FPGA Implementation of a Telecommunications Trainer System

ROSULA REYES[1,2], Ph.D., CARLOS OPPUS[1,2], JOSE CLARO MONJE[1,2]
NOEL PATRON[1,2], REYNALDO GUERRERO[2], JOVILYN THERESE FAJARDO[2]

[1]Department 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
PHILIPPINES

[2]Blue Chip Designs, Inc.
Unit 201 Xanland Place 323 Katipunan Avenue, Loyola Heights, Quezon City
npatron@bcdph.com, rguerrero@bcdph.com, jtfajardo@bcdph.com
http://www.bcdph.com
PHILIPPINES

Abstract: - Field programmable gate arrays (FPGAs) have been used in a wide range of applications including the field of telecommunications. This paper presents the use of FPGAs in the implementation of both analog and digital modulation that includes amplitude modulation, frequency modulation, phase modulation, pulse code modulation, pulse width modulation, pulse position modulation, pulse amplitude modulation, delta modulation, amplitude shift keying, frequency shift keying, phase shift keying, time division multiplexing and different encoding techniques like non-return-to-zero line code, non-return-to-zero mark line code, non-return to zero inversion line code, Unipolar return-to-zero line code, bipolar return-to-zero line code, alternate mark inversion line code, and Manchester line code. Moreover, an FPGA can be designed to emulate a particular device like an oscilloscope, a function generator, or the like. This paper describes the capability of an FPGA to internally generate a low frequency input signal and through the use of a VGA port, it is able to display the signals in an output device. However, the use of FPGAs is not limited to the aforementioned applications because of its reconfigurability and reprogrammability.

Key-Words: field programmable gate array, telecommunications trainer, digital modulation, analog modulation, encoding techniques, keying

1 Introduction

At present, telecommunications is one of the fast growing sectors in electronics. Because of its fast-paced growth, it is also rising in importance. As a result, more people are pursuing a degree in telecommunications. Because of its high demand, there is a growing need for training systems that effectively teach the fundamental principles of telecommunication systems. However, existing educational tools like telecommunications trainers are quite expensive and most colleges and universities here in the Philippines cannot afford to procure one. Thus, there is a need for a low cost basic telecommunications trainer system to respond to this demand, which will greatly benefit those students in the telecommunications field. Moreover, the availability of these low cost telecommunications trainers in colleges and universities here in the Philippines will be essential to produce globally competitive telecommunications engineers in the near future.

For the remaining sections of this paper, it is organized as follows: Section 2 presents some theories on field programmable gate arrays and its applications, Section 3 discusses the current or commercially-available telecommunications trainers in the market, Section 4 describes the telecommunications trainer, its overall system, its capabilities and other features, and finally, Section 5 gives a summary of the entire telecommunications trainer system.

2 Leveraging on the Benefits of 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.

A field programmable gate array (FPGA) is a semiconductor device containing programmable logic blocks and interconnects [1]. These interconnects are controlled by programmable switches that connect the different logic blocks in any desired configuration to implement a specific application [2]. Applications may include duplicating the functionality of certain digital circuits. Moreover, modern FPGAs can incorporate devices with simple functions like address decoders or multipliers to more complex functions like arithmetic logic units, digital signal processing, and microprocessors [3]. In effect, the trend for FPGAs is to build more intellectual property (IP) cores to facilitate the implementation of system-on-a-chip (SoC) designs [4].

In designing a product using an FPGA, a certain development process is followed. It begins with the specification of the design and ends with programming the target device or the FPGA to be used. The design is specified using a schematic capture tool or writing a source code in a hardware description language (HDL) [5],[6]. Using an HDL is usually preferred over using a schematic capture tool especially in designing large and complex circuits. Two of the HDLs endorsed by the Institute of Electrical and Electronics Engineers (IEEE) that are virtually supported by all vendors are Verilog HDL and VHDL (Very High Speed Integrated Circuit Hardware Description Language). These HDLs describe the functionality of the FPGA. Thus, there is no need to think about the gate level implementation of the circuit. Moreover, computer-aided design (CAD) tools are available that helps the designer or engineer with the rest of the FPGA design process.

With the wide-range capabilities of FPGAs and its 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 [7],[8]. Thus, designing a telecommunications trainer system using an FPGA is also possible and this will be relatively cheap compared to other existing trainers.

