

Data and Network Communications Protocol Motherboard Using Reconfigurable Hardware

ROSULA REYES^{1,2}, Ph.D., CARLOS OPPUS^{1,2}, JOSE CLARO MONJE^{1,2}
NOEL PATRON^{1,2}, RAPHAEL GONZALES², MARK GLENN RETIRADO²

¹Department of Electronics, Computer, and Communications Engineering
Ateneo de Manila University
Katipunan Avenue, Loyola Heights, Quezon City
rsjreyes@ateneo.edu, coppus@ateneo.edu, jcmonje@ateneo.edu, npatron@bcdph.com
PHILIPPINES

²Blue Chip Designs, Inc.
Unit 208 Xanland Place 323 Katipunan Avenue, Loyola Heights, Quezon City
rgonzales@bcdph.com, mgretirado@bcdph.com
<http://www.bcdph.com>
PHILIPPINES

Abstract: The introduction of reconfigurable devices such as the field programmable gate arrays (FPGA) have provided electrical, electronics and computer engineers with a versatile and cost-effective method of designing circuits, developing devices and implementing electronic, communications, computer and other related systems. This paper presents the use of FPGA in the development of a mother board to introduce the concepts of data and network communications protocol through different interfaces such as VGA, PS/2, serial communications and parallel communications. Moreover, since it is FPGA-based, this motherboard can be reconfigured or expanded to perform other protocols making it open to a lot of possibilities.

Key-Words: field programmable gate array, communications, data, network, motherboard, protocols, trainers

1 Introduction

Data handling and network communications has risen in importance in this time and age[1]. Almost all electronic equipment has some means of storing, manipulating, and communicating data with users and with other devices. With computer networks at the core of modern communication, different networks, as well as different technologies within them, need to be able to connect and understand data going to and through them. Most of this processes goes unnoticed by the regular user, even most of today's engineers know only the theory on most of this basic communication protocols and focus on the higher level logic of systems today. A basic understanding of these concepts is required for efficient high level programming. However, systems today that use these said communication protocols, such as personal computers, are already designed for other purposes. These same systems make these protocols

transparent to their users, and using them as training tools for these basic protocols will be costly. Thus, a low cost basic protocol motherboard will be beneficial especially for laboratories since low costs can mean more units and more time for hands-on exercises. Moreover, these motherboards will be FPGA-based and can be expanded with daughter boards for other purposes as the need arises.

2 Using FPGAs

Ten years ago, the thought of a single-chip design fulfilling all of the needs of a certain project was unimaginable. However, at present, the use of FPGAs has been a viable option.

An FPGA or field programmable gate array is a semiconductor device containing programmable logic components and interconnects[2]. These logic

components can be programmed to duplicate the functionality of digital circuits. To configure the FPGA, a circuit diagram or source code is written using a hardware description language (HDL) that describes the functionality of the FPGA. Thus, there is no need to think about the gate level implementation of the circuit.

With its wide-range of capabilities and ease of use, it has been able to tackle nearly any type of application imaginable. This includes designing FPGAs into products, using FPGAs to test products or even using FPGA-based hardware for controlling and manufacturing products.

Because of this flexibility, a motherboard to be used as a protocol trainer was designed using a FPGA by Blue Chip Designs (BCD).

Using a FPGA as a Central Processing Unit (CPU, various level converters and appropriate ports used by different communications standards, a communications protocol trainer can be developed.

For instance, connecting a DE-15 port and several lines of HDL, a very cheap 4096-color VGA controller is also created[3]. Another is to use the USB transceiver chip with appropriate DAC/ADC circuits and lines of HDL code, USB communication can be achieved. In addition, internal clocks can also be generated by the FPGA using phase-locked loop (PLL) circuits and most FPGAs today contain 1 to 8 PLLs[4]. Thus, only basic knowledge, coding skills, and good execution is needed to come up with an innovation.

2.1 VGA Display

This trainer is able to generate the needed signals for a VGA display monitor. There are five signals needed to display an image using the VGA port. These are the red signal, green signal, blue signal, horizontal synchronization, and vertical synchronization.

Data for the red, green, and blue (RGB) signals corresponds to the pixel data which represents the color and intensity of the pixel. This data is saved in a memory module. Then, data is retrieved from the memory module, formatted into lines of pixels, and sent to the display device with the appropriate horizontal and vertical synchronization pulses.

In addition, the pixel data used to form the image is accessed from the RAM using horizontal and vertical counters that are also used to generate the synchronization pulses for VGA output.

2.1.1 Text Display

Text is generated using a pixel map of individual characters. Each character is mapped on a grid with a size of 8 dots wide by 16 dots high. This map is used to create the pixel data of the needed character. Retrieval of the map is done by issuing the appropriate ASCII code for the character, as well as the coordinates of the individual pixel being generated. Table 1 shows a sample map for the character 'A.' 1 designates a pixel to be active on the display while a 0 otherwise.

