iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** III **Month of publication:** March 2018

DOI: http://doi.org/10.22214/ijraset.2018.3416

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Interfacing of High Speed SOC with Low Speed Components by using AHB Bus

Mr. Ravi Chandra Bathula[1], Mr. Sharath Chandra Inguva[2], Mr. P Ranjith[3],
[1, 2, 3,]*Assistant Professor, Dept. of ECE, Guru Nanak Institute of Technology, Hyderabad, Telangana, India*

***Abstract:** Recently, embedded systems design focuses on low-power dissipation and system-on-chip. Thus, ARM processor and AHB came to be popular in embedded systems. Whereas the bus has many advantages for high-performance modules, it has limitations on interfacing with low-speed devices. Therefore, a bridge is necessary between them. Although AHB-to-APB bridge is a standard in AMBA system, it is not useful because of the fixed interfacing speed. Moreover, it is difficult that the existing IP cores for the peripheral controllers are reused with APB. In this paper, we suggest more general and flexible bridge than the existing one for SoC.*

***Keywords:** ARM processor, system-on-chip, AHB, SOPC, data terminal equipment (DTE), data communications equipment (DCE).*

I. INTRODUCTION

A. Objective and Goal of The Thesis

UART is the most basic and most commonly used method of communication in the embedded system, whose performance determines whether the overall system can meet the design requirements. The implementation of UART basically uses the on-chip UART IP hard core or ARM currently. It makes the design of the whole system has great limitations for the parameters of which is solidify already on the chip(that they cannot be changed anymore)and which combine with the other on-chip peripherals that cannot be separated, although the performance is high. Because of the design beyond change, the poor flexibility, the small application, and the poor transportability, its usually unable to meet the high requirements of users

With the rapid development of SOPC (system on a programmable chip) and FPGA (field programmable gate array) and depending on the high performance, high flexibility, transportability and configuration soft core plays an increasingly important role in embedded system. Some companies (such as Altera) which have provided UART IP (intellectual property) soft core, supports only the poll and interrupt mode [3] currently. During the data transmission process, these two kinds of transmission will often interrupt the operation of CPU especially when transmitting large data, it will occupy a lot of time of CPU, thus it greatly reduces the performance of the overall system.

DMA (direct memory access) is a data exchange mode, where data is accessed directly from memory without accessing cpu. Efficiency of CPU can be improved by using DMA as the entire data transfer process is controlled by the DMA controller, the CPU can deal with other things without any disturbance. This UART IP soft core designed in this article is the adoption of the DMA mode, which greatly reduces the occupancy time of CPU and improves the performance of the whole system. As it makes the CPU free from the heavy work out to deal with other things while transmitting data. Design requirements can be better met because of its high-performance, configurable parameters, portability besides high flexibility.

Practically Data communication involves the distribution and exchange of information between people and machines. Therefore, the topic of Data communication hardware encompasses an extremely wide range of Data communication equipments includig both digital and analog devices. A Data communication system is comprised of three basic elements these are a transmitter (source), a transmission path (data channel) and a receiver (destination). The transmission path would be bi directional and the source and destination should be interchangeable for two way communications. Therefore it is usually more appropriate to describe a data communication system as connecting two end points. All end points must essentially have data terminal equipment (DTE), data communications equipment (DCE) and a serial interface.

B. Data Terminal Equipment (DTE)

Data terminal equipment can be any binary digital device that generates, transmits, receives or interprets data messages virtually. At data terminal equipment information originates or terminates essentially.

Data terminal equipment is a data communication equivalent to the person in telephone conversation.To establishes and control communication between the points in a data communication system data terminal equipment contains the hardware and software. However, data terminal equipments seldom communicate directly with other data terminal equipments. Examples of data terminal equipments include printers, personal computers and video display terminals.

C. Line Control Unit (LCU)

A device that performs all the data communication related functions virtually at a remote location. Line control unit directs the flow of the data traffic between the data communication channels and local terminals. The functions of Line control unit includes performing serial to parallel and parallel to serial data conversion, formatting data, inserting and deleting data link control characters, and also performs error detection and correction.

D. Front-End Processor (FEP)

A device that performs all the data communication related functions at a host location virtually. A FEP relives a host computer which is of slow process inherent with data communications FEPs serve as an interface between a host computer and all the data circuits that it serves. One FEP at host location can communicate with ic used at hundreds of remote locations.

