# AMBA Based Advanced DMA Controller for SoC

Abdullah Aljumah and Mohammed Altaf Ahmed
Department of Computer Engineering
College of Computer Engineering & Sciences
Prince Sattam Bin Abdulaziz University,
Al-kharj, Saudi Arabia

***Abstract*—this paper describes the implementation of an AMBA Based Advanced DMA Controller for SoC. It uses AMBA Specifications, where two buses AHB and APB are defined and works for processor as system bus and peripheral bus respectively. The DMA controller functions between these two buses as a bridge and allow them to work concurrently. Depending on the speed of peripherals it uses buffering mechanism. Therefore an asynchronous FIFO is used for synchronizing the speed of peripherals. The proposed DMA controller can works in SoC along with processor and achieve fast data rate. The method introduced significant volume of data transfer with very low timing characteristics. Thus it is a better choice in respect of timing and volume of data. These two issues have been resolved under this research study. The results are compared with the AMD processors, like Geode GX 466, GX 500 and GX 533, and the presence and absence of DMA controller with processor is discussed and compared. The DMAC stands to be better alternative in SoC design.**

***Keywords—FPGA; AMBA; DMA; DMA Controller; SoC; data transfer rate; FIFO***

## I. INTRODUCTION

Direct Memory access (DMA) design works with processor and reduced the load of it. DMA is a logical block to access the data of peripherals and easily to understand individually. But with other blocks and processor it is difficult to understand. DMA allows accessing the data easily without the involvements of the processor, from the devices connected to computer. Therefore it makes accessing the device memory and system memory easily and allows the processor to work simultaneously on its own job while on-going operations of memory usage are carried-out by externally connected devices. Doing this DMA made system performance to boost by allowing processor to perform more other task. Many hardware entities like desk drive controllers, sound, graphics and network cards are using DMA in many systems. It plays an important role in computers to access system memory directly. Similarly it plays an important role in embedded systems. DMA becomes an important unit of System on Chip (SoC) architecture. It offers significantly fast data transfer rate between memory and external devices connected to system [1]. The DMA performance enhances while using with the bus architecture [2].

The data is transferred between system memory and peripherals first time using the old Industrial Standard Architecture *(*ISA*)* bus by The Intel's first DMA 8237 first used in IBM PC in 1981[3]. Which supports four channel and capable to transfer data by 1.6 megabytes bitrate every second. Each single channel can address a full 64 kilo bytes part of memory and able to transfer 64 Kbytes in one programming [4]. The ISA bus and system bus were initially identical, and later separated by ISA Bridge. The IBM AT clones CPU works at higher frequency than ISA expansion therefore it is necessary to make them separate.

Another enhancement in bus architecture introduces, is in 1992 the Peripheral Component Interface (*PCI*) bus architecture. The concept of bus mastering introduces (first party DMA). It means a single device at a time can access the bus and known as bus master. If multiple devices have to use the same bus the arbitrary method is used [5]. Later in 1997 the packet switching concept with full duplexed mode for interfacing between system memory and multiple devices has introduces. The bus is then called PCI express (PCIe) proposed by Intel [6]. The packet switching is used and arbitrary logic replaced in the PCIe routings switches. PCIe provides background compatibility to PCI on driver levels. The PCIe increases the band width to almost double using x1 link pair for separate transmitting and receiving channels. On other side a very useful Bus architecture is going to start using by embedded products for DMA operations called advanced microcontroller bus architecture (AMBA) in System on Chip (SoC). AMBA is registered trade mark of advanced RISC machine (ARM) Ltd [7]. The first native AMBA interfaces are introduces in late 1997 with cashed cores [8]. AMBA on SoC is an on-chip interconnects specification for connecting and managing various functional blocks. It supports multi-processor designs with immense numbers of controllers and peripherals. AMBA specification defines two buses named AHB and APB and is an open standard.