Table 1 Sample Text Map for 'A'

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0
4	0	0	1	1	1	0	0	0
5	0	1	1	0	1	1	0	0
6	1	1	0	0	0	1	1	0
7	1	1	0	0	0	1	1	0
8	1	1	1	1	1	1	1	0
9	1	1	0	0	0	1	1	0
10	1	1	0	0	0	1	1	0
11	1	1	0	0	0	1	1	0
12	1	1	0	0	0	1	1	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0

2.2 Serial Port

Another communication method included in the motherboard is the serial communication. This is done through the inclusion of three (3) DE9 connectors.

In contrast to parallel communication, serial communication sends data one (1) bit at a time[1]. Although it may refer in general to all types of devices which transmits data serially, the most common standard associated with the term serial port is the RS232 standard.

Although all the ports are designed to be UARTs, the three ports on-board the motherboard are configured differently. The first two (2) ports are level converted to be able to communicate with a standard

RS232 port on a Personal Computer. The difference between these two ports is that the first port is configured to emulate a Data Terminal Equipment (DTE) while the second is configured to emulate a Data Circuit-terminating Equipment (DCE). The third port is configured for data transfer at TTL voltage levels.

The default operation for the serial ports on this training board is an implementation of an universal asynchronous receiver and transmitter(UART). A UART is a circuit that sends parallel data over a serial line. As its name implies, this operation contains a receiver module and a transmitter module. Since the transfer is asynchronous, meaning there is no clock information data sent over the line, the transmitter and receiver should beforehand agree on a set of parameters, such as transfer rate, number of data bits, stop bits and error correction bits and such, for a successful transfer of data.

2.2.1 UART Receiver

For a UART receiver, the timing of reading the serial input is essential to be able to receive the correct data. The timing is done using a baud rate generator. This generator creates a sampling clock that is a multiple of the transfer rate parameter that was agreed upon beforehand. Once a start bit is detected, this sampling clock is used to approximate the middle point of a bit to sample it appropriately. Data bits are stored as parallel data of a fixed length. A finite-state machine(FSM) is used to sample the serial data and store it in a buffer until all bits are received.

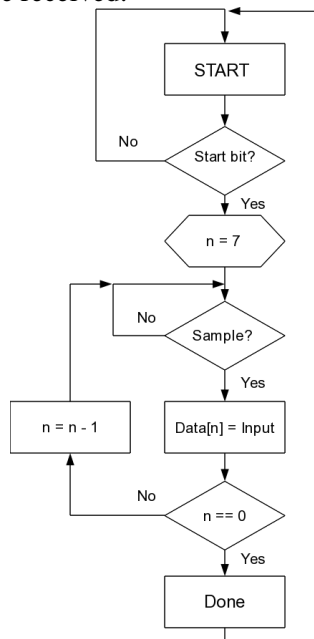


Fig. 1 UART Receiver Process

2.2.2 UART Transmitter

The UART Transmitter is basically a shift register that shifts of bits at the agreed upon transfer rate. This transmitter also needs a timing generator. Unlike the receiver, this timing can be at the same frequency as the transfer rate as this only signals when the next bit should be transmitted. Transfer is also done by a FSM, reading bits from an array one by one and transmitting them over the serial interface.

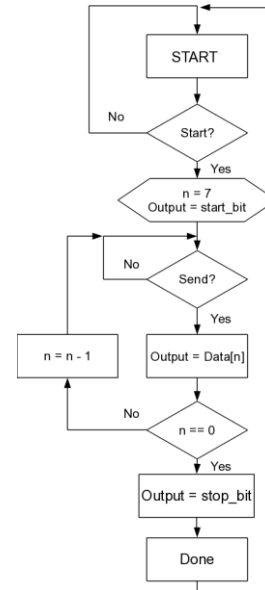


Fig. 2 UART Transmit Process

2.3 Personal System/2 (PS/2)

The Personal System/2 (PS/2) connector is used for connecting a keyboard and a mouse to a compatible computer system. The PS/2 port was first used on the Personal System/2 line of IBM computers first introduced in 1986, hence the name.

The PS/2 port contains two connections for communication purposes. One connection is for bi-directional data, which is transmitted serially. The other connection is for the clock, which specifies the timing when the data from the data connection is valid and can be sampled. The information is transmitted packets of eleven (11) bits. This packet contains a start bit, eight (8) data bits, an odd parity bit, and a stop bit[6].

The PS/2 designs on keyboard and mouse interfaces are electrically similar and employ the same communication protocol. However, a given system's keyboard and mouse port may not be interchangeable since the two devices use a different set of commands.

The motherboard is allocated with two ports for PS/2. It is designed for use with a keyboard and a mouse but it is reconfigurable through the FPGA.

2.3.1 PS/2 Keyboard

A keyboard consists of an array of keys. Activity done on a keyboard is monitored, or scanned, by a micro-controller. This micro-controller then sends the corresponding scan code over the PS/2 line.

The scan code consists of a sequence of codes that corresponds to the different activities that occur on the keyboard. When a key is pressed, the make code of the key is sent. If it is continued to be held down longer than 500ms, the make code is continually sent every 100ms. When the key is released, a break code is sent.

