DIO: 30.0485/ijearst.06.01.3478 https://ijearst.co.in/

ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

ISSN 2386-7876
Vol.01, Issue.03
Mar-2026
Pages: 712-725

LIGHT WEIGHT DIRECT MEMORY ACCESS IN FPGA USING AXI PROTOCOL

[1] **V V KRISHNA,** [2]**VESAPOGU KIRAN KUMAR RAJ,** [3]**HEMANADHA REDDDY MADARAM,** [4]**M SIDHARTHA REDDY,** [5]**T VENKATA SIVA,** [6]**SANA RAJASEKHAR**

[1]Guide, Dept of ECE, ABR College of Engineering and Technology, Kanigiri, A.P.
[2,3,4,5,6]B. Tech, Dept of ECE, ABR College of Engineering and Technology, Kanigiri, A.P.

Abstract: The project aims to design a soft core processor system with Advanced eXtensible Interface (AXI) processor bus which deals with different data capacities with 32, 64, 128, and 256 bits data width. The system deals with Direct Memory Access (DMA) unit to transfer data between the system memory and external peripheral. Memory Controller Block – Dual Data Rate (MCB-DDR5) external memory is introduced to act as main memory system. Registers in DMA controller are designed using general ring counter which consumes more power.

Keywords: Advanced eXtensible Interface, Direct Memory Access, Buffer, Clock gating.

INTRODUCTION: Embedded design techniques tools are used now a day to design embedded processor system to be configured on FPGAs. Processor system usually consist a soft core processor (MicroBlaze) or hard processor (PowerPC), processor bus, memories and peripherals. The most commonly used buses in such a type of system are Processor Local Bus (PLB v4.6). In [1] a DMA controller is designed to act with Micro blaze processor system configured on Spartan-3A FPGAs. The system is designed to perform data transfer between the internal block RAM an external peripheral. In [2] a DMA system is depicted to act with multiprocessor connected via On-chip Processer Bus (OPB). In [3] a DMA mode is proposed to act as a universal synchronous/ a synchronous Receiver/Transmitter (USART) IP soft core in Altera kit with AVALON bus. In the work AXI processor bus is used instead of PLB processor bus with adding a MCB-DDR2 external memory. AXI processor system must be adapted to deal with different data width due to application diversities to cope with this challenge the need for a flexible processor bus arose. Xilinx Company developed and advanced extensible bus interface (AXI) that is configurable on Spartan-6 FPGAs. Direct memory access (DMA) is a technique for handling data that offloads the work from the CPU to dedicated hardware. In applications that require moving considerable amounts of high-speed data, a DMA is necessary both to achieve maximum data transfer speed and to free the CPU to complete other processing tasks [1]. Advanced eXtensible Interface (AXI) protocol is the communication protocol Xilinx uses in their FPGA devices [2]. For the purposes of this paper, we will refer to the AXI4 protocol and will use AXI and AXI4 interchangeably. To communicate with a Xilinx device's memory, a DMA controller must use AXI protocol. For engineers designing systems that require DMA on an FPGA, Xilinx provides an AXI DMA controller. This DMA uses AXI4-Lite protocol for communication with the CPU, AXI4-Full protocol for communication with the memory, and AXI4-Stream (AXIS) protocol for pulling in data to write or pushing out data that has been read. Because of this, the write channel is referred to as Stream to Memory Map (S2MM) and the read channel as Memory Map to Stream (MM2S). This controller boasts many features that allow it to work in numerous applications. In many applications, though, not all of the features are necessary. In these cases, a smaller, simpler DMA is more desirable. Though Xilinx allows their AXI DMA to be configured for a wide range of modes ranging from streamlined to more fully functional, the smaller configuration still requires significant space. Other custom DMA Controllers are focused on optimization for specific applications, such as Fjeldtveft and Orlandic's CubeDMA which is customized to more efficiently access three-dimensional data acquired from hyperspectral imaging [3]. Similarly, Wang, Wei, Tao, and Nan propose a DMA optimized for increasing bus utilization for two-dimensional data in a multimedia data crypto-engine [4]. Both of these DMAs focus on increasing throughput. Getting started with direct memory access on Xilinx boards may be initially overwhelming. First of all Xilinx distinguishes AXI DMA and AXI VDMA in programmable fabric. AXI DMA refers to traditional FPGA direct memory access which roughly corresponds to transferring arbitrary streams of bytes from FPGA to a slice of DDR memory and vice versa. VDMA refers to video DMA which adds mechanisms to handle frame synchronization using ring buffer in DDR, on-the-fly video resolution changes, cropping and zooming. Video DMA is covered in next article. In addition to AXI DMA and AXI VDMA there is a DMA engine built into the ARM core which is also out of the scope of this article. Both AXI DMA and AXI VDMA have optional scatter-gather support which means that instead of writing memory addresses or framebuffer addresses to control registers the DMA controller grabs them from linked list in DDR memory. The purpose of the DMA is to reduce the processor load. As the name suggests it accessing direct memory for peripheral devices. If the DMA is used with a processor then data access from memory is done by DMA instead of the processor. DMA allows peripheral devices to access memory directly without dependence on the processor. Therefore the processor can carry out other tasks simultaneously while DMA accesses memory. With this, the overall performance of the system is improved (Abdullah, 2016). DMA. It seems to be an easy concept but not complicated with the implementation of the system with others hardware subsystem. DMA has many other important applications, such as network cards as well as graphics card, and disk drive controller etc. In a computer system, DMA plays an important role in accessing memory, and vital section or system entity embedded. In addition, it plays an important role in the SOC system. It provides speed is quite good for transferring data to peripheral devices that are connected externally. That DMA performance increases when working with bus architecture.

