International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 11 | November - 2022 Impact Factor: 7.185 ISSN: 2582-3930

# A DEVICE DRIVER FOR NAND FLASH MEMORY CONTROLLER IN OPEN POWER A2O PROCESSOR

S.Mubeen Taj [1], Dr B.Lalitha [2],Dr N.Uday Bhaskar[3]
M.Tech[1], CSE Dept[1], JNTUACEA[1], ANANTHAPURAMU[1] A.P[1], India[1]
Associate Professor[2], CSE Dept[2], JNTUACEA[2], ANANTHAPURAMU[2] A.P[2], India[2]
Assistant Professor[3], CSE Dept[3],Govt College[3], ANANTHAPURAMU[3] A.P[3], India[3]

Abstract— AMBA AXI4 is the fourth generation of the AMBA interface specification from ARM. AXI is a Parallel high performance, synchronous, high frequency, multi-master multi-slave communication interface. Flash memory is a non-volatile electronic computer memory storage medium that can be erased and reprogrammed electrically. Flash memory is widely used for storage and data transfer in consumer devices ,enter prise systems and industrial applications, .NAND flash memory is a type of non-volatile storage technology that does not require power to retain data. An important goal of NAND flash development has been to reduce the cost per bit and to increase maximum chip capacity so that flash memory can compete with magnetic storage devices such as hard disks.Device drivers are software interfaces between software applications and hardware devices .As part of complex operating system, device drivers are considered extremely difficult to develop. They are usually developed in low-level programming languages, such as C. Code written in C language will be converted into binary which is fed into processor to perform operations with desired module. This project we write the test cases for NAND flash memory through AXI4 bus interface which is connected to A2O core. Tools required for design and development of driver is GCC compiler and C language will be used to develop drivers

**Keywords**- Advanced microcontroller bus architecture (AMBA) System-on-chip(SoC), Intellactual Property (IP), AMBA, AXI, VCS, Verilog

## I. INTRODUCTION

To achieve a specific capability, the processor must be equipped with the necessary peripherals and components. If a feature is requested, the relevant hardware components can be integrated to accomplish the desired functionality. If you need to connect a lot of peripherals, it will complicate the design and production process, as well as take longer. It would have an impact on the product's release date and time to market. With the advancement of technology, more components will need to be incorporated, and the design process will become more sophisticated. As a result, the complete functionality is transferred to a single Integrated Circuit (IC) or chip in order to simplify the complexity of the design process and integration. The technique of integrating various functions that are normally required for at system into a single chip is known as System on Chip. On a single chip/IC, it has processor cores, memory subsystems (DDR controllers), IO subsystems, and another functional logic/IP. A system on a chip (SoC) is a digital substrate that could encompass analog, digital, mixed-signal, or radio frequency functionalities. A graphical processing

unit (GPU), a multi-middle critical processor unit (CPU), and system memory are the maximum not unusual place components (RAM). Signal processing, Wi-Fi communication, synthetic intelligence, and lots of different operations may be achieved relying at the form of system that has been shriveled to the dimensions of a chip.

The AMBA [2] data bus width can be 32, 64, 128 or 256 byte, address bus width will be 32bits wide. The AMBA AXI4 [3] specification to interconnect different modules in a SoC was released in March 2010. A. AMBA AXI4 architecture AMBA AXI4 [3] supports data transfers up to 256 beats and unaligned data transfers using byte strobes. In AMBA AXI4 system 16 masters and 16 slaves are interfaced. Each master and slave has their own 4 bit ID tags. AMBA AXI4 system consists of master, slave and bus (arbiters and decoders). The system consists of five channels namely write address channel, write data channel, read data channel, read address channel, and write response channel. Device drivers are software interfaces between software applications and hardware devices .As part of complex operating system, device drivers are considered extremely difficult to develop. They are usually developed in low-level programming languages, such as C. The device driver developers must have an in-depth understanding of given hardware and software platforms. Code written in C language will be converted into binary which is fed into processor to perform operations with desired module.