E. Data Communication Equipment (DCE)

DCE is a general term which is used to interface data terminal equipment to a transmission channel, such as a digital analog telephone circuit. Depending upon the application, the output of a DTE can be digital or analog. DCE(Data Communication Equipment) converts signals from a DTE to a form more suitable to be transport over a transmission channel, a DCE also converts those signals back to their original form at the receiving end of a circuit so it is called as signal conversion device. DCE s is transparent devices responsible for transporting bits (1s and 0s) between DTE s through a data communications channel. The DCE s neither knows nor Cares about the content of the data. Within the FEPs and LCUs, there is a single special-purpose integrated circuit which performs many of the fundamental data communication functions. This integrated circuit is called as universal asynchronous receiver/transmitter (UART). A UART, a USRT and a USART are not the same and do not offer the same features. All three types of circuits specify general purpose integrated circuit chips located in an LCU or FEP that allow DTEs to interface with DCEs. UARTs, USRTs, USARTs are devices that operate external to the central processing unit in a DTE that allow the DTE to communicate serially with other data communication equipment, such as DCEs .In most modern computers ,USARTs are normally included on the mother board and connected directly to the serial port.

F. Motivation for the Implementation of UART with DMA

A universal asynchronous receiver transmitter is a circuit that sends parallel data through a serial line. UARTs are frequently used in conjunction with the EIA (Electronics industries alliance) rs-232 standard, which specifies the electrical, mechanical and functional characteristics of data communication equipments. The voltage levels defined in RS-232 differ from that of FPGA I/Os. In electronic design a semiconductor intellectual property core, IP core or IP block is a reusable unit of logic, cell, or chip layout design that is the intellectual property of one party. IP cores may be licensed to another party or can be owned and used by a single party alone. The term is derived from the licensing of the patent and source code copyright intellectual property rights that subsist in the design. IP cores can be used as building blocks within ASIC chip designs or FPGA logic designs. Since the present IP soft-core of UART is based on interrupt and poll method in which the CPU has to pay its attention during the data exchange process. And if the data is very large a lot of time of CPU will be occupied thus reducing the system performance. So UART with direct memory access controller is implemented in the project.DMA will place the role of CPU for the whole data transfer process, thus allowing the CPU to concentrate on other useful tasks.

G. Uart block diagram (with out dma)

The UART takes bytes of data and transmits the individual bits in a sequential fashion and also it takes the data in serial form and converts it into Bytes and sends to CPU. Hence this UART only supports poll and interrupt mode currently. In these two modes CPU takes initiative. In polling mode to detect whether the device is ready for its next operation, the I/O device status is polled at intervals. For this purpose CPU has to pay its attention till the end of the operation. So CPU cannot deal with other useful tasks, which degrades the performance of the system.

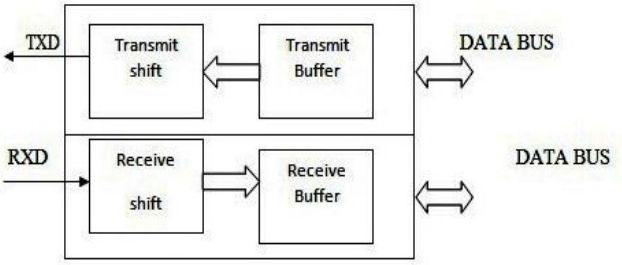


Figure 2.3 UART With Out DMA

Direct memory access (DMA) can greatly increases the polling-based system efficiency and hardware interrupts can eliminate the need for polling entirely. Both interrupt and polling modes can be eliminated by employing DMA in UART So this thesis presents the design of UART along With DMA controller. This UART differs from others due to the presence DMA read controller and DMA write controller. These two allows the CPU to concentrate on other things .i.e., CPU relinquishes operation data transfer.