Literature Survey: In the existing design literature, DMA with bus architecture is available. Intel has designed the first DMA with the IC 8237 chip number in 1981 IBM PC using this IC 8237 DMA for the first time in it product. This IC DMA 8237 uses bus architecture, with industry standard architecture (ISA) to improve its performance. This has been designed to transfer data between system memory and peripherals (Zayati et al, 2012; Oded Maler, 2012). This DMA design has four channels and transferring data 1.6 megabytes every second. Individual channels have 64 kilobytes memory address and capable of transferring 64 KB data with single programming instructions (Barry, 1997). Initially, the bus system and ISA bus were identical. Because the IBM CPU is on cloned to work on a higher frequency compared to the ISA expansion bus, they separate. The ISA bridge is used for separation (Hou, 2013). In 1992 the peripheral component interface (PCI), the new bus architecture was introduced. That communication between PCI and ISA is through the board. Next, PCI to Isa recommended adapter (Jinbiao, 2013). Because of this, the basic architectural design used for contains logic adapter blocks. Hardware block, therefore includes the PCI bus interface circuit design, ISA bus interface circuit design, and I / O find module logic blocks (HOU, 2013). This PCI-bus architecture works on the principle of the master (becoming a master). In the Architecture, only one device will get a bus control at once. Use some arbitrers. Techniques some devices can access the bus. Improvement in the bus architecture took place when used the concept of packet switching in full duplex mode to several interfaces device memory and system. This increases the bus architecture and named PCI Express (PCIE) (Li et al, 2009) (Anand, 2013; Mengshengwei, 2016). It has a pair of x1 link in it architecture, which contains channels to transmit and receive separately. Because of this, bandwidth doubled, when compared to previous architecture.

Regardless of this bus for DMA operations, embedded products use very useful specifics Bus architecture in the SOC, known as the Advanced Microcontroller Bus Architecture (AMBA) (Abdullah, 2016). Amba is a registered trademark (ARM, 2017) in the IC industry for sophisticated Reduction of Computer Set Instructions (RISC) Machines (ARM Ltd). In this continuation by at the end of 1997 the first native of the Amba interface with the cash that was cash was introduced. Amba is an interconnection specification (on-chip), which is used to manage and connect various functional blocks under SOC. Amba provides support for various controllers, processors, multiprocessor and peripheral systems. Amba is found to be an open standard system in the industry. There are two types of bus systems defined in specifications, of AMBA architecture, that is AHB and APB of AMBA architecture, such as AHB and APB. At present, Amba is widely used in the application modern cellular devices based on integrated circuits and SOC based based. This products use state machines separately for transmission and receipts, for moderate data transfer rate (Mohammed et al, 2015); (Ejidokun et al, 2018) and (Berawi, 2013). In the proposed research study, we designed direct memory access controllers (DMAC) for embedded system based products. This DMA controller works on a sophisticated microcontroller bus architecture (Amba). DMA performance and data transfer rates for large volumes data increases high when using a bus architecture like Amba. In this way, we first improved DMA controller performance and then we use this DMA machine in embedded system to improve system performance and processor performance used in the same system. The performance characteristics of the processor are embedded with DMA the controller is always better than without a DMA controller. These performance characteristics are outlined in an article by cypress Semiconductor (Sachin et al, 2010).

