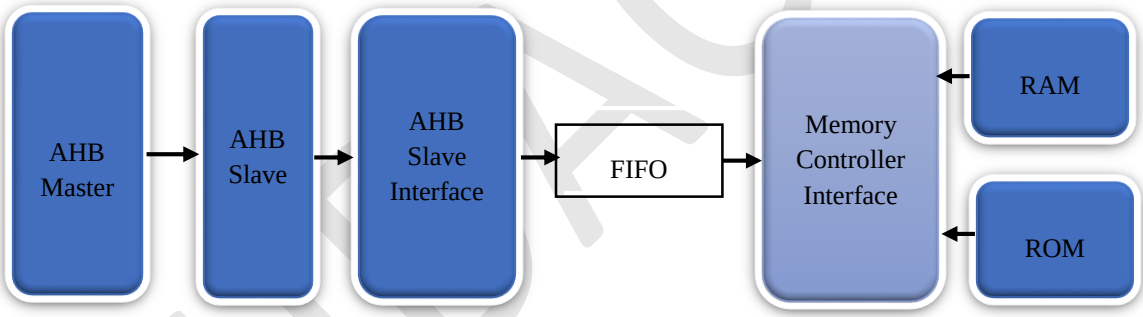


Figure 1: Top architecture of AHB memory controller

The above figure shows the top level implementation of AHB compliant Memory controller. Data is initiated by master and communicated through slave to memory controller. Initially master generates the data and control signals and further those controls cannot directly communicate with any given generic memory, hence data processed through slave .further data passes through slave interface .we have used FIFO for data and control buffering so that even slave and memory are in different CLK then also our communication is full proof. It reduces the complexity. Further data is communicated through ram or rom depends upon the read and write communication.

AHB Bus Master

The AHB Master is an interface unit which allows the processor to initiate data transfer to the AHB slave. The master takes care of address and data transfer between processor and memory. It generates the corresponding control signal as it is triggered by an arbitration unit as an HGRANTx signal. It supports burst mode operation and generates the sequence of operations according to all handshaking signals.

The bus master handles all four types of slave response and its wait state, i.e. HREADY. It generates four transfer types of HTRANS to complete the handshaking with the other interfaces of the slave. Figure 2 shows the pin diagram of the AHB Master Interface, which is necessary for establishing

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 2, September 2015)

the interface between the AHB Master and the AHB Slave with arbitration unit.