H. UART Transmitter Signal Sequence

The operation of the typical UART transmitter shown in fig: 1.4 .However, before the UART can send data it must be enabled. Also the figure shows the signalling sequence that occurs between the CPU and UART. When the transmitter is ready to send data, it asserts a signal, txfree, transmitter free to the CPU to indicate that it can send data. Then the CPU enables the UART transmitter and then transmitter enable. Now the transmitter is ready for the serial transmission of the data through the serial line. Now the CPU places the 8-bit parallel data on the data bus

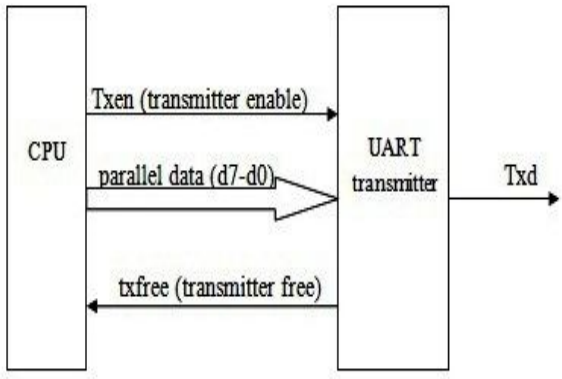


Figure 2.4 Transmitter Signal Sequence

Then the transmitter converts that parallel data in to serial bits and inserts start, parity and stop bits then transmits those bits serially on the transmission line txd. When all bits are transmitted successfully, again the transmitter indicates its readiness for next data transmission with txfree signa.

I. UART Receiver Signal Sequence

The receiving signal sequence is shown in fig: 1.4.Rxd is the serial reception line through which data is received bit by bit. The start and stop bits are removed and data is placed in internal buffers when receiver starts its reception. When the received data is valid then the receiver indicates the CPU, with the signal rxdv and sets it to high. Then the CPU makes the read signal high and collects the received data which is present in the internal data buffer of receiver. The receiver assembles the serial bits into 8-bit parallel data. This 8-bit parallel data is placed on the data bus and then CPU takes and writes this data to the memory. There may be an error in the received data in terms of parity or number of bits to indicate these errors receiver uses two signals namely parity error and receive error.

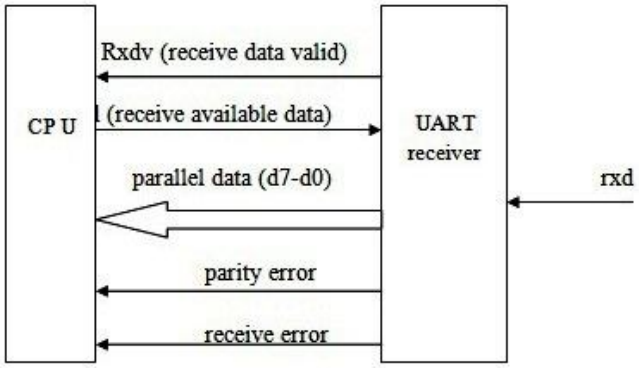


Figure 2.5 Receiver Signal Sequence

The parity bit will be received depending on the number of 1's present in the 8-bit data. Parity bit may be odd or even. Parity error signal will be high if the received parity bit is not coincided with that of transmitter. Then after the parity bit if the stop bit is not received then receive error will be one. Otherwise receive error will be zero indicating that no correct data has been received. So these are the signals that occur between CPU and UART receiver.

J. UART Block Diagram (With DMA)

The following figure shows UART block diagram without DMA. It consists of UART controller, DMA read controller, UART RX controller and DMA write controller.

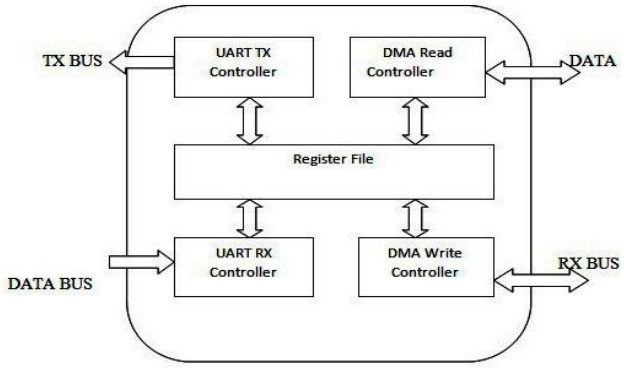


Figure 2.6 UART with DMA

K. Transmitter Block Diagram

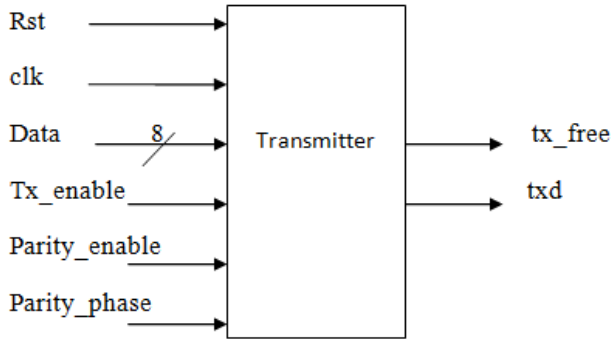


Figure 2.7 UART Transmitter

L. Transmitter State Transition Diagram:

When the process of sending the data in the transmit shift register is completed, the state machine will enter into the state of block finish. In this state, the state machine specifies the number of bytes of the data which has been sent out. If the number of bytes data that should be sent shows that all the data has not been sent out, then the count gets incremented by 1 and the state machine enters into the state of data valid to read and send the next byte of data