EXISTING METHOD:

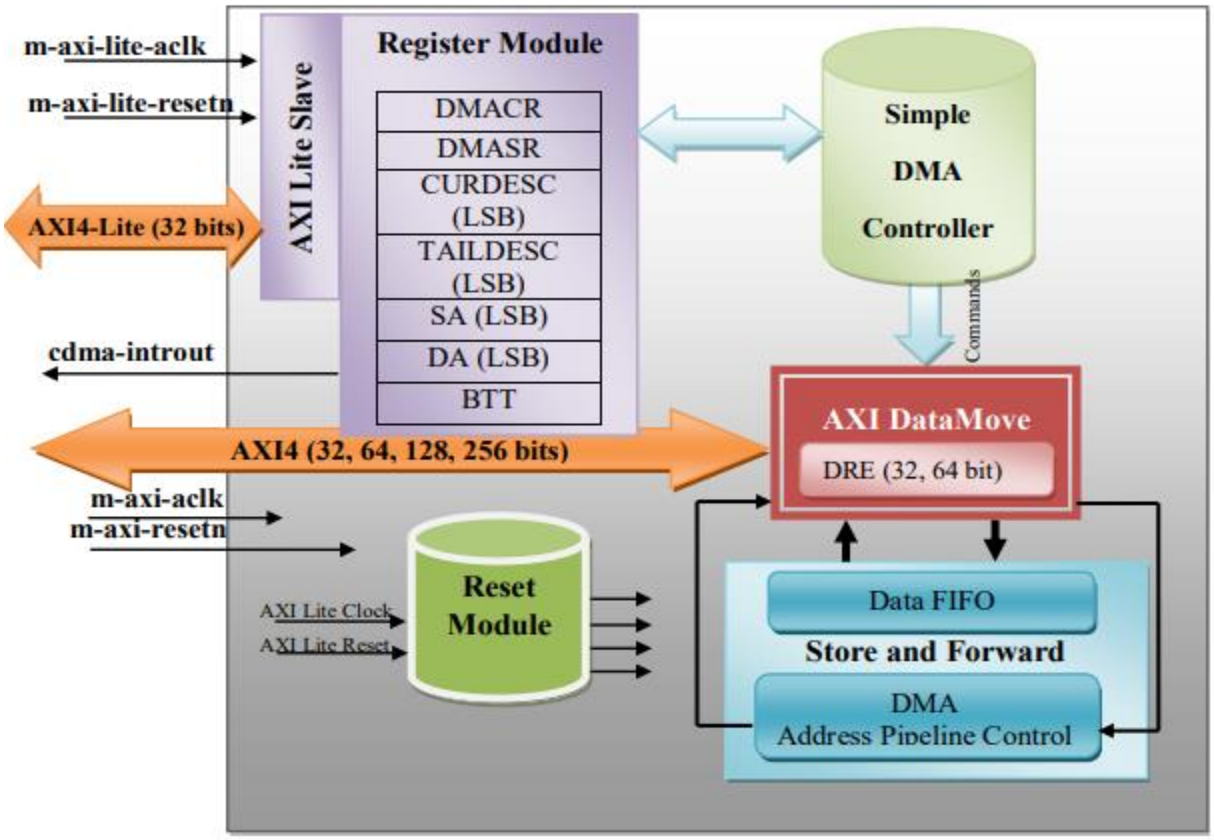


Fig:1 Main block diagram

REGISTERS AS DELAY BUFFERS ARCHITECTURE: DIGITAL BUFFER Digital Buffers and Tri-state Buffers can provide current amplification in a digital circuit to drive output load. In a previous tutorial we looked at the digital not gate commonly called an inverter, and we saw that the NOT gates output state is the complement, opposite or inverse of its input signal.