## II. AMBA AXI4 PROTOCOL

AMBA AXI3 protocol has separate address/control and data phases, but AXI4 has updated write response requirements and updated AWCACHE and ARCACHE signaling details. AMBA AXI4 protocol supports for burst lengths up to 256 beats and Quality of Service (QoS) signaling. AXI4 has additional information on Ordering requirements and details of optional user signaling. AXI3 has the ability to issue multiple outstanding addresses and out-oforder transaction completion, but AXI4 has the ability of removal of locked transactions and write interleaving. One major up-dation seen in AXI4 is that, it includes information on the use of default signaling and discusses the interoperability of components which can't be seen in AXI3

### A. AMBA AXI4 master

To perform write address and data operation the transaction is initiated with concatenated input of [awaddr, awid, awcache, awlock, awprot, awburst]. On the same lines for read address and data operations the concatenated input is [araddr, arid, arcache, arlock, arprot, arburst]. The addresses of read and write operations are validated by VALID signals and sent to interface unit.

### B. AMBA AXI4 Interconnect

The interconnect block consists of arbiter and decoder. When two masters initiate a transaction simultaneously, the arbiter gives priority to access the bus. The decoder decodes the address sent by master and the control goes to one slave out of 16. The AMBA AXI interface decoder is centralized digital block. The decoder decodes the address sent by master and goes to one slave out of 16. 0-150 locations are meant for slave-1, next 151-300 addressable locations are meant for slave-and so on till slave-16.

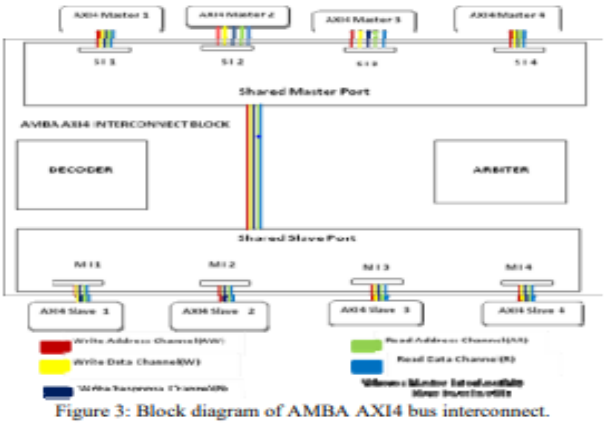



Figure 3: Block diagram of AMBA AXI4 bus interconnect.

C. AMBA AXI4 slave read/write block diagram

The AXI4 slave consists of common read/ write buffer which stores the read/ write address and data as shown in fig. 4. Pending read address register stores the remaining read addresses to be sent; pending write address register which stores the remaining write addresses to be sent and pending write data register which stores the remaining write data to be sent. The read/write state machines receive internal inputs from the read/ write buffer. The AXI4 slave test bench initiates the read or write transaction and the output from the AXI4 slave are standard read/write channel signals. The AXI4 slave receives the write data in the same order as address. Signals used to design slave module is shown in fig. 5. The test layer shown in the fig. 5 has 2 test cases.