Currently, AMBA is extensively used in Application specific Integrated circuit (ASIC) or SoC based portable recent mobile devices for example smart phones. DMA controller and embedded processor in SoC have close relation. The performance of processor will effected drastically for not integration DMA along with it [1]. Like a human body is a system and it can perform many tasks by much subsystem. Similarly embedded system comprises multiple functions. All these functions may perform by one or more processors in any embedded system, but the question here is that, if processor is involve in doing all transfer task, how will be the performance effects. It will be busy most of the time in sharing / transferring data. For this purpose the better alternative is DMA to improve the performance and avoid the extra burden on processor.

The DMA method for data transferring along with processor in FPGA (Field Programmable Gate Array) having

Prince Sattam Bin Abdulaziz University, Saudi Arabia.

many advantages. In this approach, the specific peripherals can be selected based on applications. The memory blocks, memory controllers, buses, peripherals and peripheral controller are easily added with the embedded processor system, as a result the system become progressively more impressive and useful[9]. The FPGA based method works on the principle of transmitting and receiving DMA for writing and reading respectively. It consists of two controller state machine separately, works to achieve a moderate transfer rate [10].

In the proposed method we implemented a DMA controller for embedded applications. It uses AMBA specifications and can be used in SoC based design. This DMAC is an attempt to improve data transfer characteristics. The transfer rate is achieved significantly better as compared to AMD and ARM processors used in embedded applications. It offers advanced features while keeping the very low gate count. It keeps processor in an idle state and use DMA controller to transfer volume of data. The Virtex7 FPGA is used to implement the design [11]. Two case studies have presented to compare the performance of DMAC with processor. The study examines the techniques for optimizing performance and cost in embedded system.

## II. Methodologies

### A. principal operations

Direct memory access is method in which the data will transfer between main memory and connecting peripherals and vice versa without the involvement of processor. The principle to access data from main memory by input output device independently, DMA transfer method is shown in Fig 1. DMA can perform coping (transfer) data form intra-chip in multi-core processor and memory to memory.

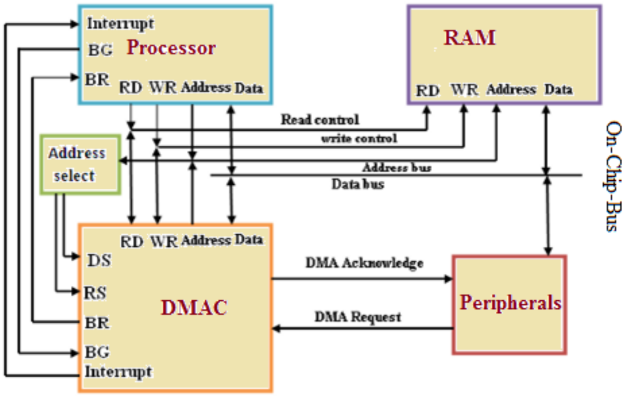


Fig. 1. DMA Transfer Methode

The logic that performs operations like addresses generation and reads writes, called DMA Controller (DMAC). The processor works in low state and configures the DMAC for data transfer, and carry on with its own task. Once processor grants the system bus, all the operations of peripherals are performed by DMAC.

Data transfer operations begins when DMAC send the bus request (BR) signal to processor to relinquish control of the bus. In response to BR, The processor first completes the task in hand and sends the bus grant (BG) signal. After receiving BG signal, the DMAC get control on system bus and generate essential signals to perform data transfer operations. DMA controller and the memory are address by the address bus. The register selects (RS) and device select (DS) lines are activated by addressing DMA controller and are perform read or write operations at the selected memory location. The acknowledge line is set by DMA when the system starts transfer operations. Data bus transfers data between external peripherals and memory. The interrupt signal is used to inform the processor about the termination of transferring data operations. The proposed method works on the similar principle.

### B. Proposed Architecture

This research study proposes the DMA controller which works on AMBA specification. The AMBA based system usually consists of Advanced High-performance Bus (AHB) and Advanced Peripheral Bus (APB). Therefore a bridge between these two is required for transfer operations and called AHB-APB bridge, this AMBA System shown in Fig 2. AMBA is open standard for 32-bit embedded processor and used in various SoCs as an on-chip system bus.