2.3.2 PS/2 Mouse

The mouse is used to detect motion on a two-dimensional surface. A mouse has an internal circuit that measures the relative distance the unit has traveled as well as the status of the buttons.

A mouse streams the data at a predetermined interval over the PS/2 port, however, upon power-up, the mouse needs to be sent commands to initialize the device. The PS/2 interface for the mouse needs to be bi-directional in this case, different from the keyboard which only sends data from the keyboard to the receiving device.

Over the predetermined interval, the mouse stores the relative movements over the x-axis and y-axis in an internal counter, as well as the three buttons, left, right and middle, states. These data are then sent through the PS/2 port in three bytes[6]. Once the data are sent, these internal counters are reset.

Table 2 Mouse Data Packet Format

Byte 1	y_V	x_V	1	y_8	x_8	m_B	r_B	l_B
Byte 2	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
Byte 3	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0

Table 2 denotes the bits in the packet sent by the mouse. The x-axis movement is denoted by x_8-x_0 with x_V signaling an overflow on the counter. The y-axis movement is contained in y_8-y_0 with y_V as the overflow signal. Middle, right and left buttons status signals are m_B , r_B and l_B respectively.

2.4 Universal Serial Bus (USB)

Universal Serial Bus (USB) is a specification to establish communication between devices and a host controller, usually personal computers. It was developed to interface external devices to a host controller by eliminating the different types of connectors and using a single standard for physical and logical connections.

A USB transceiver chip is used for the USB module. It is prepared on the motherboard with its corresponding circuit for signal preparation. This transceiver is controlled using the FPGA. Also, handshaking and data transfer is also controlled by the FPGA.

2.5 Parallel Port

This trainer is equipped with a sixteen (16) pin parallel interface or otherwise called a General Purpose Input/Output (GPIO) port. Aside from the Vcc pin and the ground pin, other pins are customizable to the users specifications. For this motherboard, the default configuration is the same as IEEE1284 standard for parallel communications.

Before USB, parallel communication was used to interface a wide range of peripheral devices to a computer. The most common of these devices was the printer. Currently, other interfaces were developed and are used other than the parallel port, predominantly the USB.

Recently, manufacturers consider parallel communications to be outdated and respective hardware to be legacy devices, opting to use more recent interfaces, such as USB or Firewire. As such, they removed parallel communication devices altogether.

2.6 Summary

The Data and Network Communications Protocol Motherboard Using Reconfigurable Hardware was designed to develop a firm base of knowledge by giving more of a hands-on approach to learning.

The core of this motherboard is a reconfigurable FPGA. Using a compiler and some basic skills. A user will be able to design and implement his own project using the different devices on the board.

The board is outfitted with a VGA port to be connected to a VGA ready monitor. The user will design his own timing signals to be able to implement a VGA controller as well as design the display itself.

Two PS/2 ports are provided. The user will be able to use these as ports for a keyboard as well as for a mouse.

A USB port is also provided. This USB is already designed to transmit over a USB cable and what is needed from the user are the control signals as well as the data to be sent over the line.

A general input/output port is also provided. Primarily it will be used as a parallel port for parallel communications. It will also be used as an interface for daughter boards to be designed and implemented for additional functions for the motherboard.

For serial communications three DB9 ports are provided. They are designed to emulate communication devices using serial communications such as DTE and DCE devices.

California, USA, 7 Volumes, IEEE Computer Society
2009, Volume 5, pp. 343-347
(ISBN 978-0-7695-3507-4).

3 Acknowledgments

The authors wish to extend their gratitude to the Philippine Council for Advanced Science and Technology Research and Development of the Department of Science and Technology for funding this study.

4 References

- [1] Carlson, A. Bruce, *Communication Systems*, McGraw-Hill, 2002.
- [2] Smith, Doug J., *HDL Chip Design*, Doone Publications, 2001.
- [3] Rosula Reyes, Carlos Oppus, Jose Claro Monje, Noel Patron, Reynaldo Guerrero, and Jovilyn Therese Fajardo. "FPGA Implementation of a Telecommunications Trainer System". *International Journal of Circuits, Systems and Signal Processing*. NAUN Press, Vol. 2, Issue 2, 2008, pp. 87-94. (ISSN: 1998-4464).
- [4] Maxfield, Clive, *The Design Warrior's Guide to FPGAs: Devices, Tools, and Flows*, Elsevier Science and Technology, 2004.
- [5] Brown, Stephen, Zvonko Vranesic, *Fundamentals of Digital Logic with Verilog Design*, McGraw-Hill, 2008.
- [6] Chu, Pong P., *FPGA Prototyping by Verilog Design*, John Wiley & Sons, Inc., 2008.
- [7] Rosula Reyes, Carlos Oppus, Jose Claro Monje, Noel Patron, Raphael Gonzales, and Jovilyn Therese Fajardo. "FPGA-based DSP Trainer," CSIE 2009, 2009 WRI World Congress on Computer Science and Information Engineering, March 31 - April 2, 2009, Los Angeles,