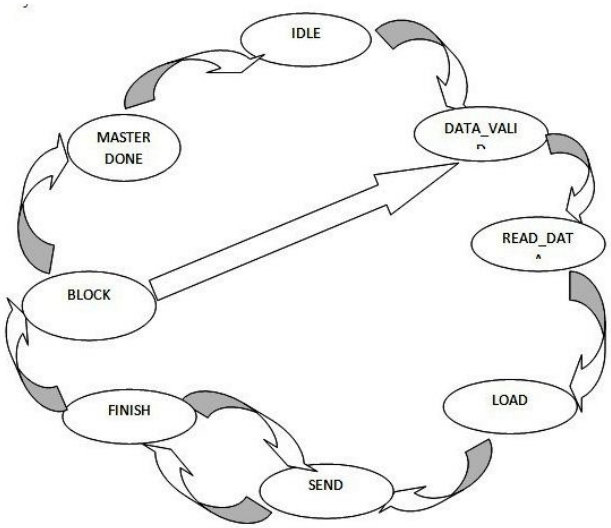


Figure 2.8 State Transition Diagram of the UART Transmitter

M. Receiver

When the receiver has received all of the bits in the data word, it may check for the Parity Bits (both sender and receiver must agree on whether a Parity Bit is to be used), and then the receiver looks for a Stop Bit. Regardless of whether the data received is correct or not, the UART automatically discards the Start, Parity and Stop bits. Then the receiver reassembles the serial bits into parallel data and loads into a data buffer. If the sender and receiver are configured identically, these bits are not passed to the host. If another word is ready for transmission, the Start bit for the new word can be sent as soon as the Stop Bit for the previous word has been sent. Next section describes the inputs and output signals of receiver.

N. Receiver Block Diagram

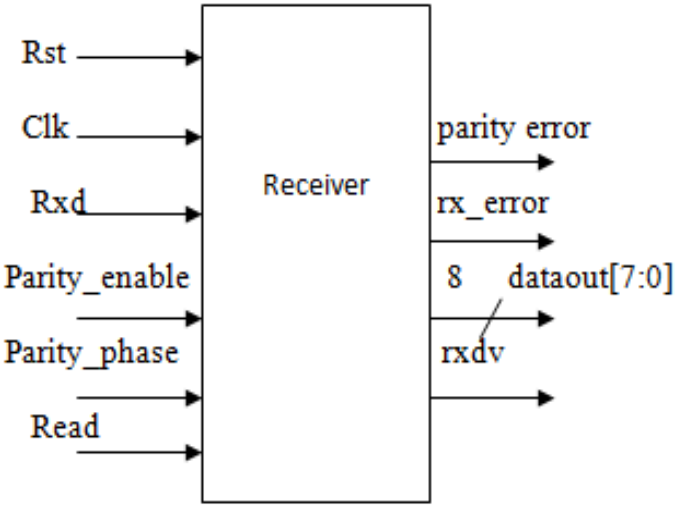


Figure 2.9 UART Receiver

O. Receiver State Machine

The reception of serial port uses the basic frame format. when the start-bit is detected at low-level then bytes of data is received bit by bit under the control of the clock in the baud rate. After the complete reception of all the data stop bit is detected at high-level. In this paper, a UART receiver controller is designed using the way of finite state machine in the hardware description language of verilog HDL, thus completing the timing control of the data reception of serial port.

