DMA CONTROLLER FOR SYSTEM ON-CHIP BASED APPLICATIONS

K.Rama Rao[1], B. Naga Phanidhar[2], G.thirupal[3], B.Leela Srikanth[4.]
[1]Assistant Professor,Deparment of ECE,ALIET,JNTUK,Vijayawada.
[2,3,4,]UG Students,Department of ECE,ALIET,JNTUK,Vijayawada.

Abstract-In this paper,we are going to design the functionality of Direct Memory Access (DMA) Controller in Verilog. Direct memory access (DMA) is a feature avaliable in modern computers which allows certain subsystems of computer hardware to access system memory for reading and/or writing independently without the involvement of Central processing unit. Computers that are equipped with DMA channels is capable of transferring data to and from devices with less CPU load than the computers that does not have DMA channels.Data is transferred from one Memory-to another memory.we use the various design techniques for developing Intel 8237 DMA ip core.In this project,we will be trying to use various modelling techniques for designing ip cores.So that it can be used in most of the power level circuits and processors. Ip cores are used in ASIC's because it leads to control power, speed and size. These IP core forms the main part of the embedded circuits and control the working of the circuit and process of high speed data transfer rate and much suitable in SOC products.We are trying to achieve the maximum frequency of 306.24MHz and with a minimum time period of 3.265nsec.Synthesis done on Xilinx tool Xilinx 14.7 ISE synthesizer.

Keywords—*DMA Controller; IP Core; SOC.*

I. INTRODUCTION

Many system-On-chip integrated circuits contain embedded cores with different scan frequencies. Many IP core are design software like Xilinx, Modelsim,and Leonardo Spectrum, etc. which can used to design IP core like DMA, Interrupt Controller etc. These IP core can be Power aware an Implement on SoC by choosing different design technique and various modelling techniques .These all modelling technique and tools like Xilinx ISE also provide RTL view which will help to make IP cores to use in any Processor design. Today's SoCs are omposed of a wide variety of modules, such as microprocessor cores, memories, peripherals, and customized blocks directly related to the targeted application. To effectively perform simulation-based design verification of peripheral cores, it is necessary to stimulate the description in a broad range of behaviour possibilities, checking the produced results. Different strategies for generating suitable stimuli have been proposed by the research community to functionally verify these modules and their interconnection when embedded in a SoC's. Direct Memory Access allows devices to transfer data without subjecting the processor a heavy overhead. Otherwise, the processor would have to copy each piece of data from the source to the destination. This is typically slower than copying normal blocks of memory since access to I/O devices over a peripheral bus is generally slower than normal system RAM. During this time the processor would be unavailable for any other tasks involving processor bus access. But it can continue to work on any work which does not require bus access. DMA transfers are essential for high performance embedded systems where large chunks of data need to be transferred from the input /output devices to or from the primary memory.

II. PAST RESEARCH

In this literature survey,we came to know that olden computers have DMA controllers made of 8257 but we are performing DMA operation with 8237 which has better performance when compared with 8257.the programmable control and dynamic reconfiguration features are enhanced which lead to the change in data transfer rate remarkably.8257 has Is only a four channel device whereas 8237 is also a four-channel device which has a capability of expanding the DMA channels to any number from four channel inputs. The 8237 is capable of transferring data with the rates of up to 1.6 megabyte per second.Each channel has a capability of addressing 64k-byte section memory and can transfer up to 64k bytes with a single programming.

III. DMA CONTROLLER

The basic idea of creating DMA controller is to transfer blocks of data directly between memory and peripherals without the involvement of CPU. The data transferring does not use microprocessor but uses only data bus .Normally when transfer of data is initiated it takes up to 29 clock cycles.with this DMA transfer mechanism data transferring requires only 5 clock cycles.The present day DMA,s the capability to transfer as fast as 60 M byte per second.The rate of transfer is limited by speed of memory and its peripheral devices.Various peripherals can be interfaced with DMA. With need of speed in these years DMA has became an essential feature of all modern computers and gadgets.So that it allows peripheral devices to communicate without subjecting the CPU to a heavy load.If not the CPU has to copy each and every bit of data from source to destination.DMA copies a block of memory from the RAM or from any buffer to the device. DMA is essential to high performance of embedded systems.