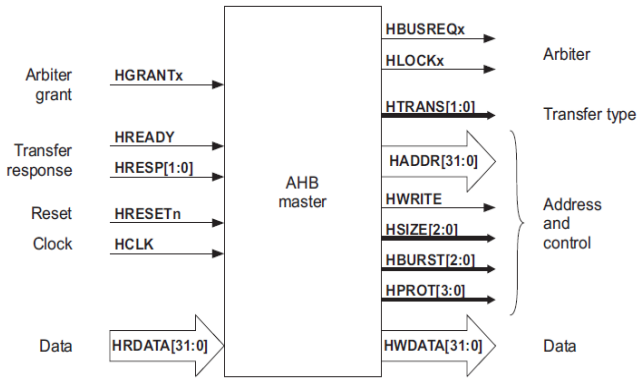


Figure 2: AHB Bus Master Interface

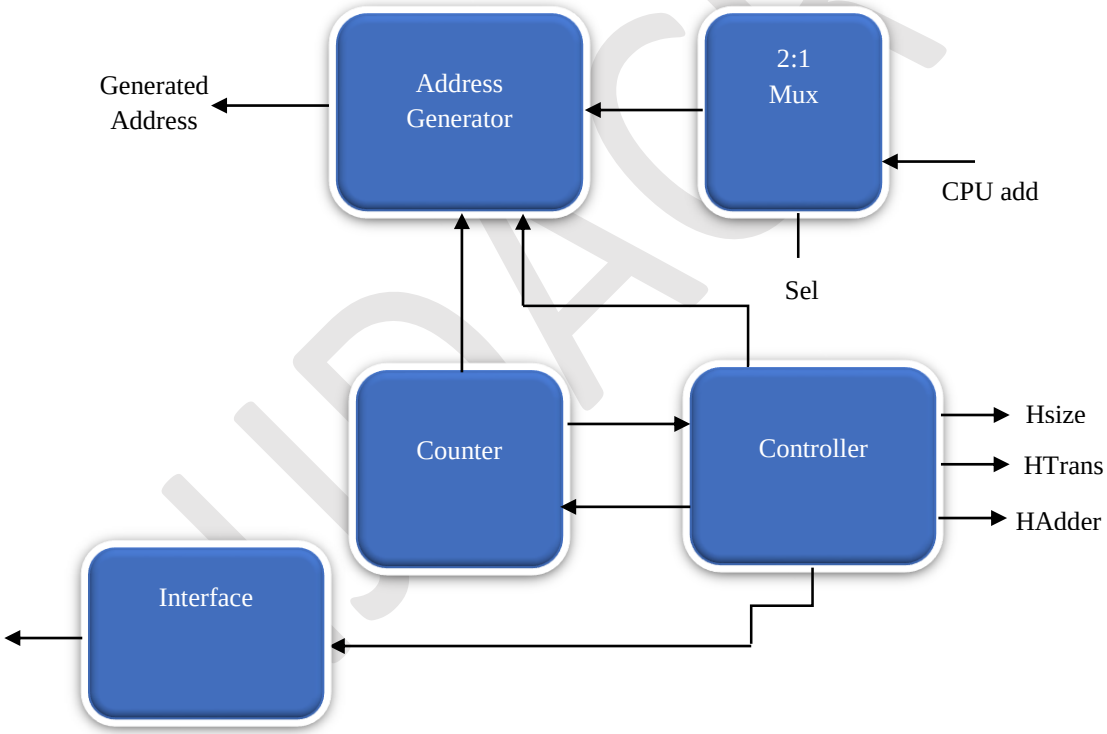


Figure 3: AHB Master

AHB Bus Slave

An AHB bus slave responds to transfers initiated by bus-masters within the system. The slave uses a HSELx select signal from the decoder to determine when it should respond to a bus transfer. All other signals required for the transfer, such as the address and control information, will be generated by the bus master.

The fig 4 shows the state diagram of memory controller .It is a finite state machine implementation initial condition is reset state which is an idle state when no operation is there. When start signals arrived then this FSM triggers, depends upon the instruction its operation is decided by CMD state. According to instruction it moves to ram read, ram write and rom read operation.

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 2, September 2015)