MEMORY ORGANIZATION: Delay buffer works quite similarly like a fixed jitter buffer, that is it will delay the frame retrieval by some interval so that caller will get continuous frame from the buffer. This can be useful when the operations are not evenly interleaved, for example when caller performs burst of put() operations and then followed by burst of operations. With using this delay buffer, the buffer will put the burst frames into a buffer so that get() operations will always get a frame from the buffer (assuming that the number of get() and put() are matched). The buffer is adaptive, that is it continuously learns the optimal delay to be applied to the audio flow at run-time. Once the optimal delay has been learned, the delay buffer will apply this delay to the audio flow, expanding or shrinking the audio samples as necessary when the actual audio samples in the buffer are too low or too high.

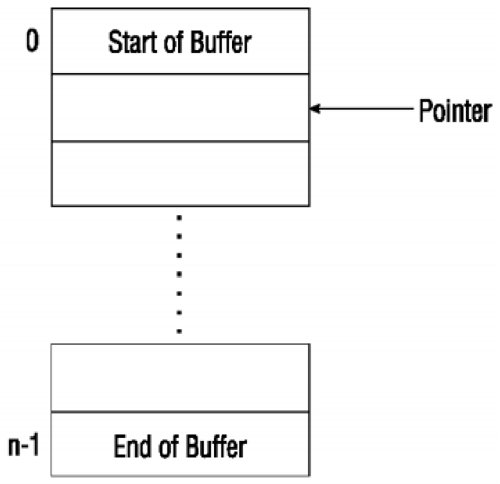


Fig:2 Buffer

PROPOSED METHOD:

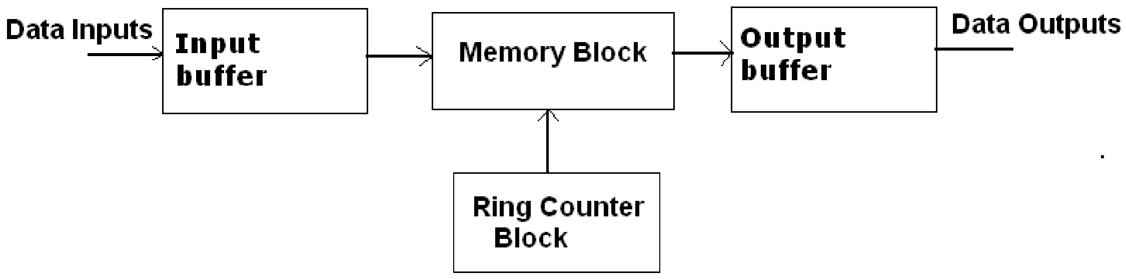


Fig3 : Proposed Block of Memory Organization

INPUT BUFFER: The Input buffer is also commonly known as the input area or input block.
When referring to computer memory, the input buffer is a location that holds all incoming information before it continues to the CPU for processing. Input buffer can be also used to describe various other hardware or software buffers used to store information before it is processed. Some scanners (such as those which support “include” files) require reading from several input streams.

MEMORY BLOCK:

(RAM) Random-access memory (RAM) is a form of computer data storage. Today, it takes the form of integrated circuits that allow stored data to be accessed in any order (that is, at random). "Random" refers to the idea that any piece of data can be returned in a constant time, regardless of its physical location and whether it is related to the previous piece of data.

The word "RAM" is often associated with volatile types of memory (such as DRAMmemory modules), where the information is lost after the power is switched off. Many other types of memory are RAM as well, including most types of ROM and a type of flash memory called NOR-Flash.

Scan design has been the backbone of design for testability (DFT) in industry for about three decades because scan-based design can successfully obtain controllability and observability for flip-flops. Serial Scan design has dominated the test architecture because it is convenient to build. However, the serial scan design causes unnecessary switching activity during testing which induce unnecessarily enormous power dissipation. The test time also increases dramatically with the continuously increasing number of flip-flops in large sequential circuits even using multiple scan chain architecture. An alternate to serial scan architecture is Random Access Scan (RAS). In RAS, flip-flops work as addressable memory elements in the test mode which is a similar fashion as random access memory (RAM). This approach reduces the time of setting and observing the flip-flop states but requires a large overhead both in gates and test pins. Despite of these drawbacks, the RAS was paid attention by many researchers in these years. This paper takes a view of recently published papers on RAS and rejuvenates the random access scan as a DFT method that simultaneously address three limitations of the traditional serial scan namely, test data volume, test application time, and test power.