Device which is wishing to perform DMA transfer has following bus request signals.

A. The Processor first completes the bus cycle which is presently executed and then issues the bus grant signal to the device.

B. The device then sends the bus grant acknowledgement signal.

C. The processor senses in the change in the state of bus grant acknowledgement signal and starts listening to the data and address bus for DMA activity.

D. The DMA device performs the transfer from the source to destination address.

E. During this transfer, the processor monitors the addresses on the bus and checks for any modification in location during DMA operations is cached in the processor. If the processor detects any cached address on the bus, it can perform any one of the two actions:

Processor updates the internal cache when a DMA write is detected.
Whenever the DMA operations are completed, the device releases the bus by giving a bus release signal.
Processor acknowledges the bus release and resumes its bus cycles from the point where it left off.
The DMA I/O mechanism provides direct access to the memory when the processor is temporarily disabled.DMA controller borrows the address bus, data bus, and control bus from the microprocessor temporarily and transfers the data bytes directly from I/O port to a series of memory locations or vice-verse.DMA transfer is also used to do high-speed memory-to memory transfers.In a microprocessor-based system,two control signals are used.One for request and other to acknowledge a DMA transfer.The HOLD signal is a bus request signal which asks the microprocessor to release control of the buses after the executing the current bus cycle. The HLDA signal is a bus grant signal which indicates that the microprocessor has released control of its buses by placing the buses at their high-impedance states. The HOLD input has a higher priority when compared with INTR or NMI Interrupt.

IV. METHOD OF IMPLEMENTATION

There are three cases of data transfer.

A. From a peripheral interface to the external memory

B. From the external memory to a peripheral interface

C. From one memory location to another.

Data transfer from a peripheral interface to the external memory: When data has to be transferred from one peripheral interface to external memory,data is first stored in a 64 byte FIFO buffer.When this buffer is filled up to half of its size, i.e.32 bytes, the DMA controller is notified and the data in the FIFO is transferred to the external memory. If any one of these memory addresses is recently used by the CPU,then the data is stored in

Operation steps of DMA Controller :

Processor invalidates the internal cache entry for the address involved in DMA write operation (or)

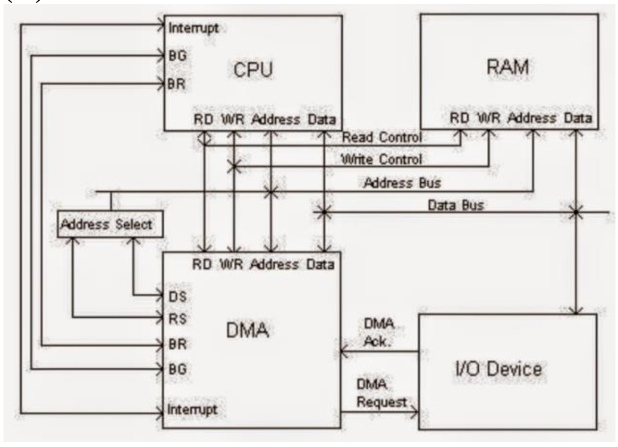

Fig.1:block diagram of DMA controller

the internal cache memory as well.Data transferred from a FIFO to the memory in large amounts and is known as burst transfer,the length of each packet in burst transfer is either 16 or 32 bytes as chosen by the software programmer in an internal register.
Data transfer from the external memory to a peripheral interface:
When the DMA controller receives a signal that a FIFO buffer is getting emptied and almost half is empty, i.e. less than or equal to 32 bytes, it begins a transfer of data from the external memory. When data is transferred from the external memory it is read in bursts of 16 or 32 bytes (as chosen by the software programmer). If one of the memory addresses where the data is located has recently been in use by the CPU, the data is read from the internal cache memory.
Data transfer from one external memory location to another: In order to increase the performance of the DMA, as with previous transfers the data is read from the external memory when the FIFO buffer is becoming half empty. The data is either read from, or written to the cache - depending on the direction of the data flow - if one of the memory addresses have been in use by the CPU recently.

