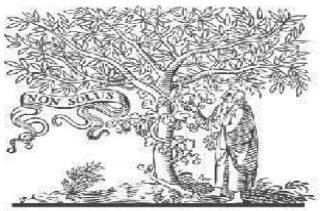
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT



ELSEVIER
SSRN

IJIEMR Transactions, online available on 20th Feb 2019.
Link : https://ijiemr.org/downloads/Volume-08/Issue-02

Title: **Implementation of ZigBee communication between FPGA and PC**

volume 08, Issue 02, Pages: 160-168

Paper Authors: Theleru Surender Reddy[1], Masanipalli Kranthi Kumar[2]

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Implementation of ZigBee communication between FPGA and PC

Theleru Surender Reddy[1], Masanipalli Kranthi Kumar[2]
Associate Professor[1], Assistant Professor[2]
DEPARTMENT OF ECE
MALLA REDDY ENGINEERING COLLEGE(MREC)

Abstract- ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on the IEEE 802.15.4-2003 standard for wireless personal area networks (WPANs), such as wireless headphones connecting with cell phones via short-range radio. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth. ZigBee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life, and secure networking.TheZigBee Alliance is a group of companies that maintain and publish the ZigBeestandard.ZigBee is a low-cost, low-power, wireless mesh networking proprietary standard. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the low power-usage allows longer life with smaller batteries, and the mesh networking provides high reliability and larger range.

Index Terms- ZigBee, IEEE 802.15.4-2003 standard, radio-frequency, wireless personal area networks

I. INTRODUCTION

XBee is a wireless network protocol specifically designed for low rate sensor and control networks. There are a number of applications that can benefit from the XBee protocol: building automation networks, home security systems, industrial control networks, remote metering and PC peripherals are some of the many possible applications. Compared to other wireless protocols, the XBee wireless protocol offers low complexity, reduced resource requirements and most importantly, a standard set of specifications. It also offers three frequency bands of operation along with a number of network configurations and optional security capability. UART The Universal Asynchronous Receiver and Transmitter (UART) controller is the key component of the serial communication subsystem of a computer. The UART takes bytes of data and transmits the individual bits in sequential fashion. At the destination, a second UART reassembles the bits in to complete bytes. Serial transmission is commonly used with modems and for non-networked communication between computers, terminals and other devices. The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing, or break interrupt). The UART includes a programmable baud rate generator that is capable dividing the timing reference clock input by divisors of 1 to 216, and producing a 16 clock for driving the internal transmitter. Provisions are also included processor-interrupt system.Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link. The Serial Port is harder to interface than the Parallel Port. In most cases, any device you connect to the serial port will need the serial transmission converted back to parallel so that it can be used. This can be done using a UART. This project deals with design and functional simulation of the UART and XBee module interfaced between the FPGA and PC. A BEHAVIORAL model is to be developed using VHDL, which consists of two blocks namely-Receiver and Transmitter. The interface with the CPU parallel transmission and with the modem for serial transmission has to be functionally simulated and results are to be analyzedIn this project we use two Zigbee devices one at the Transmitter end and one at the receiver end. Here, interchangeable represents a single Zigbee device can act as a transmitter device and as well as a receiver device. Here we will implement the Zigbee Communication between FPGA and a PC.The Parallel data is converted into Serial byte by the UART in the PC and this serial data is transmitted to the Zigbee device at the transmitter side.The second Zigbee device in the receiver side receives this serial byte and is given to the UART which converts serial data into parallel byte and is finally given to the FPGA.In this way transmission of the data takes place between FPGA and PC.

The Universal Asynchronous Receiver and Transmitter (UART) controller is the key component of the serial communication subsystem of a computer. The UART takes bytes of data and transmits the individual bits in sequential fashion. At the destination, a second UART reassembles the bits in to complete bytes. Serial transmission is commonly used with modems and for non-networked communication between computers, terminals and other devices. The UART performs serial-to-parallel

conversion on data characters received from a peripheral device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing, or break interrupt). The UART includes a programmable baud rate generator that is capable of dividing the timing reference clock input by divisors of 1 to 216, and producing a 16 clock for driving the internal transmitter. Provisions are also included processor-interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link. The Serial Port is harder to interface than the Parallel Port. In most cases, any device you connect to the serial port will need the serial transmission converted back to parallel so that it can be used. This can be done using a UART.This project deals with design and functional simulation of the Universal Asynchronous Receiver and Transmitter. A BEHAVIORAL model is to be developed using VHDL, which consists of two blocks namely-Receiver and Transmitter. The interface with the CPU parallel transmission and with the modem for serial transmission has to be functionally simulated and results are to be analyzed.

Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver. Instead, the sender and receiver must agree on timing parameters in advance and special bits are added to each word, which are used to synchronize the sending and receiving units. When a word is given to the UART for Asynchronous transmissions, a bit called the "Start Bit" is added to the beginning of each word that is to be transmitted. The Start Bit is used to alert the receiver that a word of data is about to be sent, and to force the clock in the receiver into synchronization with the clock in the transmitter. After the Start Bit, the individual bits of the word of data are sent, with the Least Significant Bit (LSB) being sent first. Each bit in the transmission is transmitted for exactly the same amount of time as all of the other bits, and the receiver ``looks" at the wire at approximately halfway through the period assigned to each bit to determine if the bit is a 1 or a 0. For example, if it takes two seconds to send each bit, the receiver will examine the signal to determine if it is a 1 or a 0 after one second has passed, then it will wait two seconds and then examine the value of the next bit, and so on.

II. Existing Work or Literature Survey

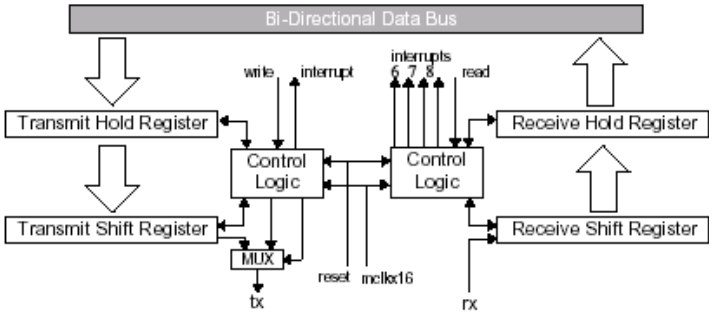
RS-232 is the name of the serial data interface standard used to connect computers to modems. A summary of the serial port pins and their function follow RS-232 Communication is asynchronous. That is a clock signal is not sent with the data. Each word is synchronized using its start bit, and an internal clock on each side, keeps tabs on the timing. The diagram below shows the expected waveform from the UART when using the common 8N1format. 8N1 signifies 8 Data bits, No Parity and 1 Stop Bit. The RS-232 line, when idle is in the Mark State (Logic 1). A transmission starts with a start bit, which is (Logic 0). Then each bit is sent down the Line, one at a time. The LSB (Least Significant Bit) is sent first. A Stop Bit (Logic 1) is thenappended to the signal to make up the transmission. The diagram shows the next bit after the Stop Bit to be Logic 0. This must mean another word is following, and this is its Start Bit. If there is no more data coming then the receive line will stay in its idle state (logic 1). We have encountered something called a "Break" Signal. This is when the data line is held in a Logic 0 state for a time long enough to send an entire word. Therefore if you don't put the line back into an idle state, then the receiving end will interpret this as a break signal.The data sent using this method, is said to be framed. That is the data is framed between a Start and Stop Bit. Should the Stop Bit be received as logic 0, then a framing error will occur. This is common, when both sides are communicating at different speeds. The above diagram is only relevant for the signal immediately at the UART. RS-232 logic levels uses +3 to +25 volts to signify a "Space" (Logic 0) and -3 to -25 volts for a "Mark" (logic 1). Any voltage in between these regions (i.e. between +3 and -3 Volts) is undefined. Therefore this signal is put through a "RS-232 Level Converter". This is the signal present on the RS-232 Port of your computer.

In modern systems there are two types of Break signals. If the Break is longer than 1.6 seconds, it is considered a "Modem Break", and some modems can be programmed to terminate the conversation and go on-hook or enter the modems' command mode when the modem detects this signal. If the Break is smaller than 1.6 seconds, it signifies a Data Break and it is up to the remote computer to respond to this signal. Sometimes this form of Break is used as an Attention or Interrupt signal and sometimes is accepted as a substitute for the ASCII CONTROL-C character. Marks and Spaces are also equivalent to ``Holes" and ``No Holes" in paper tape systems. Note: Breaks cannot be generated from paper tape or from any other byte value, since bytes are always sent with Start and Stop bit. The UART is usually capable of generating the continuous Spacing signal in response to a special command from the host processor. The sender does not know when the receiver has ``looked" at the value of the bit. The sender only knows when the clock says to begin transmitting the next bit of the word. When the entire data word has been sent, the transmitter may add a Parity Bit that the transmitter generates. The Parity Bit may be used by the receiver to perform simple error checking. Then at least one Stop Bit is sent by the transmitter. 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. If the Stop Bit does not appear when it is supposed to, the UART considers the entire word to be garbled and will report a Framing Error. When the entire data word has been sent, the transmitter may add a Parity Bit that the transmitter generates. The Parity Bit may be

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

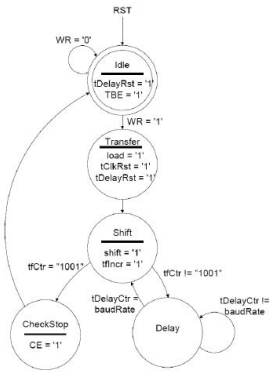
used by the receiver to perform simple error checking. Then at least one Stop Bit is sent by the transmitter.

Transmitted bytes are queued up awaiting transmission. When a byte is moved from the UART transmitter holding register to the UART transmitter shift register, an interrupt is generated and the next byte is taken from the transmitter buffer by the ISR and written to the UART holding register. The PC16550D s an improved version of original 645 Universal Asynchronous Receiver/Transmitter (UART).Functionally identical to the 16450 on power up (CHARACTER mode) the PC16550D can be put into an alternate mode (FIFO mode) to relieve the CPU of excessive software overhead.In this mode internal FIFOs are activated allowing 16 bytes(plus 3 bits of error data per byte in the RCVR FIFO) to be stored in both receive and transmit modes. Two pin functions have been changed to allow signaling of DMA transfers. The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a MODEM and parallel-to-serial conversion on data characters received From the CPU. The CPU can read the complete status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing, or break interrupt). The UART includes a programmable baud rate generator that is capable of dividing the timing reference clock input by divisors of 1 to (216b1), and producing a 16 c clock for driving the internal transmitter logic. Provisions are also included processor-interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link. The UART is fabricated using National Semiconductor's advanced M2CMOS process.



The UART chip has two sections transmitter and receiver. The basic function of transmitter section is to convert given parallel data to serial form. That is, when transmitting, the UART takes 8 bits of parallel data converts the data to a serial bit stream that consists of a start bit (logic „0"), 8 data bits (LSB first), and one or more stop bits (logic „1"). The UART is connected to a micro controller data and address bus so that the CPU can read and write to the registers. The following Two 8 bit registers are used in Transmitter.The bit stream coming in on RXD is not synchronized with the local clock (clk1x). If we attempted to read RXD at the raising edge of clk1x we would have a problem if RXD changed near the clock edge. We could have setup and hold time problems. If the bit rate of the incoming signal differed from clk1x by a small amount, we could end up reading some bits at the wrong time. Hence RXD should synchronize with the local clock (clk1x).Beside the registers the main blocks of the UART are the transmitter, receiver and baud generator. The baud generator divides down the system clock to provide the local clock.

The UART has two main functional blocks that can be thought of as two separate circuits packaged in one component. The UART includes a circuit for receiving serial information, and a circuit for transmitting serial information. The receiver takes in a byte of serial data transmitted to the RXD port, and converts it into one byte of parallel information. This byte is then placed on the DBOUT port. The transmitter takes a byte of parallel information found on the DBIN port, and converts it into a byte of serially transmitted data. This data is transmitted on the TXD port. Figure 5 shows the UART separated into its two circuits.The transmitter portion of the UART accepts a byte of data from the DBIN port and transmits it as serial data on the TXD port. The baud rate for the transmission is the same as the baud rate for the receiving portion of the UART, so it can be changed in the same way as described in the "Receiving Protocol" section. To transmit the byte stored in the DBIN port, thetransfer portion of the UART must contain a transfer controller, two counters for synchronization, and a transfer shift register. The transfer controller uses the two synchronization counters to control the rate at which the data byte must be transmitted across the TXD port, and the number of bits transmitted. One counter is used to delay the transfer controller between transmissions, while the other counter is used to keep track of how many transmissions have been sent out. The TXD port is set equal to the least significant bit of the transfer shift register, allowing data to be transmitted by simply shifting the register to the right once at the time of each transmission. A state diagram of the transfer controller can be seen below:



The transfer portion of the UART stays idle until the WR port goes high. When this happens, the UART is told to write whatever data is in the DBIN port, and the next state, Transfer, is loaded in the transfer state machine. The Transfer state prepares the transfer shift register to transmit data. By setting load = „1",

the shift register is loaded with a start bit, the data byte in DBIN, a proper parity bit, and a stop bit. The two reset signals, tClkRst and tDelayRst, are also set to „1" to reset the two synchronization counters. The next state loaded is the Shift state. The Shift state sets the shift signal to „1" in order to shift the shift register right one. The tfIncr signal is also set to „1" to increment the data counter. If the data counter is not equal to nine, all of the transfer shift register has not been shifted yet, and the Delay state is gone too. If all of the transfer shift register has been shifted, the Wait Write state is loaded. The Delay state is entered so that the transmission process sends out the data at the appropriate baud rate. Once the tDelayCtr is equal to the baudRate constant, the delay state is left, and the shift state is reloaded. Once the WaitWrite state has been entered, the transmission is complete. This state is needed to make sure that the WR port that was held high to start the transfer process has been unasserted.Without this state, if the WR port is held high for too long, a garbage transmission will occur.

The Field Programmable Gate Array or FPGA is a type of device that is widely used in the logic or digital electronic circuits. FPGAs are semiconductor devices that contain programmable logic and interconnections. The programmable logic components, or logic blocks as they are known, may consist of anything from logic gates, through to memory elements or blocks of memories, or almost any element. The great advantage of the FPGA is that the chip is completely programmable and can be re-programmed. In this way it becomes a large logic circuit that can be configured according to a design, but if changes are required it can be re-programmed with an update. Thus if circuit card or board is manufactured and contains an FPGA as part of the circuit, this is programmed during the manufacturing process, but can later be re-programmed to reflect any changes. Thus it is field programmable, giving rise to its name. Although FPGAs offer many advantages, there are naturally some disadvantages. They are slower than equivalent ASICs (Application Specific Integrated Circuit) or other equivalent ICs, and additionally they are more expensive. (However ASICs are very expensive to develop by comparison). This means that the choice of whether to use an FPGA based design should be made early in the design cycle and will depend on such items as whether the chip will need to be re-programmed, whether equivalent functionality can be obtained elsewhere, and of course the allowable cost. Sometimes manufacturers may opt for an FPGA design for early product when bugs may still be found, and then use an ASIC when the design is fully stable. FPGAs are used in many applications. In view of the cost they are not used in cheap high volume products, but instead FPGAs find applications in a variety of areas where complex logic circuitry may be needed, and changes may be anticipated. FPGA applications cover a wide range of areas from equipment for video and imaging, to circuitry for aerospace and military applications, as well as electronics for specialized processing and much more.

ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on the IEEE 802.15.4-2006 standard for wireless personal area networks (WPANs), such as wireless headphones connecting with cell phones via short-range radio. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth. ZigBee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life, and secure networking. The ZigBee Alliance is a group of companies which maintain and publish the ZigBee standard.

ZigBee operates in the industrial, scientific and medical (ISM) radio bands; 868 MHz in Europe, 915 MHz in countries such as USA and Australia, and 2.4 GHz in most jurisdictions worldwide. The technology is intended to be simpler and less expensive than other WPANs such as Bluetooth. ZigBee chip vendors typically sell integrated radios and microcontrollers with between 60K and 128K flash memory, such as the free scale MC13213, the Ember EM250 and the Texas Instruments CC2430. Radios are also available stand-alone to be used with any processor or microcontroller. Generally, the chip vendors also offer the ZigBee software stack, although independent ones are also available. "In the US, as of 2006, the retail price of a Zigbee-compliant transceiver is approaching $1, and the price for one radio, processor, and memory package is about $3. Comparatively, the price of consumer-grade Bluetooth chips is now under $3. In other countries the prices are higher. For example in the UK, (March 2009) the one-off cost to a hobbyist for a barebones ZigBee surface-mount transceiver IC varies from £5 to £9, with pre-assembled modules around £10 more (excluding aerials).The first stack release is now called Zigbee 2004. The second stack release is called Zigbee 2006, and mainly replaces the MSG/KVP structure used in 2004 with a "cluster library". The 2004 stack is now more or less obsolete.

III. WRITE DOWN YOUR STUDIES AND FINDINGS

The software is designed to be easy to develop on small, inexpensive microprocessors. The radio design used by ZigBee has been carefully optimized for low cost in large scale production. It has few analog stages and uses digital circuits wherever possible. Even though the radios themselves are inexpensive, the ZigBee Qualification Process involves a full validation of the requirements of the physical layer. This amount of concern about the Physical Layer has multiple benefits, since all radios derived from that semiconductor mask set would enjoy the same RF characteristics. On the other hand, an uncertified physical layer that malfunctions could cripple the battery lifespan of other devices on a ZigBee network. Where other protocols can mask poor sensitivity or other esoteric problems in a fade compensation response, ZigBee radios have very tight engineering constraints: they are both power and bandwidth constrained. Thus, radios are tested to the ISO 17025 standard with guidance given by Clause 6 of the 802.15.4-2006 Standard.

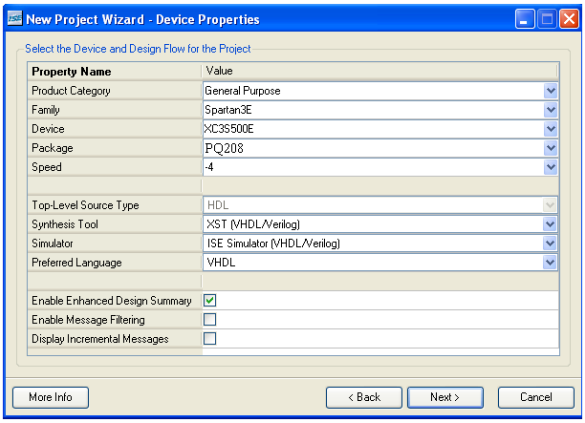
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Most vendors plan to integrate the radio and microcontroller onto a single chip.

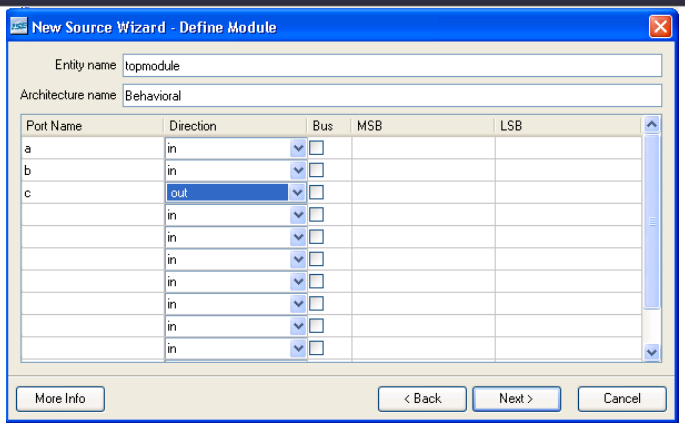
ZigBee protocols are intended for use in embedded applications requiring low data rates and low power consumption. ZigBee's current focus is to define a general-purpose, inexpensive, self-organizing mesh network that can be used for industrial control, embedded sensing, medical data collection, smoke and intruder warning, building automation, home automation, etc. The resulting network will use very small amounts of power -- individual devices must have a battery life of at least two years to pass ZigBee certification.At its core, ZigBee is mesh network architecture. Its network layer natively supports three types of topologies: both star and tree typical networks and generic mesh networks. Every network must have one coordinator device, tasked with its creation, the control of its parameters and basic maintenance. Within star networks, the coordinator must be the central node. Both trees and meshes allow the use of ZigBee routers to extend communication at the network level (they are not ZigBee coordinators, but may act as 802.15.4 coordinators within their personal operating space), but they differ in a few important details: communication within trees is hierarchical and optionally utilizes frame beacons, whereas meshes allow generic communication structures but no router beaconing.As the size and the complexity of digital system increases, more computer aided design tools are introduced into the hardware design process. The early papered pencil design methods have given way to sophisticated design entry, verification and automatic hardware generation tools. The newest addition to this design methodologies the introduction of hardware description language (HDL).Actually the use of this language is not new languages such as CDI, ISP and AHPL have been used for last some years. However, their primary application has been the verification of designs architecture. They do not have the capability to model design with a high degree of accuracy that is, their timing model is not precise and/or their language construct implies a certain hardware structure newer languages such as VHDL have more universal timing models and imply no particular hardware structure. Hardware description languages have two main applications documenting a design and modeling it. Good documentation of a design helps to ensure design accuracy and design portability. Since a simulator supports them inherent in a HDL description can be used to validate a design. Prototyping of complicated system is extremely expansive, and the goal of those concerned with the development of hardware languages is to replace this prototyping process with validation through simulation and silicon compilation. Once an entity has been modeled, it needs to be validated by the VHDL system. A typical VHDL system consists of an analyzer and a simulator. The analyzer reads in one or more design units contained in a single file and compiles them into a design library after validating the syntax and performing some static semantic checks. The design library is a place in the host environment where compiled design units are stored.

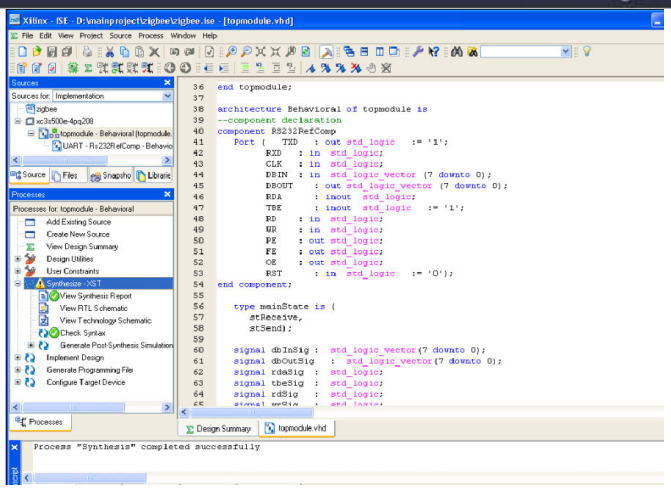
The major capabilities that the language provides along with the features that differentiate it from other hardware description languages.The language can be used an exchange medium between chip vendors and CAD tools users. Different chip vendors can provide VHDL descriptions of their components to system designers. CAD tool users can use it to capture the behavior of the design at a high level of abstraction for functional simulation. The language supports hierarchy that is a digital system can be modeled as a set of interconnected components, each component, in turn can be modeled as a set of interconnected sub components. The language is not technology specific, but is capable of supporting technology specific features. It can also support various hardware technologies, for example you may define new logic types and new components, also specify technology specific attributes. By being technology independent, the same model can be synthesized into different vendor libraries. It supports both synchronous and asynchronous timing models. Various digital modeling techniques such as finite state machine descriptions, algorithmic descriptions and Boolean equations can be modeled using the language. Test benches can be written using the same language to test other VHDL models.

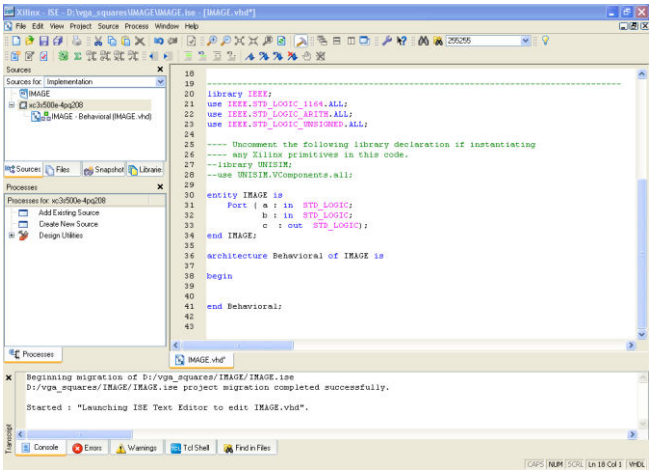
IV. RESULTS AND DISCUSSION

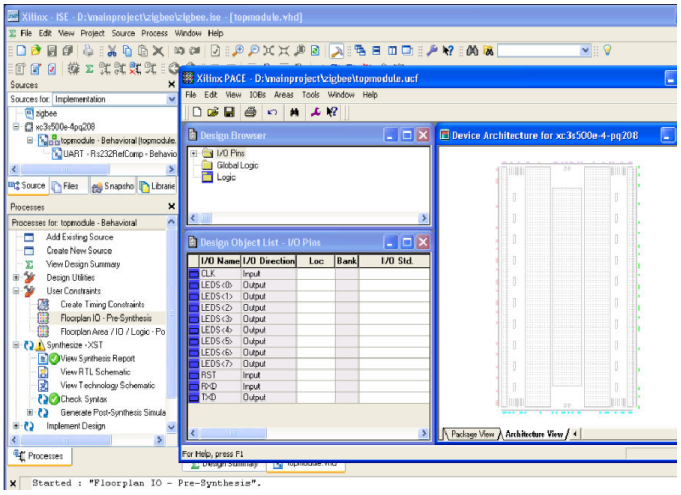


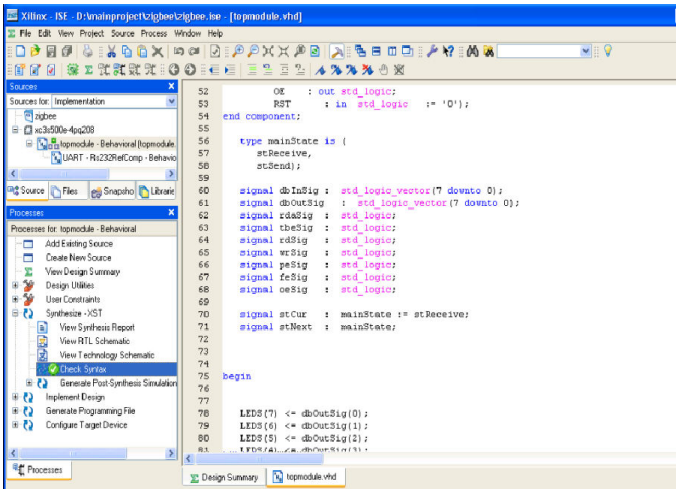
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

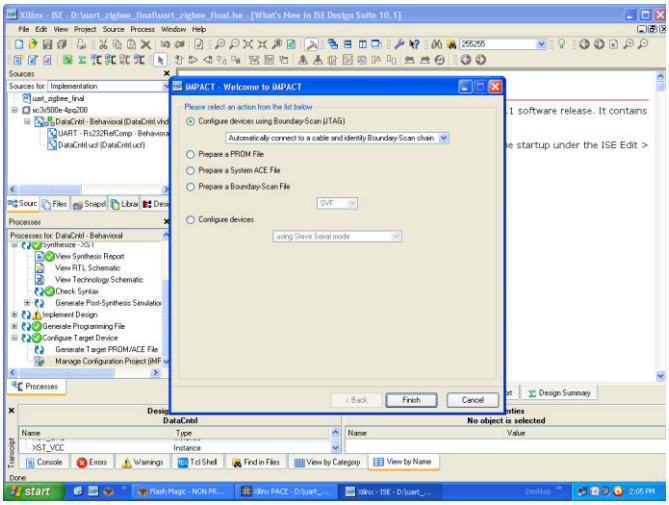




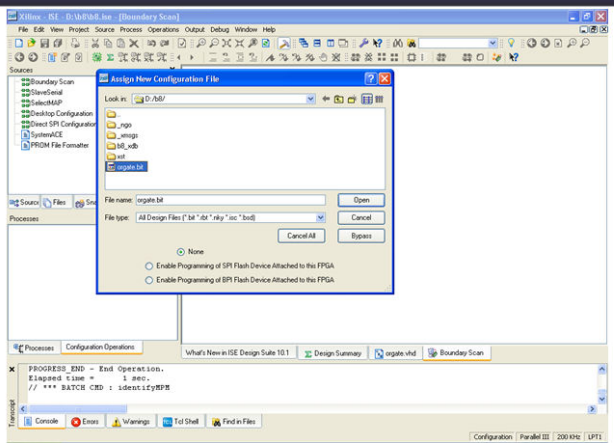


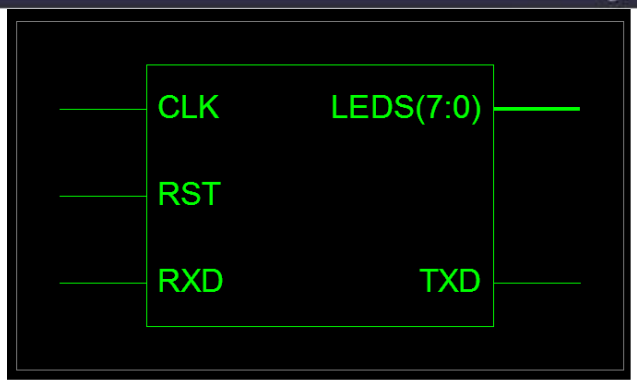




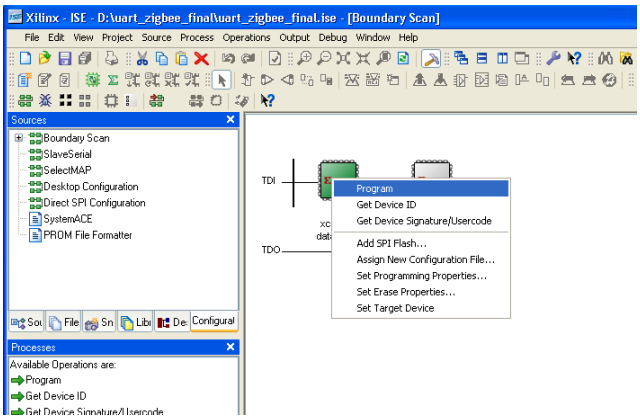


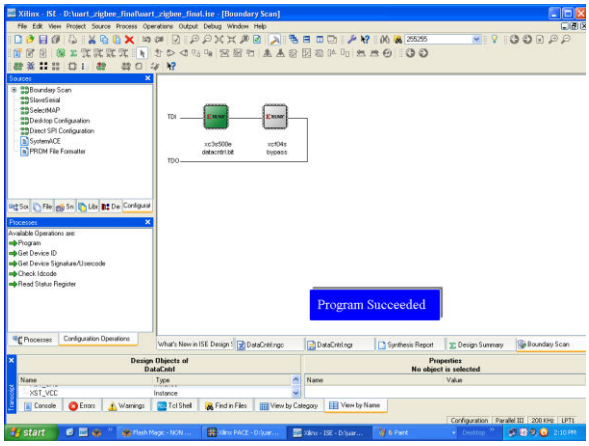
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

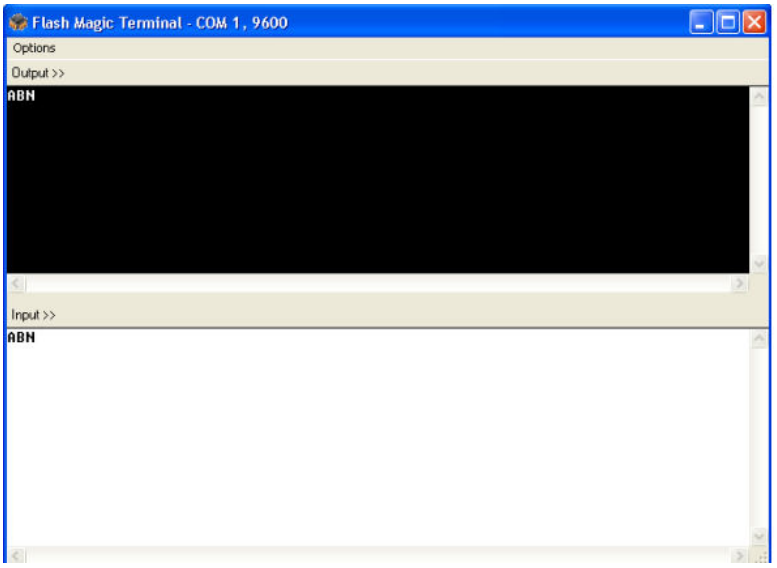




Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Total Number Slice Registers	69	9,312	1%
Number used as Flip Flops	60		
Number used as Latches	9		
Number of 4 input LUTs	50	9,312	1%
Logic Distribution			
Number of occupied Slices	49	4,656	1%
Number of Slices containing only related logic	49	49	100%
Number of Slices containing unrelated logic	0	49	0%
Total Number of 4 input LUTs	58	9,312	1%
Number used as logic	49		
Number used as a route-thru	8		
Number used as Shift registers	1		
Number of bonded IOBs	12	158	7%
Number of BUFGMUXs	2	24	8%







V. CONCLUSION

As one of its defining features, ZigBee provides facilities for carrying out secure communications, protecting establishment and transport of cryptographic keys, cyphering frames and controlling devices. Essentially XBEE can offer the „TV remote

control" for their lights, thermostat, and security system. XBEE also has great potential in the commercial segment, with applications such as lighting and HVAC control in hotels and other commercial buildings. The cost savings is potentially enormous when you can control energy usage on an almost real-time basis." We have successfully transmitted the data between UART implemented FPGA and PC with the help of the wireless XBEE module which is having higher data rates and lower cost when compared to Bluetooth devices. The data is received at the FPGA and is retransmitted back to the PC which can be seen in Flash Magic tool. The logic is implemented in FPGA by using the software tools.

REFERENCES

www.Zigbee.Org
www.forums.xilinx.com
www.fpga-faq.com
http://en.wikipedia.org/wiki/ZigBee
www.fpgaforum.blogspot.com
VHDL Primer - Bhaskar.