RING COUNTER:

A **ring counter** is a type of counter composed of a circular shift register. The output of the last shift register is fed to the input of the first register.

There are two types of ring counters:

A *straight ring counter* or *Overbeck counter* connects the output of the last shift register to the first shift register input and circulates a single one (or zero) bit around the ring. For example, in a 4-

register one-hot counter, with initial register values of 1000, the repeating pattern is: 1000, 0100, 0010, 0001, 1000... . Note that one of the registers must be pre-loaded with a 1 (or 0) in order to operate properly.

A *twisted ring counter* (also called *Johnson counter* or *Moebius counter*) connects the complement of the output of the last shift register to its input and circulates a stream of ones followed by zeros around the ring. For example, in a 4-register counter, with initial register values of 0000, the repeating pattern is: 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000... . If the output of a shift register is fed back to the input. aringcounter results. The data pattern contained within the shift register will recirculate as long as clock pulses are applied. For example, the data pattern will repeat every four clock pulses in the figure below. However, we must load a data pattern. All **0**'s or all **1**'s doesn't count.

We make provisions for loading data into the parallel-in/ serial-out shift register configured as a ring counter below. Any random pattern may be loaded. The most generally useful pattern is a single **1**.

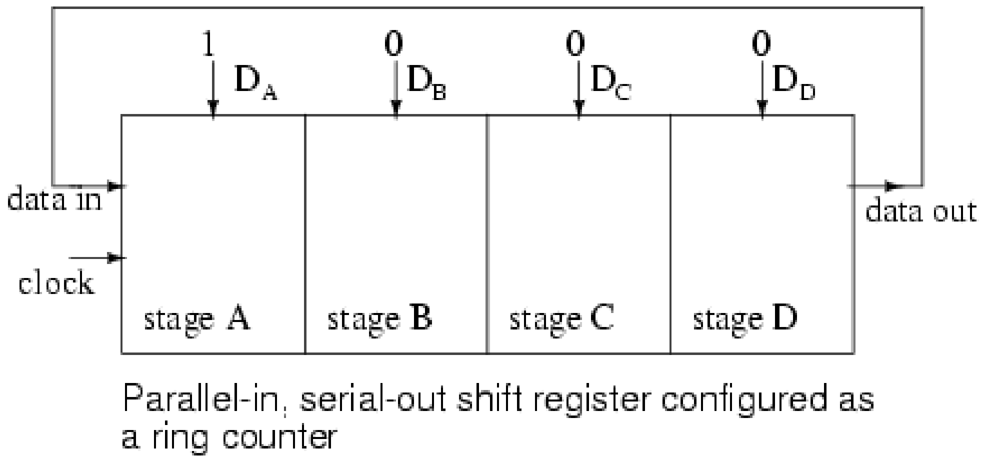
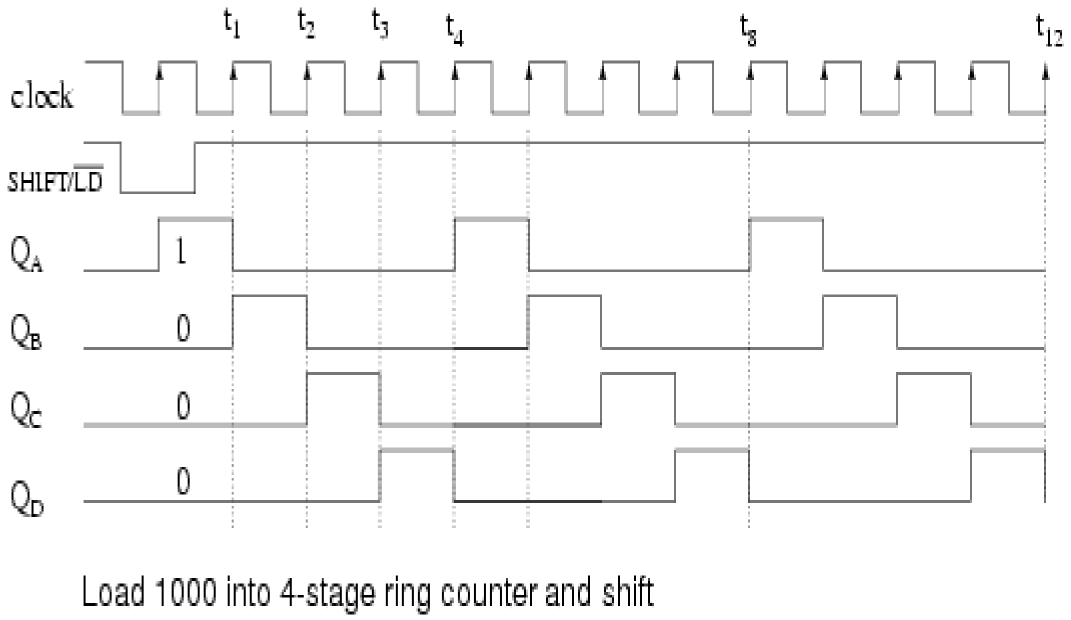