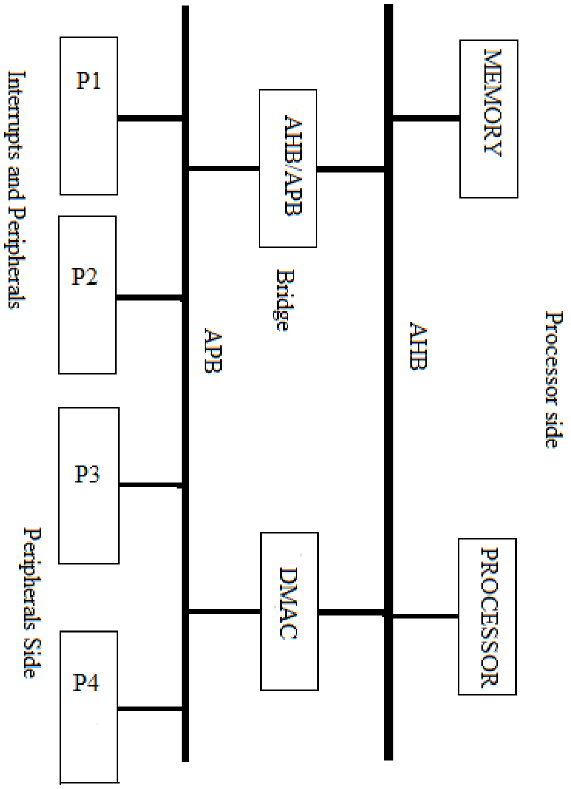


Fig. 2. AMBA System

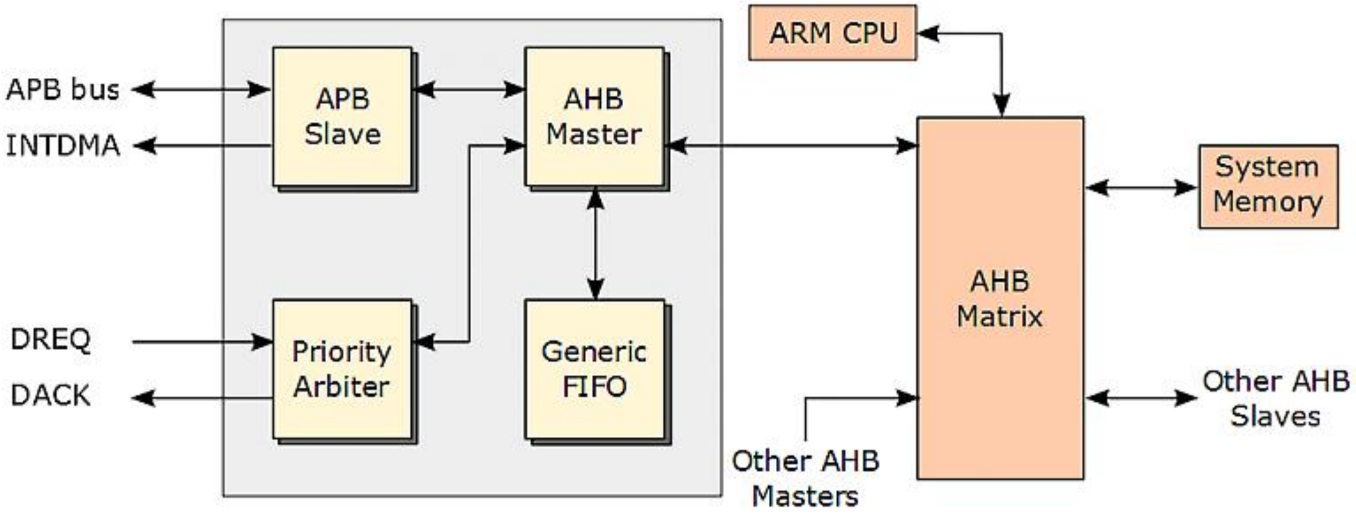


Fig. 3. AMBA Based DMA Controller for Embedded Applications

Now a day it works the de facto standard for SoC design [9]. It delivers variety of transactions like single and burst transfer, in which the single and several data packets are transferred respectively. The AMBA system usually consist of various components like memory (on chip RAM), processor, AHB/APB Bridge and peripherals like Interrupt ,UART, Timer and GPIO(General-purpose input/output).