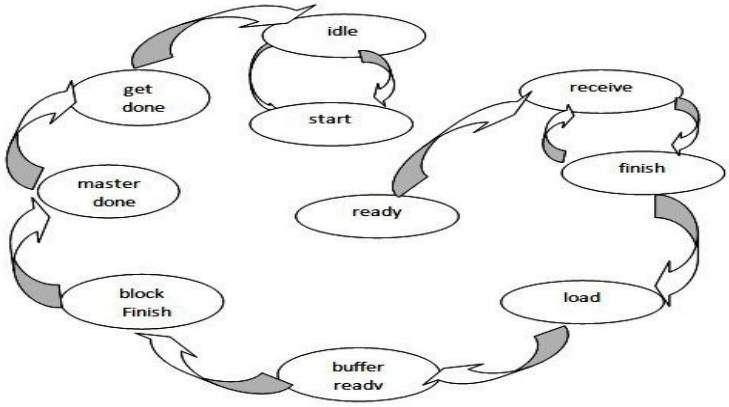


Figure 2.10 State Transition Diagram of the UART Receiver

P. DMA Block Diagram

Different inputs and outputs of the DMA from and to UART and memory are shown in the figure. Initially the DMA doesn't know from where it has to read or to where it has to route the data .Generally in order to make DMA to know the source address , destination address and transfer count CPU programs the DMA. Program_dma an input signal to the block, if it is high CPU programs DMA registers. A 2 bit signal reg_addr is provided as input, the source address is assigned when it is 00.Destination address is given to DMA by the CPU when it is 01, and finally if it is 10 transfers length is assigned. A input line reg_data holds 9 bit wide source address and destination address. After programming make sure that the input signal enable_dma is high, this is the signal given to DMA from CPU to activate the DMA .There is other signal called strat_dma which is input to the block which will initiate the data transfer. There are two 8 bit data lines, one is from memory to DMA i..e mem2dma_data other is from UART to DMA i..e uart2dma_data.

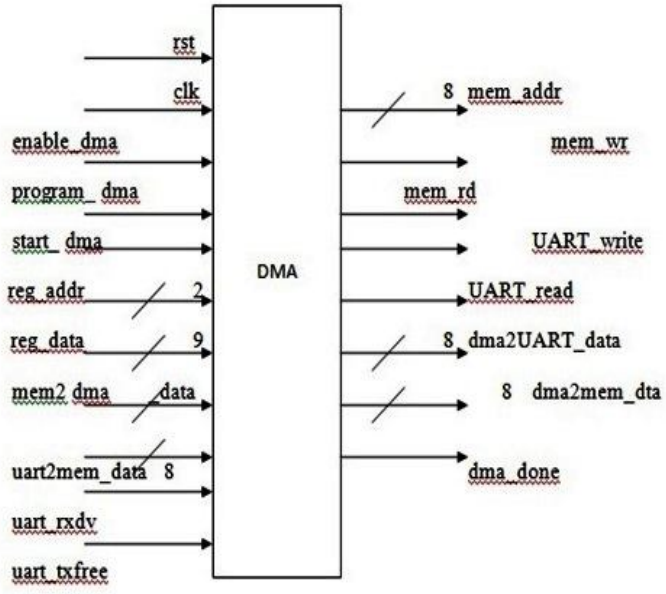


Figure 2.11: Block diagram of DMA

Q. DMA State Machine

When the processor wants to send data through serial port then it is necessary to make configuration to the UART sent controller and the Master Read type DMA controller through the register file with the interface of Avalon-MM slave to set the serial port, the number of bytes of the data to be sent and the address of the data are stored in the memory.FIG shows DMA state transition diagram.

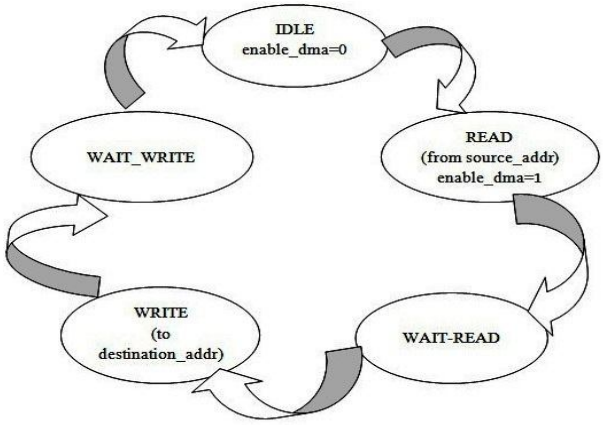


Figure 2.12 DMA State Transition Diagram

II. SIMULATION RESULTS

A. Logic diagrams

1) Memory: memory is a low speed device and it is interfaced with high speed SOC by using ahb bus.