Fig4 : Ringcounter In Parallel In Serial Out Shift Register

Loading binary **1000** into the ring counter, above, prior to shifting yields a viewable pattern. The data pattern for a single stage repeats every four clock pulses in our 4-stage example. The waveforms for all four stages look the same, except for the one clock time delay from one stage to the next. See figure below.



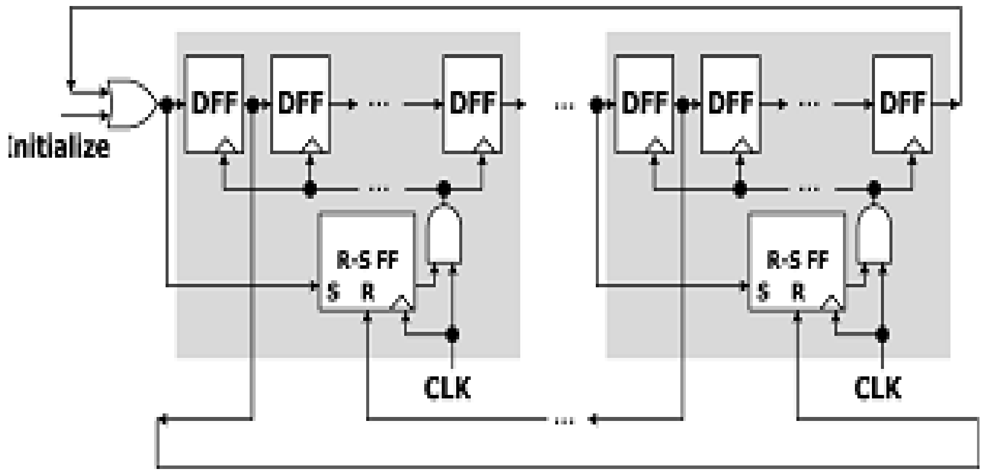


Fig5: Ring Counter With SR Flip-Flops

The above block diagram shows the power controlled Ring counter.First, total block is devided into two blocks. Each block is having one SR FLIPFLOP controller

DDR5: Double Data Rate 5 Synchronous Dynamic Random-Access Memory (DDR5 SDRAM) is a type of synchronous dynamic random-access memory. Compared to its predecessor DDR4 SDRAM, DDR5 was planned to reduce power consumption, while doubling bandwidth.[5] The standard, originally targeted for 2018,[6] was released on July 14, 2020.[2]. A new feature called Decision Feedback Equalization (DFE) enables input/output (I/O) speed scalability for higher bandwidth and performance improvement. DDR5 has about the same 14 ns latency as DDR4 and DDR3.[7] DDR5 octuples the maximum DIMM capacity from 64 GB to 512 GB.[8][3] DDR5 also has higher frequenciesJ than DDR4, up to 8GT/s which translates into 64 GB/s (8 gigatransfers/second × 64-bits/module / 8 bits/byte = 64 GB/s) of bandwidth per DIMM. Rambus announced a working DDR5 dual in-line memory module (DIMM) in September 2017.[9][10] On November 15, 2018, SK Hynix announced completion of its first DDR5 RAM chip; running at 5.2 GT/s at 1.1 V.[11] In February 2019, SK Hynix announced a 6.4 GT/s chip, the highest speed specified by the preliminar DDR5 standard.[12] The first production DDR5 DRAM chip was officially launched by SK Hynix on October. The separate JEDEC standard Low Power Double Data Rate 5 (LPDDR5), intended for laptops and smartphones, was released in February 2019.Compared to DDR4, DDR5 further reduces memory voltage to 1.1 V, thus reducing power consumption.DDR5 modules incorporate on-board voltage regulators in order to reach higher speeds.