With only an appropriate FPGA, an analog-to-digital converter (ADC), a digital-to-analog converter (DAC), and a couple of other analog components, an inexpensive telecommunications trainer can be implemented. For instance, get 12 one-half watt resistors and several lines of HDL, a very cheap 4096-color VGA controller is created. Another is to use the ADC and several lines of HDL code then, an inexpensive oscilloscope-like instrument can be achieved. Again, with some HDL codes and the DAC, a function generator can also be created. External function generators and oscilloscopes are also not needed because these can be generated internally by the FPGA. In addition, internal clocks can also be generated by the FPGA using phase-locked loop (PLL) circuits where most FPGAs today contain 1 to 8 of these. Thus, only basic knowledge, coding skills, and good execution is needed to come up with an innovation.

3 Costly Telecommunications Trainers

Commercially available telecommunications trainers have capabilities that vary from the basic analog modulation to complex digital modulation and demodulation modules. Purchase of these trainers usually includes a mainframe and the choice of additional modules depending on the requirements of the customers. Thus, the overall cost of the trainer varies accordingly, from the most basic to the complete set costing approximately USD 4,500.00 to USD 20,000.00, respectively. In effect, most colleges and universities in the Philippines do not own one.

Moreover, existing telecommunications trainers need additional equipments like oscilloscopes, function generators, power supplies and spectrum analyzers. Oscilloscopes are used to display the output signal while function generators are used to generate an input signal. However, these equipments are purchased separately adding up to the cost of using a telecommunications trainer.

In comparison to the abovementioned prices, the telecommunications trainer developed in this paper shown in Fig.1 cost approximately USD 1,000.00 inclusive of the built-in signal generator and oscilloscope-like display.

Fig.1 BCD telecommunications trainer

4 Telecommunications Trainer System

FPGAs have been used in a wide variety of applications especially in the implementation of digital systems. Some of these systems are aimed to be used for educational purposes either to provide as a substitute for current expensive learning tools or to provide another method of helping the students grasp the theoretical concepts better. An example of such a tool is the 16-bit microprocessor core implemented using an FPGA [9].

Moreover, FPGA devices are also used in teaching digital design courses because it increases the motivation of students and it also improves the learning process [10]. Thus, this serves as a spring board to also implement an FPGA device that teaches different concepts in telecommunications. The following subsections describes a telecommunications trainer system implemented using an FPGA.

4.1 Overall System

The telecommunications trainer system is composed of a control unit, a processing unit, and a display unit as shown in Fig.2. The control unit is a keyboard connected to the PS/2 port of the processing unit. Basically, this is where the user enters his/her input for the trainer. Moreover, the processing unit contains the FPGA plus the different I/Os connected to it. Finally, the display unit is any output device (e.g. monitor, LCD, etc.) that has a VGA port.

(a)

(b)

Fig.2: (a) The telecommunications trainer overall system (b) in detail.

The modules for the telecommunications trainer were written in Verilog HDL and it used Altera's Quartus II project navigator software for the entire design process. Then, the design was downloaded to Altera's Cyclone II FPGA.

The next sections will cover the modules included in the telecommunications trainer.

4.2 Signal Generator

This telecommunications trainer does not need a function generator for its input signals because a function generator can easily be implemented in an FPGA. Thus, a module that emulates a signal generator was implemented. The output for this module serves as the input signals needed for the different modulation and encoding techniques included in this trainer.

For the encoding schemes, its input is a stream of 1's and 0's. This is realized by a 1-bit register that holds a value for a definite duration of time determined by a counter. As shown in Fig.3, the bit stream generator inputs are a clock and a constant value called period. For every clock cycle, the counter (accumulator) increments until it is equal to the period. Then, the output is complemented that is, if the previous output is 1, the current output will be 0, and vice versa.

Moreover, for keying and pulse modulation, its input is a series of discrete levels. This is implemented using a multi-bit register that holds a discrete value for a finite duration of time and it changes to another discrete value after the elapsed time, which is controlled by a counter. These counters can be modified so that the speed of the input changes varies. This is similar to how the bit stream generator was implemented except that the counter is not a 1-bit register but it is a multi-bit register holding a specific value.

Fig.3: Block diagram of a bit stream generator

Further on, the input for analog modulation, pulse amplitude modulation, and pulse code modulation is a sine wave. The sine wave is implemented using Direct Digital Synthesis (DDS) shown in Fig.4. Initially, points of a waveform in digital format are stored in memory. These points are then recalled to generate the waveform using an address counter (accumulator) pointing to the memory. For every clock cycle, the address counter adds up depending on the desired frequency. In this trainer, the structure of the DDS is composed of a read-only-memory (ROM) containing values of one-fourth of a sine wave cycle, an address counter, and a logic circuit to invert and flip the values in the ROM in order to output a full sine wave cycle after running the address counter four times. By modifying the speed and the increment of the address counter, the frequency of the sine wave generated is controlled.