V. SYSTEM ON CHIP

In modern days SOC (System on Chip) design has a high level of integration consisting of several design components (which are also known as IP -Intellectual property).It is possible with the shrinking process technologies. In other words, a SOC is truly an IC which can implement almost or all the functions of a complete electronic system.
A typical SOC design may contain one or more programmable components such application-specific

intellectual property (IP) cores, digital signal processor cores,general-purpose processors cores, along with analog front end,IO devices,on-chip memory and several other application-specific circuits.

The major challenge in SOC design is on chip communication between the different components.Different bus protocols are used for interconnection which has considerable large impact on the performance of the SOC designs
supporting high performance and low power on-chip communication.Encourage modular system design to improve

VI. AMBA -AHB PROTOCOL:

The Advanced Micro-controller Bus Architecture (AMBA) is a protocol which is used as an open standard bus protocol;on-chip interconnects the specification of connection and the management of functional blocks in a system-on-chip (SoC). The AMBA bus can be easily applied to small scale SoCs. Therefore, the AMBA bus has been the representative of the SOC market with bus efficiency.
The specifications of AMBA define all structural configuration,the transfer modes,signals, and other bus protocol details for the AHB, APB, and AXI buses. The AMBA AHB is used here to interface with any peripherals which are of low bandwidth and does not require high performance of the pipe lined bus interface.

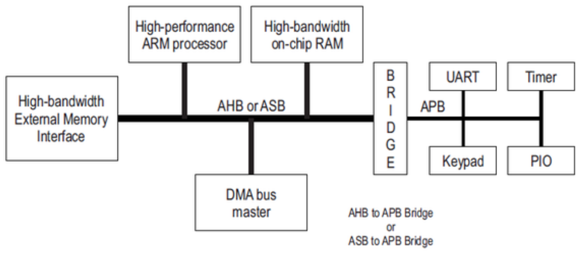
High-performance ARM processor
High-bandwidth on-chip RAM
High-bandwidth External Memory Interface
AHB or ASB
B R I D G E
UART
Timer
APB
Keypad
PIO
DMA bus master
AHB to APB Bridge or ASB to APB Bridge

Fig.2: AMBA-AHB bus interface

AMBA is a high-bandwidth,high-speed bus that support multi master bus management to get the best experience in terms of system performance. AMBA AHB bus acts as the backbone for high performance system bus in an System on Chip and its applications is provided by ARM CPU.Wide range of architectures of AMBA is used for providing flexibility in the implementation and backward-compatibility with existing AMBA interfaces.Separate read and write data channels are available that provide low-cost Direct Memory Access (DMA) and supports for issuing multiple outstanding addresses. It support for out-of-order transaction completion. It permits easy addition of register stages to provide timing closure.Objective of AMBA specifications is to: Facilitate right-first-time development of embedded micro controller products with one or more CPU's, GPU's or signal processors, are technology independent. To allow reuse of IP cores, peripheral and system macro cells across diverse IC processes.Minimize silicon infrastructure while processor independence, and the development of reusable peripheral and system IP libraries.In many cases, the IP cores
are designed with different interfaces and communication protocols as a result of this can be a problem while integrating into an SOC.To avoid this problem,some standard on-chip bus structures and protocols were used.There are Some publicly available bus architectures developed by leading manufacturers I.e. CoreConnect from IBM , AMBA from ARM , SiliconBackplane from Sonics.