RESULTS:

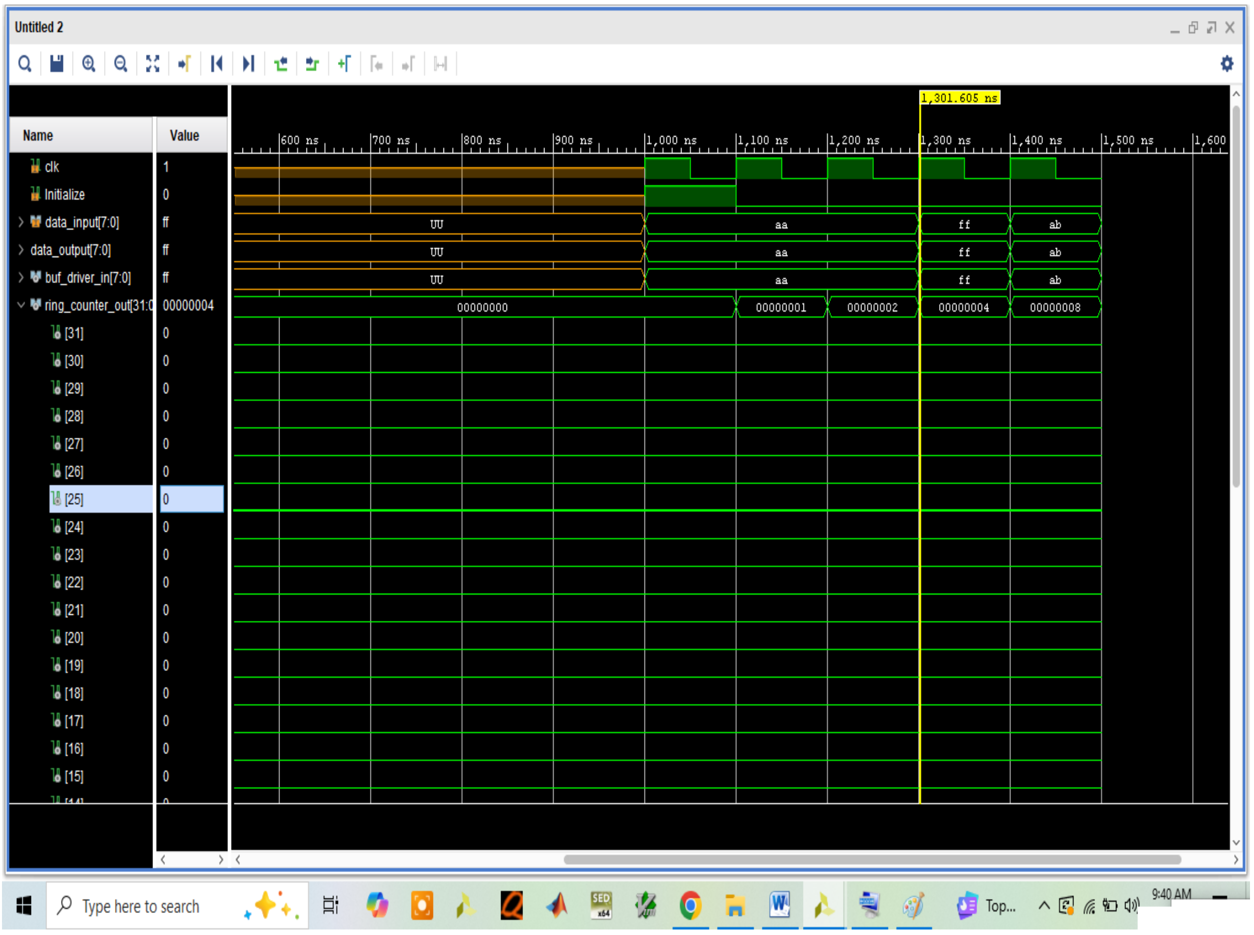


Fig a: Proposed simulation results

CONCLUSION: Based on observations of investigations that we can declare as follows: AXI-based high-performance DMAC can be considered a good alternative to SOC Design. Data transfer volume and time are important problems. This architecture is good Efforts to improve data transfer characteristics. DMAC has completed the second problem. This is highlighted with a ratio of two cases. As illustrated in all cases above, The recommended DMAC stands out to become superior to transfer data at high speed, for For example, multimedia transfer operations. The RTL Simulation of DMA controller has been verified and validated by suitable test benches. The Synthesis of DMA controller has been successfully completed by the extraction of Synthesized Net-list with unit delays.