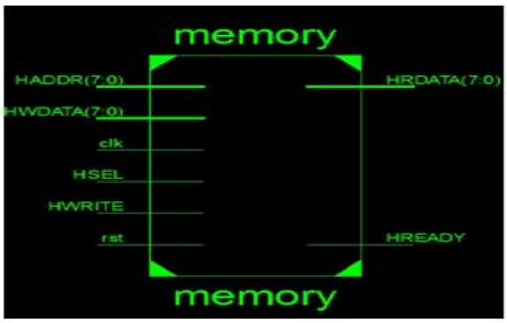


Figure 5.1: Logic diagram of 8bit memory

➢ **Direct memory access:** DMA is used to access memory directly

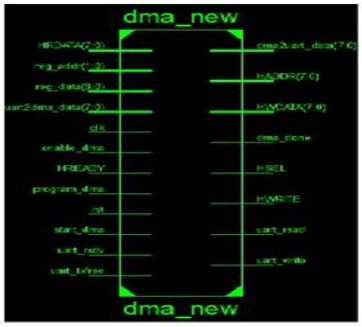


Figure 5.3: logic diagram of DMA

2) UART transmitter: UART is a universal asynchronous receiver and transmitter. And it is a low speed device. Following figure shows logic diagram of uart transmitter

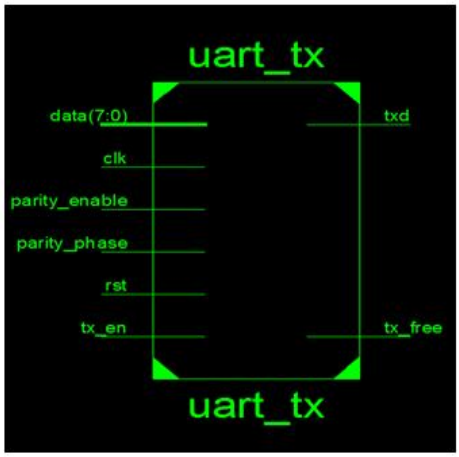


Figure 5.2: logic diagram of UART transmitter

- **UART receiver**: UART is a universal asynchronous receiver and transmitter And it is a low speed device. Following figure shows logic diagram of UART receiver.