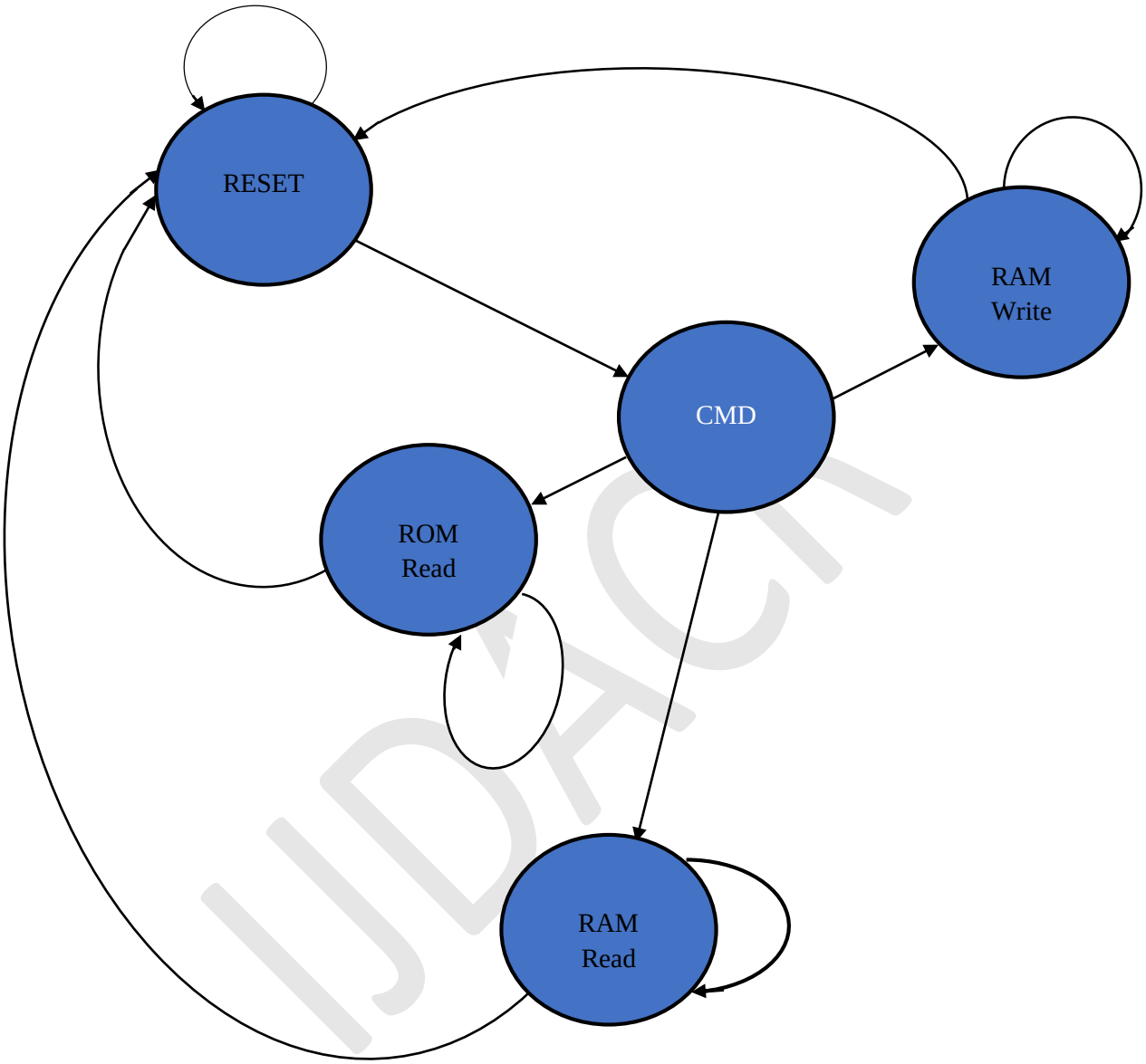


Figure 4: Memory controller

Memory Controller

The two main tasks for the core memory controller are to handle all the timings between different commands and to keep track of which rows that are currently activated. The activation of rows are time consuming and therefore the core memory controller has a look ahead functionality where the arbitrator can notify which command that is in turn to be executed after the current one has finished. This makes it possible to activate the row in advance if the next command is not accessing the same bank or chip as the current command.

FIFO

FIFO is a method of processing and retrieving data. In a FIFO system, the first items entered are the first ones to be removed. In other words, the items are removed in the same order they are entered.

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 2, September 2015)

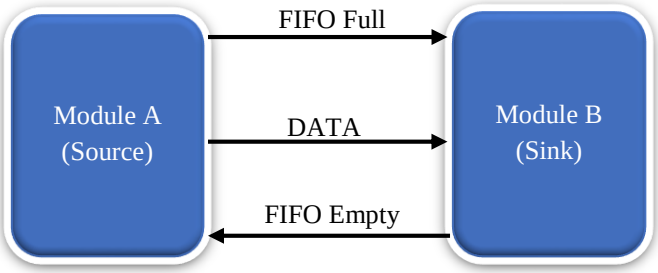


Figure 5 FIFO

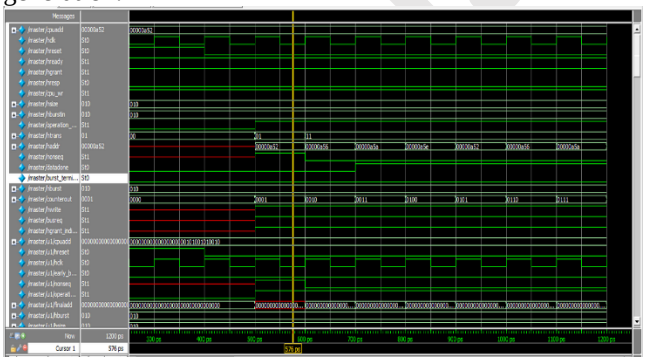
In Figure 5, two modules (called the source and the sink) are connected to one another. When data is being passed from module to module, the source is the module that is outputting data. The sink is the module that is receiving that data. Figure 5 also shows three signals between the two:

Data, FIFO Full, and FIFO Empty. Data is the wire that actually passes data from the source to the sink. FIFO Full and FIFO Empty are known as handshaking signals which allow the source and the sink to communicate with regards to when it is time to pass the data.