Future Scope: Future progress of this work includes Recommended practices will be extended to other peripherals. Generation test bench at Sophisticated verification language to stimulate various connected peripheral modules.

References:

[1] Zayati, A. , Biennier, F., Badr, M.M.Y., 2012. Towards lean service bus architecture for Industrial integration infrastructure and pull manufacturing strategies. Journal of intelligent Manufacturing, Springer, Volume 23, Issue 1, pp 125-139.

[2] Barry, B., 1997. The Intel microprocessors Brey Architecture: Programming and Interfacing, Prentice-Hall international, Inc. Fourth Edition 1997. Pp-469.

[3] Jinbiao, Hou., 2013,. Study on PCI Bus and ISA Bus Conversion Design. Internasional Journal of Digital Content Technology and its Applications (JDCTA). Volume7 Number 4, doi:10.4156/jdcta.vol7.issue4.55.

[4] Li,Bo.,Peng, Yu., Liu, Da-Tong., and Peng Xi-Yuan., 2009., A High Speed DMA Transaction Method for PCI Express Devices. Journal of Electronic Science and Technology of China, Vol. 7, No. 4, Pp-380-84.

[4]ARM ltd., 2017. AMBA Trademark License, Obtained from http://arm.com/about/trademarks/arm-trademark-list/AMBA-trademark.php. Accessed o n October 8, 2017.

[5] Flynn, D., 1997. ARM, AMBA: Enabling Reusable on chip Designs. IEEE Micro- pp-20- 27.

[6]ARM., AMBA Specification. 2.0., 1999. URL. http://www micro.deis.unibo.it/~magagni/amba99.pdf. Accessed on December 17, 2016

[7] ARM, 2019. ARM Developer the Products category processors Cortex-A8, Last accessed on 16th Jan-2019.https://developer.arm.com/products/processors/cortex-a/cortex-a8

[8] Sinha, R., Roop, P., Sinha, S.B., 2014. Correct-by Construction Approaches for SoC Design, the AMBA SoC Platform, Springer, pp 22.

[9] Aljumah, A., Ahmed, M.A., 2015. Design of High speed Data Transfer Direct Memory Access Controller for System on Chip based Embedded Products. Journal of Applied Sciences, 15(3): pp. 576-581.

[10] Xilinx's., 2017. all programmable SoC evaluation kit ZC702, Zynq-700, Obtained from the URL, http://www.xilinx.com/products/boards-and-kits/ek-z7-zc70.html#hardware. Accessed on October 8, 2017.

[11] Roberts-Hoffman, K., Hegde, P., 2009. ARM Cortex-A8 vs. Intel Atom: Architectural and Benchmark Comparisons. University of Texas at Dallas EE6304 Computer A rchitecture Course Project – fall. Last accessed on 2nd Nov-2018.

[12] Abdullah, Aljumah., Ahmed, Mohammed, Altaf., 2016. AMBA Based Advanced DMA Controller for SoC. International Journal of Advanced Computer Science and A pplications. Vol. 7, No. 3, 188.

[13] Anand, S., 2013. Implementing a PCI-Express AMBA interface Controller on Spartan 6 FPGA. Chalmers University of Technology Sweden.

[14] Hou, J., 2013. Study on PCI Bus and ISA Bus Conversion Design. International Journal of Digital Content Technology and its Applications (JDCTA), Vol 7, Number 4, 55.

[15] Oded, Maler., 2012. Optimizing DMA Data Transfers for Embedded Multi-Cores. Thesis submitted to the Docteur De L'université De Grenoble.

[16] MengShengwei., 2016. Design of a PCIe Interface Card Control Software Based on WDF IEEE Xplore: Instrumentation and Measurement, Computer, Communication and C ontrol (IMCCC), Fifth International Conference.

[17] Mohammed, Altaf, Ahmed., Elizabeth, Rani, D., Syed, Abdul, Sattar, 2015. FPGA Based High Speed Memory Bist Controller For Embedded Applications. Indian Journal of S cience and Technology, Vol 8(33)