### C. System Bus and Peripheral Bus

There are two system buses are contributes while transfer operation carried out. First one is AHB and second one ASB (Advanced System Bus). The AHB supports up to 128 bits transfer and used in high performance systems. It can support both single and multiple bus masters. It consists of master, slave, decoder and arbiter. It is available first time in AMBA 2 and later it is upgraded in AMBA 3. The simple AMBA system consists of one or more masters connected to one or more slave devices. The multiple master configuration uses arbitrator logic to choose one among several at a time. The One master multiple slaves configuration is called AHB-Lite.

The Advanced System Bus (ASB) is also a high performance bus. It also uses arbitrator for multiple masters. It is a synchronous bus, allows only one master at a time. ASB support a rich feature of pipeline where address and data transfer take place in parallel.

On the other side The peripheral bus known as APB(Advanced Peripheral Bus) is a low performance bus used to connect the peripherals to system bus of SoC. APB is interfaced with system bus (like AHB) through bridge called AHB-APB bridge for transfer operations. It allows the AHB master to address the slave on the other side of peripheral. It guaranteed about the connection but between master and slave but not ensure of correctness.

The Functional block AMBA based DMA controller for embedded applications shown in Fig 3. It consists of ARM processor, system memory, a generic first in first out (FIFO), priority arbitrator, AHB matrix and APB slave. This DMA supports AMBA AHB-Lite. It support memory to memory, memory to peripherals and peripheral to memory transfer. This DMA Controller mainly divide in two part one side is AHB and other side is APB. DMA usually can work in one master and many slaves mode (like AHB-Lite protocol) and multiple masters and multiple slaves' mode. For this design it works as one master and multiple slave mode and can be enhance to multiple master mode in future. Therefore it is compatible with AHB-lite protocol where one master with multiple slaves' are used and the configuration is shown in Fig 4. This configuration not uses the arbiter block to select master. It contains a decoder, which decides master requests into slave triggering signals. Additionally, if there are multiple masters, an arbiter is included which decides which master gets to access the bus at any time. AHB masters can carry out burst transfers where multiple data elements are read/written from/to a slave in a particular transaction.

Multiplexer is used to confirm that only one slave from several can access the data bus at a single time. Decoder is used for selection of the slave to perform transfer operation from several. It selects the slave and set the select input of multiplexer simultaneously to select respective slave to read

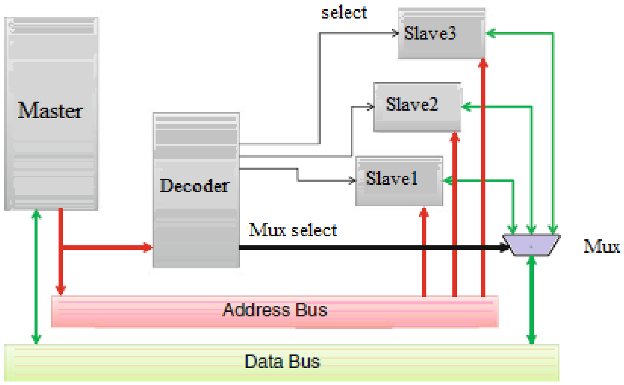


Fig. 4. One Master Multiple Slave Configuration

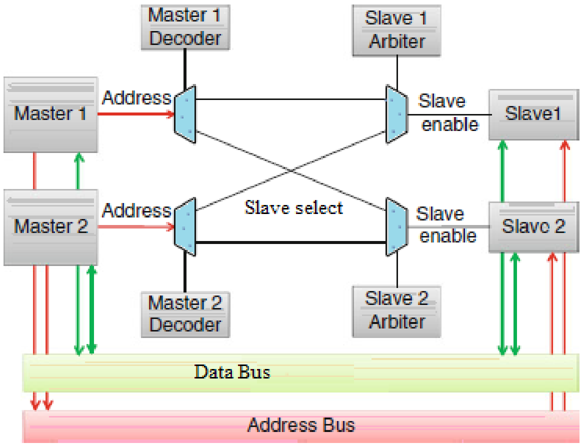


Fig. 5. Multiple Masters and Multiple Slave Configuration

from and write to data bus. DMA can also works in multiple masters and multiple slave mode, this configuration is shown in Fig 5.