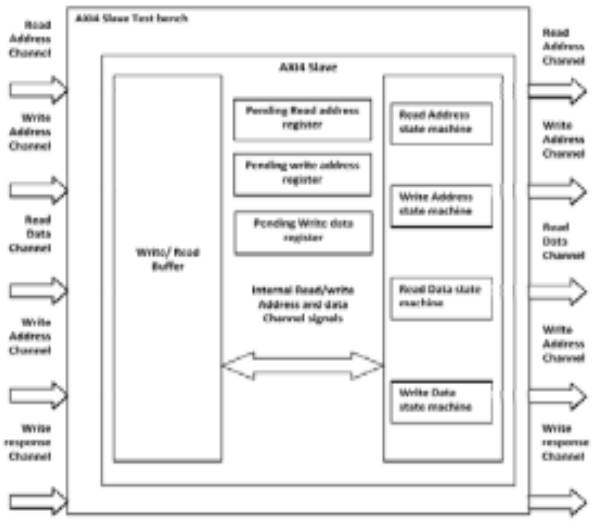


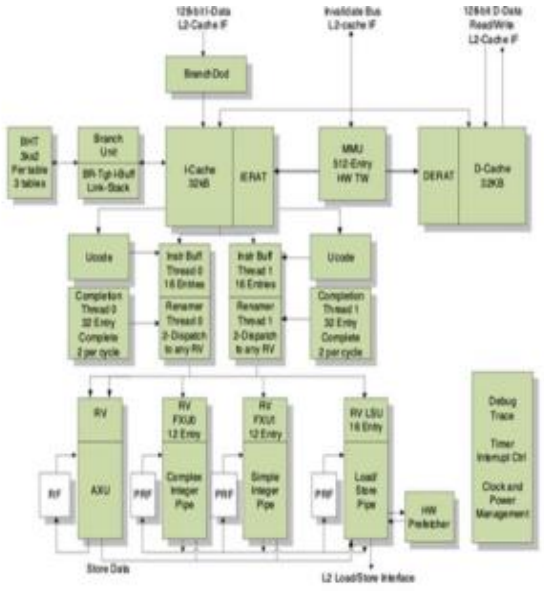

Figure 4: AMBA AXI4 slave Read/Write block Diagram.

## III. EXISTING SYSTEM

In a SoC, it houses many components and electronic modules, to interconnect these a bus is necessary. There are many buses introduced in the due course some of them being AMBA developed by ARM, CORE CONNECT developed by IBM, WISHBONE developed by Silicore Corporation, etc. Different buses have their own properties the designer selects the bus best suited for his application. The AMBA bus was introduced by ARM Ltd in 1996 which is a registered trademark of ARM Ltd. Later advanced system bus (ASB) and advanced peripheral bus (APB) were released in 1995, AHB in 1999, and AXI in 2003[6]. AMBA bus finds application in wide area. AMBA AXI bus is used to reduce the precharge time using dynamic SDRAM access scheduler (DSAS). Here the memory controller is capable of predicting future operations thus throughput is improved. Efficient Bus Interface (EBI) [8] is designed for mobile systems to reduce the required memory to be transferred to the IP, through AMBA3 AXI. The advantages of introducing Network-on-chip (NoC) within SoC such as quality of signal, dynamic routing, and communication links was discussed in. To verify on-chip communication properties rule based synthesizable AMBA AXI protocol checker is used. This integration could result in a plethora of portable devices that can be carried anywhere at any time without relying on systems to function. In the real world, these SoCs have been used to do a wide range of tasks, including video, audio, games, wireless communication, and many others.SoCs are used in today's laptops and personal computers to focus on the area. time, and power so that overall performance can be improved by integrating all duties of a system on a single chip.

## IV. PROPOSED WORK

The technique of integrating various functions that are normally required for at system into a single chip is known as System on Chip. On a single chip/IC, it has processor cores, memory subsystems (DDR controllers), IO subsystems, and another functional logic/IP. A system on a chip (SoC) is a digital substrate that could encompass analog, digital, mixed-signal, or radio frequency functionalities. The goal of developing a System on Chip (SoC) is to reduce area, power, time. and cost as we progress into the future. These requirements can be satisfied by combining many operations onto a single lowpower processor utilizing an SoC. This integration could result in a plethora of portable devices that can be carried anywhere at any time without relying on systems to function. In the real world, these SoCs have been used to do a wide range of tasks, including video, audio, games, wireless communication, and many others.SoCs are used in today's laptops and personal computers to focus on the area. time, and power so that overall performance can be improved by integrating all duties of a system on a single chip.



Workflow of proposed model

## V. RESULTS

Simulation is carried out in VCS tool and Verilog is used as programming language.

A. Simulation result for write operation

The AResetn signal is active low. Master drives the address, and the slave accepts it one cycle later. The write address values passed to module are 40, 12, 35, 42 and 102 as shown in fig. 8 and the simulated result for single write data operation is shown in fig. 9. Input AWID[3:0] value is 11 for 40 address location, which is same as the BID[3:0] signal for 40 address location which is identification tag of the write response. The BID[3:0] value is matching with the AWID[3:0] value of the write transaction which indicates the slave is responding correctly. BRESP[1:0] signal that is write response signal from slave is 0 which indicates OKAY

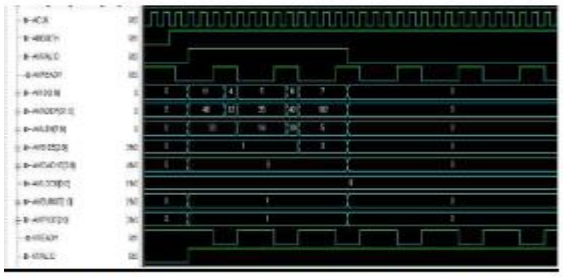


Figure 8: Simulation result of slave for write address operation.