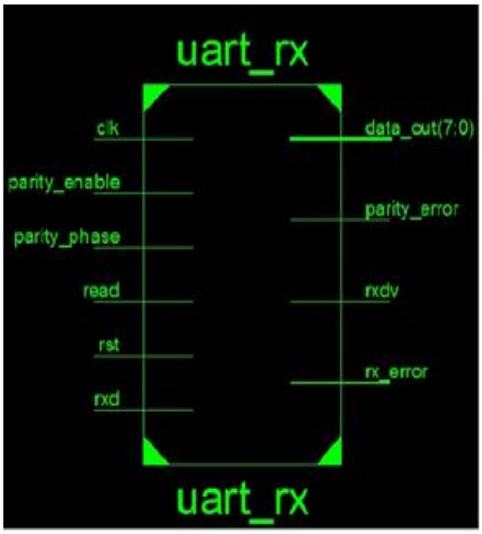


Figure 5.4: logic diagram of UART receiver

III. CONCLUSION

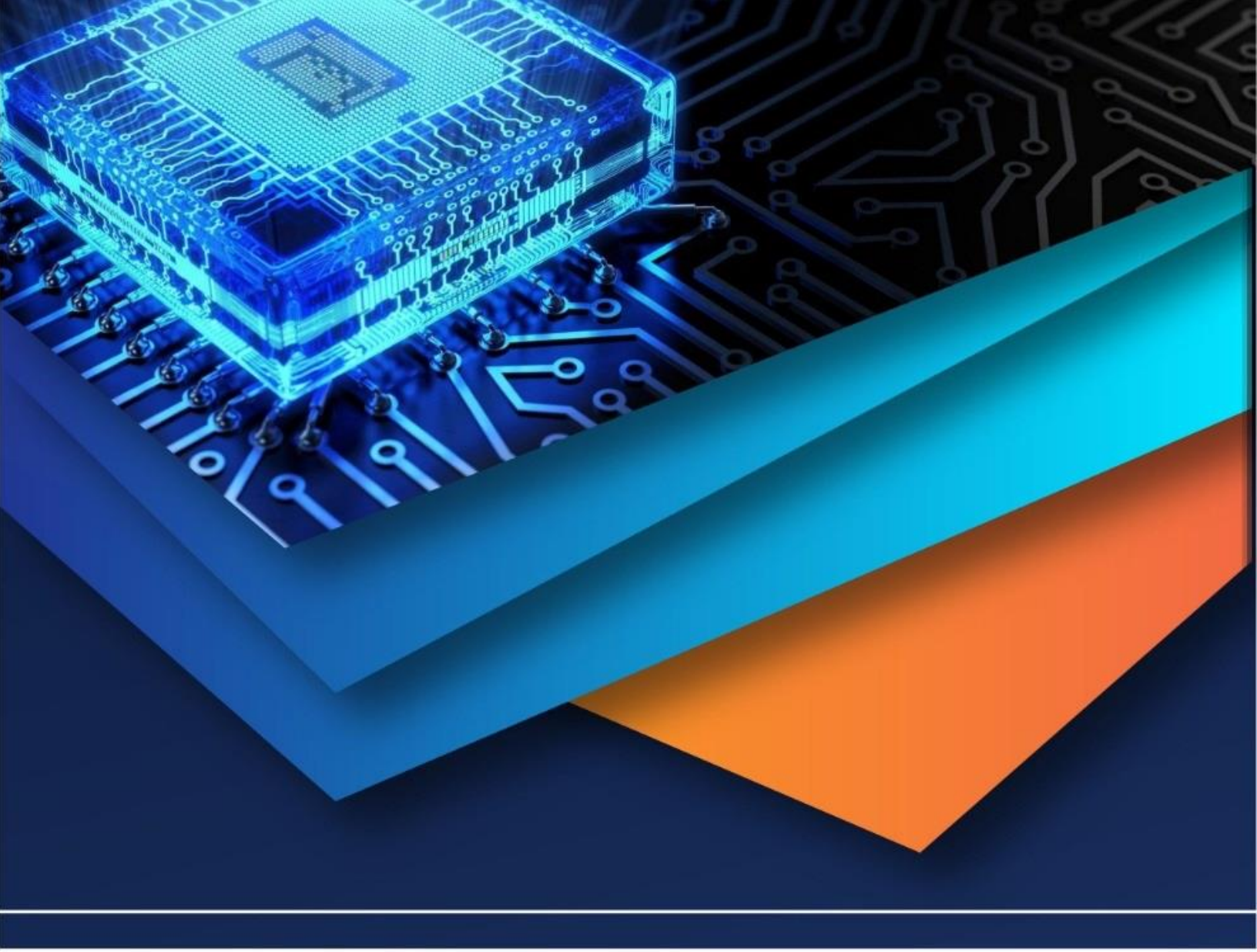
Existing system was running on ASB bus which is having drawbacks of handling high performance device with low speed peripherals, so only we design the AHB with master DMA controller and slave as UART bus protocol.
In this project proposed DMA(direct memory access) is having specifications of burst mode, parallel operation, function of APB(Advanced peripheral bus) ,no non –buffer mode, high transfer rates because of these specifications it is used in this project.
Ultimately this is one of the real time example of CPU interfacing with modem printer.

A. Advantages

1) Performance of the system is improved by transferring the data directly between I/O devices and memory
2) CPU performs free operations that doesn't use any system buses
3) Low cost communication

REFERENCES

[1] L. K. Hu and Q.CH. Wang, "UART-based Reliable Communication and performance Analysis , Computer Engineering, Vol 32 No. 10, May 2006, pp15-21
[2] Yang fuguang. Efficient UART communication and its applications based on DMA in ARM. Chinese Academic Journal Web Publishing General Library, 2008.
[3] Altera Corp, UART Core Users Guide
[4] http://www.freebsd.org/doc/en_US.ISO8859-1/articles/serial-UART/index.html
[5] http://www.asic-world.com/examples/verilog/UART.html
[6] http://opencores.org/
[7] http:// www.amba.com.
[8] Lingge Jiang,"Theories and Methods for Reactive Energy Measurement", Zhangjiajie Power Bureau, Zhangjiajie 427000, China.
[9] Wei Wang, Xiao Wang, Xiaoqing Huang, and Da-peng Xie The research on Hilbert digital filter of reactive power measurement? Southwest Jiaotong University, Chengdu 610031, China
[10] Sicong Wu, Weiwe Zhao, Yanhong CHhen,??Application research of Hilbert transform in reactive energy measurement",Hunan



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY