FPGA PROTOTYPING OF UNIVERSAL ASYNCHRONOUS RECEIVER-TRANSMITTER (UART) USING ALTERA VHDL IMPLEMENTATION

NABIHAH @ NORNABIHAH BTE AHMAD

A project report is submitted as partial fulfillment
of the requirements for the award of the
Master Degree of Engineering (Electrical)

Faculty of Electrical Engineering
Kolej Universiti Teknologi Tun Hussein Onn

NOVEMBER 2005

DEDICATION

Special dedication to my beloved husband Zakhir b Ngadengon and my son Muhammad Zahin Naufal, my parent-in laws and my families for all your support and love, makes all things possible for me.

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude toward my supervisor, Prof Madya Awtar Singh who gives his invaluable advice, time, suggestions and guidance throughout this study. Also, I am tremendously grateful to my friends who been very supportive and cooperative in helping me out to complete this project successfully.

ABSTRACT

External devices such as modems and other computers need to communicate serially. In order to provide this communication, a universal asynchronous receiver-transmitter (UART) can provide an asynchronous serial data communication with I/O outputs devices such as keyboard, mouse or keypad. It can transmit serial data on over it's transmit line (TxD) and receive serial data over it's receive line (RxD). This project describes a universal asynchronous receiver-transmitter (UART) design. It is design from Very High Speed Integrated Circuit Hardware Description Language (VHDL) description, and then to Field Programmable Gate Array (FPGA) implementation. VHDL is used to provide a simple way of design entry through behavioral description. This project covers VHDL integration issue involved in the flow from high-level description to a fully simulated and synthesized. FPGA University Program Educational Board (UP2) is used to achieve fast prototype build and logic circuit verification. Then analysis the features of other existing UART technology design.

ABSTRAK

Alatan luaran seperti modem dan komputer lain memerlukan komunikasi secara sesiri. Untuk menyediakan komunikasi sebegini, Penerima- Penghantar Tak Segerak Umum (UART) boleh menyediakan komunikasi tak segerak secara sesiri dengan keluaran I/O peralatan seperti papan kekunci, tetikus dan kekunci kecil. Ia boleh menghantar data sesiri melalui talian hantar (TxD) dan menerima data sesiri melalui talian terima (RxD). Projek ini menghuraikan rekabentuk Penerima- Penghantar Tak Segerak Umum (UART). Ia direkabentuk menggunakan bahasa Very High Speed Integrated Circuit Hardware Description Language (VHDL) dan kemudian diimplementasi ke Field Programmable Gate Array (FPGA). VHDL digunakan bagi menyediakan jalan mudah untuk masukan rekabentuk melalui huraian kelakuan. Projek ini merangkumi integrasi VHDL merangkumi pergerakan daripada tahap huraian tinggi kepada simulasi penuh dan sintesis. FPGA University Program Educational Board (UP2) digunakan untuk mendapatkan binaan prototaip cepat dan pengesahan litar logik. Kemudian membuat analisis tentang ciri-ciri rekabentuk teknologi UART lain yang sedia ada.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

1.1 Foreword

This chapter will discuss about the background, purpose, objectives, and the scope of the project.

1.2 Problem Statement

Quick turn prototyping for logic ICs is critical for system development and meeting critical market demands. Nowadays a designer must choose either to

implement logic either in ASIC or FPGA technology. FPGA is good for design Turn-Around-Time (TAT) and flexibility for FPGA. FPGA provide a good solution for fast prototyping and flexible IC solution to develop a product and, hence, a system quickly. FPGA offer an advantage of a low cost for a low volume production and can be reprogrammable.

Implementation with FPGA accomplished through VHDL synthesis into RTL or gate level representation. Coding in VHDL is simple and easily understood, both for prototype build, future upgrade and modification. The logic and code are also easily verified and fixes is fast, and the result can be simulate instantly. It enables a standard design practice for digital circuit in industry. It also provides a simple way to test out and modify code design. It able to analyze the code and perform optimization.

Another problem is about serial communication. Bits have to be moved from one place to another using wires or some other medium. Over many miles, the expense of the wires becomes large. To reduce the expense of long communication links carrying several bits in parallel, data bits are sent sequentially, one after another. So the alternative is by using UART to convert the transmitted bits from sequential to parallel from at each end of the link. Universal Asynchronous Receiver-Transmitter is known, as UART is a piece of computer hardware that translates between parallel bits of data and serial bits. UART acts as the interface between an I/O bus and a serial device. It controls a computer's interface to its attached serial device such as a mouse or modem, which communicates with a computer one bit at a time. So this project approached a UART design using VHDL to implement into FPGA.

1.3 Objectives

The objectives of this project are:

i. To design a serial asynchronous receiver and transmitter that enables communication with standard I/O serial devices such as mouse, modem or keyboard.
ii. To design a UART that can be prototyping with an FPGA, then testing and debugging the FPGA prototype in-system.
iii. To analyze the features of UART technology and make comparison with other existing UART technologies.

1.4 Scope Project

The scope of the project is as followed:

i. To design a UART by using VHSIC hardware description language (VHDL) in Altera MAX+PLUSII environment.

ii. To synthesize the VHDL code of design, obtain the RTL or gate level representation that is able to download into FPGA device.

iii. To simulate the UART using VHDL language to verify the code functions properly under Altera MAX+PLUSII environment.

iv. To implement the asynchronous receiver-transmitter in University Program Educational Board (UP2) to achieve fast prototyping and design verification.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter provides a brief review of existing Universal Asynchronous Receiver-Transmitter (UART) Design and FPGA Architecture. By reviewing relevant UART design and FPGA architecture will allows achieving better insight of the need of the new design and critically prepare for the ideas and their implication presented in the following chapters.

2.2 UNIVERSAL ASYNCHRONOUS RECEIVER-TRANSMITTER DESIGN

The word "asynchronous" indicates that UART recover character timing information from the data stream, using designated "start" and "stop" bits to indicate the framing of each character. The history of the first UART-like devices was rotating mechanical commutators. These sent 5-bit codes for mechanical teletypewriters, and replaced morse code. Later, ASCII required a seven-bit code. When IBM rationalized computers in the early 1960s with 8-bit characters, it became customary to store the ASCII code in 8 bits. An example of an early 1980s UART was the National Semiconductor 8250. In the 1990s, newer UART were developed with on-chip buffers. This allowed higher transmission speed without data loss and without requiring such frequent attention from the computer. For example, the National Semiconductor 16550 has a 16 byte FIFO. Variants include the 16C550, 16C650, 16C750, and 16C850.

Depending on the manufacturer, different terms are used to identify devices that perform the UART functions. Intel called their 8251 device a "Programmable Communication Interface". The term "Serial Communications Interface" (SCI) was first used at Motorola around 1975 to refer to their start-stop asynchronous serial interface device, which others were calling a UART. The less-common 5, 6 and 7 bit codes are now sometimes simulated with 8-bit UART. The unused high-order bits are set to 1, the value of the stop bit and idle line. This technique cannot send or receive at full speed, but provides some level of compatibility for older equipment.

2.2.1 New IC Caps Two Decades of UART Development by MAXIM

Maxim has introduced a tiny universal asynchronous receiver/transmitter (UART) that is compatible with the miniature electronic components in today's portable products. Compared with well-established UART already on the market, the new MAX3100 offers numerous advantages such as lower cost, higher speed (to 230kbaud), lower power and lower voltage operation (<3V), and special features that include IrDA timing for IR communications with other serial infrared (SIR)-compatible devices, or for reducing power in opto-isolated applications and a FIFO buffer to relieve the processing burden in small systems. It also includes Schmitt-trigger inputs and high output drive, for direct optocoupler interface in isolated systems.

In converting these requirements to silicon, Maxim has produced a tiny, full-featured UART called the MAX3100. To help minimize size and pin count, it features a synchronous serial peripheral interface (SPI) for communications. A serial interface for a serial-interface IC may sound paradoxical, but it enables a complete, full-featured UART to fit in the footprint of an SO-8 package (the actual package is a 16-pin QSOP). Many µCs include the serial interface built into the MAX3100. Thus, the MAX3100 enables high-performance communications for most systems-without major trade-offs in size, cost, and power, and without the additional trade-offs associated with software UART.

The MAX3100 UART feature provides an interface between the synchronous serial-data port of a µP (compatible with SPI™, QSPI™, and Microwire™ standards), and an asynchronous serial-data communications port such as RS-232, RS-485, or IrDA. For a brief description of SPI, see the sidebar to this article, "Serial Peripheral Interfaces."

The MAX3100 combines a simple UART and baud-rate generator with an SPI interface and interrupt generator. Writing to an internal register configures the UART for baud rate, data-word length, parity enables, and enable of the 8-word receive FIFO. This "write configuration" register contains four interrupt mask bits, and it also selects between normal UART and IrDA timing. The programmable baud-rate generator is capable of rates from 300baud to 230kbaud. The MAX3100 oscillator accepts a crystal of 1.8432MHz or 3.6864MHz, and it also accepts a square wave at X1 with a 45% to 55% duty cycle.

2.2.2 Universal Asynchronous Receiver/Transmitter by Lattice Semiconductor Corporation

This application note describes a fully configurable UART optimized for and implemented in a variety of Lattice devices, which have superior performance and architecture compared to existing semiconductor ASSPs (application-specific standard products). This UART reference design contains a receiver and a transmitter. The receiver performs serial-to-parallel conversion on the asynchronous data frame received from the serial data input SIN. The transmitter performs parallel-to serial conversion on the 8-bit data received from the CPU. In order to synchronize the asynchronous serial data and to insure the data integrity, Start, Parity and Stop bits are added to the serial data.

This design can also be instantiated many times to get multiple UART in the same device. For easily embedding the design into a larger implementation, instead of using tri-state buffers, the bi-directional data bus is separated into two buses, DIN and DOUT. The transmitter and receiver both share a common internal Clk16X clock. This

internal clock, which needs to be 16 times of the desired baud rate clock frequency, is obtained from the on-board clock through the MCLK input directly. However, when implementing the design into ispMACH™ 5000VG devices, the Clk16X clock can be generated flexibly through the ispMACH 5000VG on-chip PLL by using MCLK as the PLL reference clock input. Faster performance than industry standard hardwired devices.

The UART features are Inserts or extracts standard asynchronous communication bits (Start, Stop and Parity) to or from the serial data. It also holding and shifting registers eliminate the need for precise synchronization between the CPU and serial data. Interactive control signaling and status reporting capabilities and can separate input and output data buses for use as an embedded module in a larger design. Transmitter enabled by new data writes to Transmit Holding Register. Receiver synchronizes off the Start bit. Receiver samples all incoming bits at the center of each bit.

This Lattice UART design has been implemented in several Lattice CPLD devices. The design software used for this implementation is the latest Lattice ispDesignEXPERT™ version (Lattice ispLEVER™ design software version 9.0 or later for ispMACH 5000VG implementation). After the design was completed, the VHDL code was compiled. The use of VHDL with a CPLD allows functional and hardware changes to be made quickly and easily. The number of macrocells required varies, depending on the addition of new functions and/or removal of unneeded features. This design uses 37 I/O pins. The maximum operating frequency currently is 55MHz with the M4A5-192/96-6VC. This is compared to the industry-standard UART device with a speed of 24MHz. The maximum baud rate frequency is 107.5 MHz.

2.2.3 Altera a6404 Universal Asynchronous Receiver Transmitter

The design of a6404 UART provides an interface between a microprocessor and a serial communication channel using VHDL in Altera Quartus II environment. It optimized for the FLEX, Stratix GX, Cyclone and APEX families, which uses 162 logic elements (LEs). This UART is programmable word length, stop bit and parity, which operate, in full duplex. It also includes a status flags for parity, overrun and framing error.

2.2.4 iniUART by INICORE Incorporation

The iniUART is an innovative, flexible implementation of a Universal Asynchronous Receiver Transmitter (UART) device. iniUART, which uses the RS- 232 serial protocol, provides the interface between a microprocessor and a serial port or between the system and a standard serial port. The core contains a highly accurate programmable baud rate generator, serial receiver and transmitter communications channels, and interrupts control signals. The feature of this UART is it the transfer rate is 1200bps to 115.2kbps with better accuracy of 0.1% from 8MHz Clock. The data format is 7 or 8 bits data and it also has a parity enable either odd or even and error detection. This UART can handle 1 or 2 stop bits.

The iniUART core may be used as a data link layer with parallel interfaces and event communication. Application-specific blocks (e.g., interrupt controller, special interfaces, and status reporting circuits) can then be built around the iniUART and will not affect the main functionality. The baud rate generator is not a simple prescaler, but

an innovative DCO (digitally controlled oscillator), which allows generating all baud rates from the system clock. There is no special quartz needed for the purpose.