This mode of operation uses arbiter logic to select one from many masters. Multiple masters can request to arbiter but arbiter grant request to only one. This method offers pipeline protocol, where address transfer, data transfer and arbitration can take place instantaneously.

DMA operation starts when the processor configures the DMA, AHB master catches the access to the bus when the arbiter grant the request of an AHB master. After receiving access the AHB master completes the data transmission between AHB and FIFO. The APB master request to achieve the control of APB, It will gets access to the bus and complete the data transfer between APB and FIFO, after arbitration completes transfer with the help of the APB bridge. The APB and AHB operations are carried out independently; therefore DMAC could accomplish these two operations concurrently.

## III. Results and Comparison

The AMBA based Advanced DMA controller for SoC is implemented in Verilog hardware descriptive language (HDL). The simulation and synthesis are performed by Modelsim and Xilinx tool respectively. The simulation process is automated for design scheme for various test conditions under writing and reading operations, and the respective wave forms are represented in Fig 6. The transfer processes are executed upon the access to bus is approved. The hardware is synthesized, when functional verification found to be error free by modelsim. Virtex-7 target device is selected with speed grade-4. The results are achieved by successful completion of synthesis process on FPGA.

It is quite obvious from the synthesis results that the design utilizes 168 LUTs at 476MHz of maximum frequency in 2.10 nsec times. These values are tabulated in TABLE I.

The results point out to have gained maximum frequency to be 476 MHz or 476,000,000 cycles/sec, which meant that the DMAC is able to transfer 1904 megabytes data per second [4*476 M = 1904 megabytes/sec].

The performance results are examined by comparing the existing embedded processor of AMD. The embedded processor used in SoC such as AMD Geode GX 466 works on 333 MHz frequency, Geode GX 500 at 366MHz and Geode GX 533 on 400 MHz [12]. Therefor data transfer rate of these processors with 32 bit data width results in 1332M bytes [4*333M = 1332 megabytes/sec], 1364M bytes [4*366M = 1364 megabytes/sec] and 1600M bytes [4*400M = 1600 megabytes/sec]. The comparison table is shown by TABLE II.

The data transfer rate, clearly much better in case of proposed DMA controller when compared to above three illustrations. The comparison is plotted in the graph of Fig 6.

There are few latest processor work on high frequency when compared to DMAC, obviously transfer more bytes in respect of time. However DMAC transfer more significant volume of data. Where these processors failed to maintained the same rate of transfer for large volume of data. As number of cycles increases there transfer rate decreases. The DMA functional verification is shown in Fig 7 indicating read and write operation of DMA.

Another comparison is plotted in Fig 8 for the rate of data transfer with and without DMA along with processor. Using the CPU consumes more instruction cycles to copy the data and it generally consumes more power. Instead DMA can perform the same task without processor. This way DMA is used to optimize the power and speed. Thus it is quite cleared from this comparison DMA can play a significant role in SoC for data coping. The DMAC is pleasantly incorporating significant volume of data transfer. This not only benefit through high volume data transfer, but also increased transfer rate as well, consumes low power and reducing the stress on the processor is another impact.