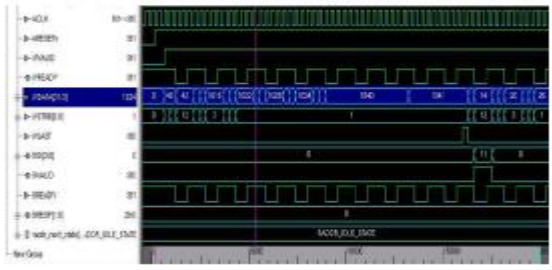


Figure 9: Simulation result of slave for single write data operation.

B. Simulation result for read operation

The read address values passed to module are 45, 12, 67, 98 as shown in fig. 11 and the simulated result for single read data operation is shown in fig. 12. Design of AMBA AXI4 protocol for System-on-Chip communication 42 Simulation result of slave for read address operation. Input ARID[3:0]

value is 3 for 12 address location, which is same as the RID[3:0] signal for 12 address location which is identification tag of the write response. The RID[3:0] and ARID[3:0] values are matching, which indicates slave has responded properly. RLAST signal from slave indicates the last transfer in a read burst.

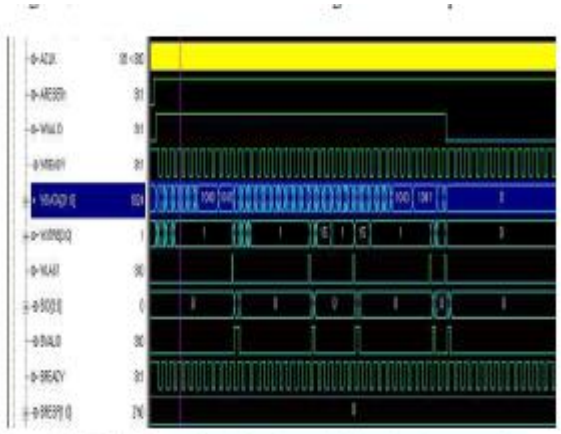


Figure 10: Simulation result of slave for multiple write data operation.

.

## VI. CONCLUSION AND FUTURE SCOPE

NAND flash memory is a type of non-volatile storage technology that does not require power to retain data. An important goal of NAND flash development has been to reduce the cost per bit and to increase maximum chip capacity so that flash memory can compete with magnetic storage devicessuch as hard disks. This work developed, simulated, and synthesized the AXI4 NAND Flash Controller, which was utilized in A20 processor based fabless soc. This proposed architecture has the potential to improve data transmission performance using AXI4. This design is verified in Xilinx Vivado and Mentor Questa using memory-based function verification.

The plan for the future is to increase memory capacity and data transfer speed using DMA handshake

## REFERENCES

[1]. Gong Xin, Dai Zibin, Li Wei, Feng Lulu, “Design and Implementation of a NAND Flash Controller in SoC” IEEE International Conference of Electron Devices and Solid-State Circuits, 2011.

[2]. C. Lin and L. Dung, "A NAND Flash Memory Controller for SD/MMC Flash Memory Card," in IEEE Transactions on Magnetics, vol. 43, no. 2, pp. 933-935, Feb. 2007.

[3]. Lattice Semiconductor NAND Flash Memory Controller Specifications

[4]. S. T. Jose and C. Pradeep, "Design of a multichannel NAND Flash memory controller for efficient utilization of bandwidth in SSDs," 2013 International MutliConference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013.

[5]. K. Agarwal, V. K. Magraiya and A. K. Saxena, "Verification and Simulation of New Designed NAND Flash Memory Controller," 2013 International Conference on Communication Systems and Network Technologies, 2013.