2.2.5 Inventra™ M16x50 Enhanced UART with FIFO & IrDA Support

The M16x50 offers programmable word length (from five to eight bits), together with an optional parity bit and 1, 1½ or 2 stop bits. If enabled, the parity can be odd, even or forced to a defined state. The M16x50 includes a 16-bit programmable baud rate generator, an 8-bit scratch register and two DMA handshake lines, TXRDY and RXRDY, which are used to indicate when the FIFO is ready to transfer data to the CPU. Further, the DMA handshake lines may be configured either to support single-byte transfers (DMA Mode 0) or multiple-byte transfers (DMA Mode 1).

The M16x50 also includes eight modem control lines and a diagnostic loop-back mode. Interrupts can be generated for a range of TX Buffer/FIFO, RX Buffer/FIFO, Modem Status and Line Status conditions. The M16x50 offers both hardware flow control (in which the transmission and reception of data characters is controlled via the RTS and CTS modem status signals) and software flow control. Basic IrDA 1.0 SIR modulation/demodulation has been added to the M16x50 UART. This feature uses the x16 TXCLK and RXCLK to generate 3/16 width pulses.

2.2.6 Designing a UART using VHDL

A simple UART design using VHDL by Bertrand CUZEAU has been implemented. This design is synthesizable to FPGA technology, which it demo application applied on the existing ALSE demo boards. The baud rate is from 1200 to 115200 baud with a clock of 14.7456 MHz and no internal FIFO use.

Code is tested on Synthesis tools. The UART is design with an internal baud rate generator with two-baud rates selectable input signal. The design is very compact with asynchronous RS232 character-based transmit and receive function.

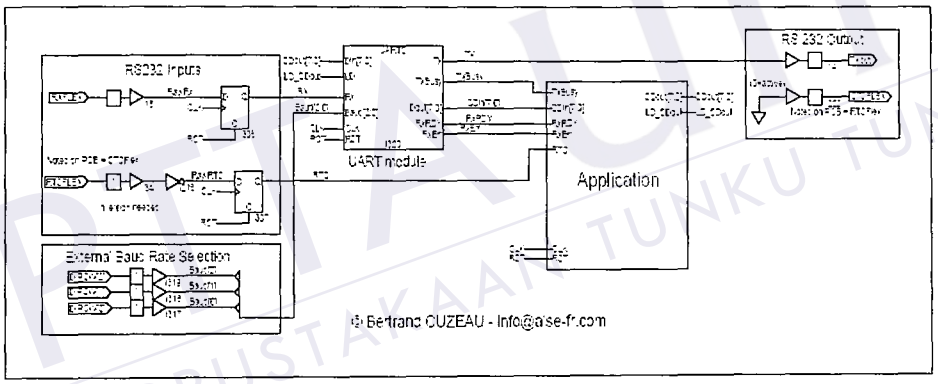

Figure 2.1: UART Architecture

REFERENCES

1. Zoran Salcic and Asim Smailagic (2000) "Digital Systems Design and Prototyping Using Field Programmable Logic and Hardware Description Languages." 2nd Edition

2. James O. Hamblen and Michael D. Furman (2001) "Rapid Prototyping of Digital Systems, A Tutorial Approach." United States of America: Kluwer Academic Publishers 2nd Edition. 2001

3. Bakri Madon (2004) "CMOS VLSI Design Fundamentals."

4. Nigel Horspool and Peter Gorman (2001) "The ASIC Handbook." United States of America: Prentice Hall

5. Zainalabedin Navabi (1998) "VHDL: Analysis and Modelling of Digital Systems."2nd Edition.United States of America: McGraw Hill

6. Pei An (1998) "PC Interfacing using Centronic, RS232 and Game ports." England: Newnes

7. Janick Bergeron (2000) 'tWriting Testbenches, Functional Verification of HDL Models." United States of America: Kluwer Academic Publisher.

8. Weng Fook Lee (2000). "VHDL Coding and Logic Synthesis with Synopsys." United States of America: Academic Press.

9. Peter Ngyren (2004). "An Application Programming Interface for Hardware and Software Threads

10. http://www.fpga.com.cn/freeip/uart_an_lattice.pdf

11. http://www.altera.com/literature/an/an311.pdf