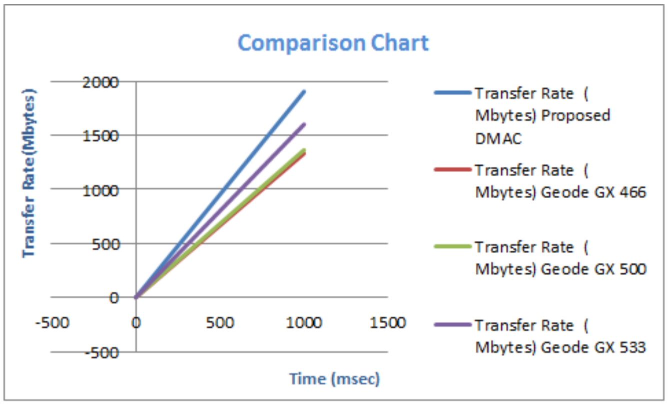
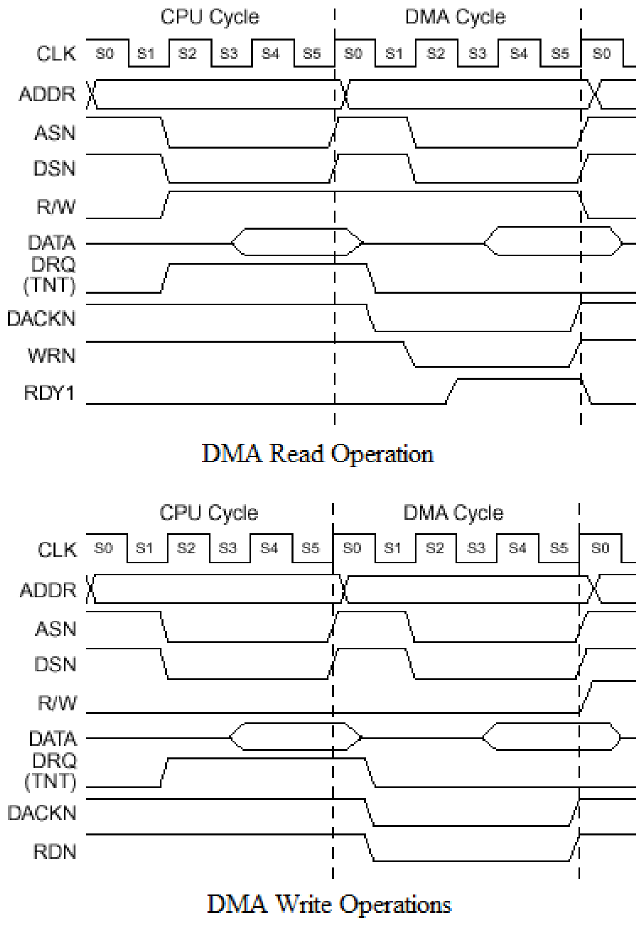


Fig. 6. Performance Comparison between DMAC and AMD Geode GX Series Processors GX 466,500 and 533



DMA Read Operation

DMA Write Operations

Fig. 7. DMA Read and Write Operation

## IV. Conclusion

The AMBA based Advanced DMA controller for SoC is supposed to be a good alternative to use in SoC based embedded design. This architecture is an effort to increase data transfer characteristics. The timing and volume of data transfer are serious complications. The DMA controller has fixed these two issues. This is emphasized with three cases of comparison.

After preparing out the characteristics of these examples, the proposed DMA controller outlooks to be better alternative for high speed data transfer in innumerable application fields such as multimedia processing. Future improvements include the application of the proposed practice to various multi-processor cores where speed and power are desired.

TABLE I. Synthesis Results gor DMAC

| Design | *Max Ferequency (MHz)* | *Time (nsec)* | *LUTs* | *Area Utilisation* | *Cycles per Sec* |
|---|---|---|---|---|---|
| **DMAC** | **476** | **2.10** | **168** | **>1%** | **476 mega** |

## Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript.