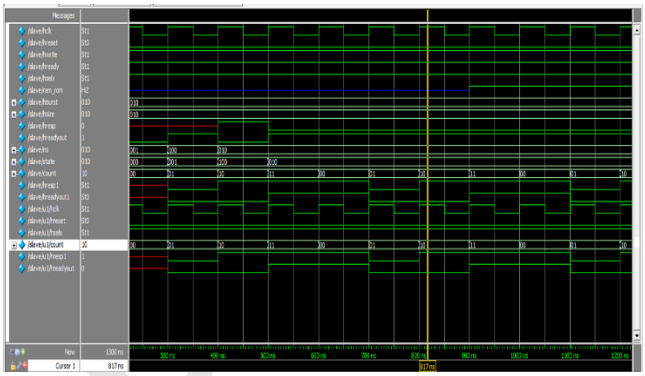
The FIFO Full signal indicates that the FIFO is full, put valid data on the Data line. FIFO Full is what is called a state signal: it is high only when data is valid. If data is not valid on the Data line during a particular cycle, Valid should be low during that cycle.

III. Simulation Results

The figure given below shows the simulation results of master. CPU initiates the first address and rest of address is generated by master according to HBURST and HSIZE. Master will generate HTRANS signal as a response. When hready is low then master will not generate it next address. Once hready is high then master starts is next address generation.



In figure given below result shows the handshaking between slave and slave interface where slave is directly interacting with master and its response is given by slave, Further the read write operation is forwarded to memory controller by slave interface.



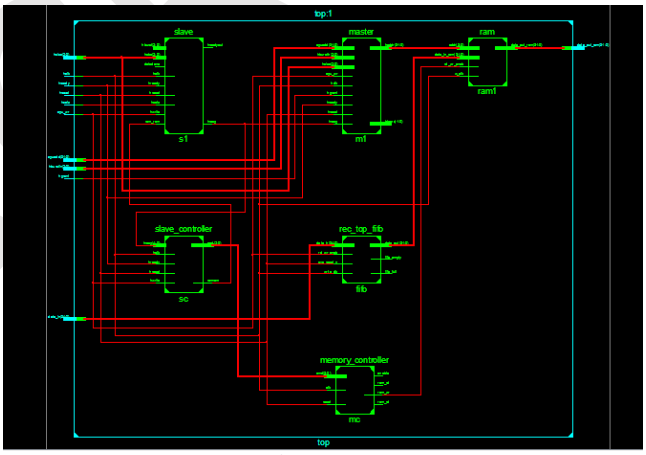


Figure 6: Top architecture of AHB master slave memory controller

Final Results
Device Utilization Summary:

Selected Device: 3s50pq208-5

Number of Slices:	160	out of	768	20%
Number of Slice Flip Flops:	111	out of	1536	7%
Number of 4 input LUTs:	273	out of	1536	17%
Number used as logic:	177			
Number used as RAMs:	96			
Number of IOs:	108			
Number of bonded IOBs:	79	out of	124	63%
IOB Flip Flops:	32			
Number of GCLKs:	3	out of	8	37%

Timing Summary
Speed Grade: -5

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 2, September 2015)

Minimum period: 6.425ns (Maximum Frequency: 155.647MHz)
Minimum input arrival time before clock: 7.269ns
Maximum output required time after clock: 6.141ns

Table 1: Power Supply Summary

	Total	Dynamic	Quiescent
Supply Power (mW)	27.34	0.00	27.34

Table 2: Power Supply Currents

Supply Source	Supply Voltage	Total Current (mA)	Dynamic Current (mA)	Quiescent Current (mA)
Vccint	1.200	5.08	0.00	5.08
Vccaux	2.500	7.00	0.00	7.00
Vcco25	2.500	1.50	0.00	1.50

IV. CONCLUSION

In this paper AHB based Memory Controller is designed. The design has been implemented using VHD for SOC solution. The design has taken care of balance between area and speed. Master IP has been implanted and its address generator block is implemented with Single adder instead of 64 typical adder hence we have manage to reduce the area with help of proper design .Every implementation is in structural approach hence it's become a well-documented frame. There is separate implementation of slave and slave interface with inter faces the memory controller further. To avoid the handshaking complexity we have used FIFO for memory controller interface and slave interface. Even it increases the latency but at the same time it makes design simple and bottleneck problem free. The design has been synthesized on XILINX 13.1 Spartan 3, and simulated in MODELSIM.