VII. SIMULATION RESULTS

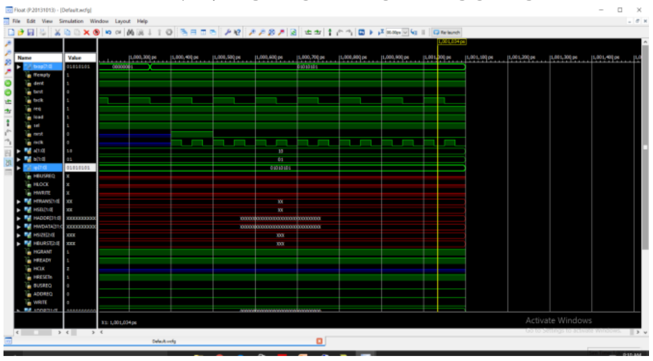

Fig.3:transmitter simulation waveform

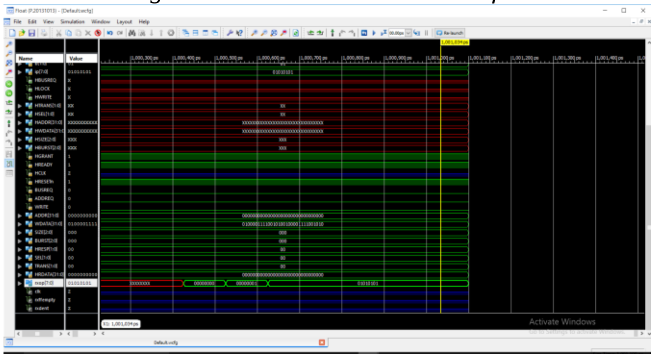

Fig.4:receiver simulation waveform

VIII. CONCLUSION

DMA design consists of asynchronous FIFO, DMA engine in both transmitted side (when peripherals want to write) and received side (when peripheral want to read) and a common dual port RAM. The whole design implemented in Verilog. HDL for easy integration in SoC. Simulation has done for individual blocks and a complete top-level block architecture using xilinx synthesis tool(version 14.7).The simulation result of AMBA based DMAC has achieved and indicates reading just after writing in a single clock.

REFERENCES

[1] Olugbon, A., S.Khawam, T.Arslan, I.Nousias and I.Lindsay, 2005. An AMBA AHB-based reconfigurable SoC architecture using multiplicity of dedicated flyby DMA blocks.
[2] Intel 8237 data sheet.
[3] ARM Ltd., 2007. PrimeCell® DMA controller (PL330) technical reference manual. ARM DDI 0424A, ARM Ltd., Cambridge, UK.
[4] Flynn, D., 1997. AMBA: Enabling reusable on-chip designs. IEEE Micro, 17: 20-27.

Hwang, K., 1993. Advanced Computer Architecture: Parallelism, Scalability, Programmability. McGraw-Hill, New York.

[5] Liang, J., S. Swaminathan and R. Tessier, 2000. ASOC: A scalable, single-chip communications architecture. Proceedings of the International Conference on Parallel Architectures and Compilation Techniques, October 15-19, 2000, Philadelphia, PA., pp: 37-46.

[6] Ma, G., 2009. Design and implementation of an advanced DMA controller on AMBA-based SoC. Proceedings of the 8th International Conference on ASIC, October 20-23, 2009, Changsha, Hunan, pp: 419-422.

[7] Asia and South Pacific Conference on Design Automation, Volume 2, January 18-21, 2005, New York, pp: 1256-1259.

[8] Tiwari, A. and D.J. Dahigaonkar, 2011. AMBA dedicated DMA controller with multiple masters using VHDL. Int. J. Inform. Technol. Knowledge Manage., 4: 285-288.

[9] Xilinx Corporation, 1998. Xilinx FPGAs: A technical overview for the first-time user. Application Note, XAPP 097 December 12, 1998.

[10] J. Liang, D. Swaminathan Flynn, "AMBA: Enabling Reuseable On-Chip Designs", IEEE Micro, vol. 17, no. 4, pp. 20-27, July/August 1997

[11] Guoliang Ma, Hu He, "Design and Implementation of an Advanced DMA Controller on AMBA-Based SoC", IEEE 8th International conference, pp. 419-422, 2009.

[12] Bogliolo, G. De Micheli, "A survey of Design Techniques for System-Level Dynamic Power Management", IEEE Trans. On VLSI Systems, vol. 8, no. 3, pp. 299-316, June 2000.