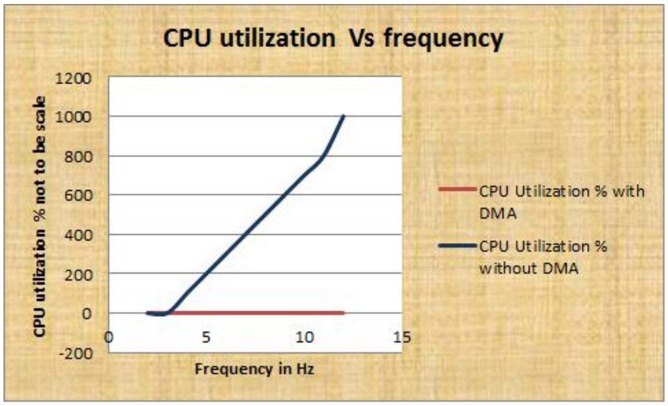


Fig. 8. Processor Utilization(instruction cycles) with and without DMA

TABLE II. Transfer Rate Per Unit Time

| **Transfer Time (msec)** | **Transfer Rate ( Mbytes)** | | | |
|---|---|---|---|---|
| | ***Proposed DMAC*** | ***Geode GX 466*** | ***Geode GX 500*** | ***Geode GX 533*** |
| 0.01 | 0.01904 | 0.01332 | 0.01364 | 0.016 |
| 0.1 | 0.1904 | 0.1332 | 0.1364 | 0.16 |
| 1 | 1.904 | 1.332 | 1.364 | 1.6 |
| 10 | 19.04 | 13.32 | 13.64 | 16 |
| 100 | 190.4 | 133.2 | 136.4 | 160 |
| 1000 | 1904 | 1332 | 1364 | 1600 |

## Acknowledgment

This project was supported by Deanship of Scientific Research, Prince Sattam bin Abdul Aziz University under the research project No. 2015/01/4320.

## References

[1] Sachin Gupta, Applications Engineer Sr, and Lakshmi Natarajan, Applications Engineer Sr, Cypress Semiconductor Corp. Published in EE Times Design (http://www.eetimes.com). Optimizing Embedded Applications using DMA, November 2010, pp-1-6.

[2] Ahlem Zayati , Frédérique Biennier,, Mohamed Moalla, Youakim Badr Springer Journal of Intelligent Manufacturing, "Towards lean service bus architecture for industrial integration infrastructure and pull manufacturing strategies", February 2012, Volume 23, Issue 1, pp 125-139.

[3] Lewis, Peter H. (1988-04-24). "Introducing the First PS/2 Clones". The New York Times. Retrieved 6 January 2015.

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

[5] Craddock, David (New Paltz, NY); Glendening, Beth A. (Poughkeepsie, NY); Gregg, Thomas A. (Highland, NY); Greiner, Dan F. (San Jose, CA) Computer Companies; "Pci Function Measurement Block Enhancements" in Patent Application Approval Process (USPTO

20150261715). Investment Weekly News (Oct 10, 2015): 1040. NewsRx LLC.

[6] Tao Jiang, Rui Hou, Jian-Bo Dong, Lin Chai, Sally A. McKee, Li-Xin Zhang and Ning-Hui sun Adapting Memory Hierarchies for Emerging Datacenter Interconnects. Journal of Computer Science and Technology 30(1): 97–109 Jan. 2015. DOI 10.1007/s11390-015-1507-4.

[7] AMBA specifications, Obtained under URL: http://www.arm.com/products/system-ip/amba-specifications.php, ARM ltd, 2015.

[8] David Flynn ARM, AMBA: Enabling Reusable on chip Designs. IEEE Micro- July-Aug 1997 pp-20-27.

[9] J. M. Weber and M. J. Chin, Using FPGAs with Embedded Processors for Complete Hardware and Software Systems , *Lawrence Berkeley National Lab, 1 Cyclotron Road, Berkeley, CA 94720* .

[10] Abdullah Aljumah and Mohammed Altaf Ahmed, "Design of High speed Data Transfer Direct Memory Access Controller for System on Chip based Embedded Products".Journal of Applied Sciences, 15(3): Jan 2015, pp. 576-581.

[11] Mohammed Altaf Ahmed, D. Elizabeth Rani and Syed Abdul Sattar. FPGA Based High Speed Memory BIST Controller For Embedded Applications , Indian Journal of Science and Technology, Vol 8(33), DOI: 10.17485/ijst/2015/v8i33/76080, December 2015 ,ISSN (Print) : 0974-6846 , (Online) : 0974-5645

[12] White Paper AMD Geode™ GX and LX Processors Typical CPU Core Power Consumption Determination, 32077C - April 2006. (Specifications of AMD Geode).