Fig.4: Block diagram of a basic direct digital synthesizer (DDS)

4.3 Implementation of the Modulation and Encoding Techniques

The following subsections show the general concepts behind the implementation of the different modulation and encoding schemes with its demodulator and decoding counterparts respectively. Fig.5 shows the main modules implemented in this telecommunications trainer namely: analog modulation, pulse modulation, keying, and time division multiplexing.

Fig.5: The telecommunications trainer main modules

Under analog modulation, there are three modules implemented as shown in Fig.6. These are amplitude modulation, frequency modulation, and phase modulation.

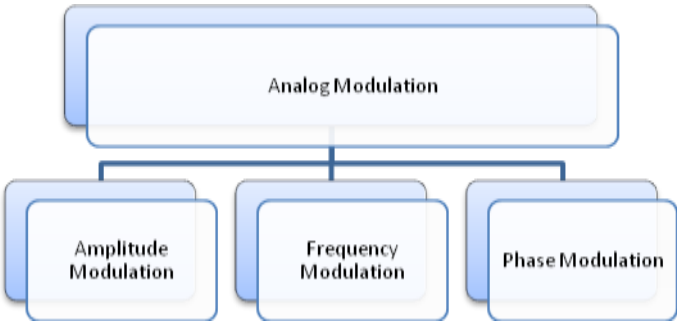


Fig.6: Analog modulation techniques

Moreover, pulse modulation includes the techniques shown in Fig.7 namely: pulse amplitude modulation, pulse code modulation, pulse position modulation, pulse width modulation, delta modulation, and line codes. For line codes, it includes alternate mark inversion, bipolar return-to-zero, Manchester line encoding, non-return-to-zero inversion, non-return-to-zero mark, and unipolar return-to-zero as shown in Fig.8.

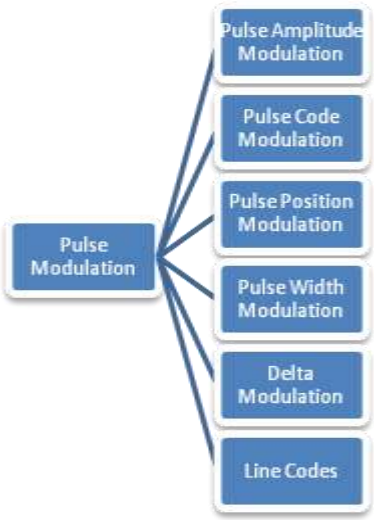


Fig.7: Pulse modulation techniques

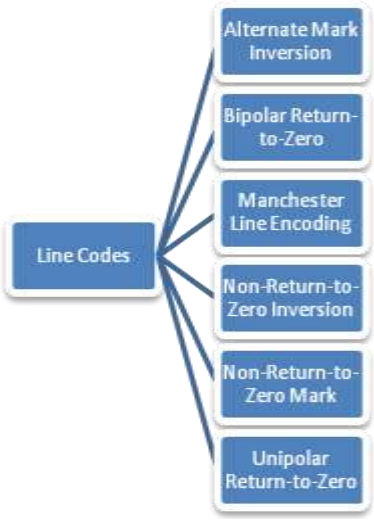


Fig.8: Line coding techniques

Also, under keying, it includes amplitude shift keying, frequency shift keying, and phase shift keying as shown in the Fig.9.

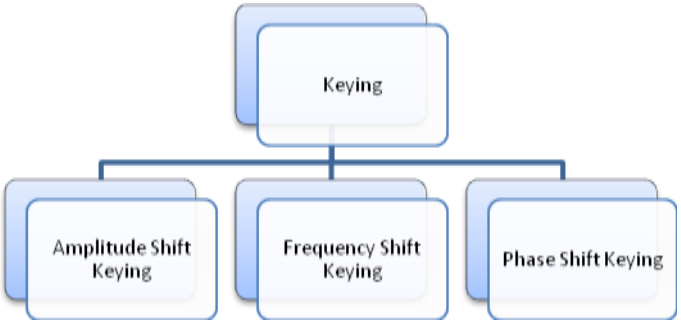


Fig.9: Keying techniques

4.3.1 Amplitude Modulation and Amplitude Shift Keying

Amplitude modulation (AM) transmits analog information by modifying the amplitude of the carrier signal. The input signal is multiplied with the modulation index and the product is added with 1. Then, the resulting sum is multiplied by the carrier signal generated using DDS as shown in Fig.10. With this, AM attributes is changed by modifying the modulation index and the frequency of the carrier signal.

The demodulation for AM is performed by simple peak detection. A logic circuit with comparators is used to determine the peak that is, after the peak is detected, a register holds the value of the peak until another peak is detected. The use of comparators in demodulation was also implemented in other demodulation schemes [11].

Amplitude shift keying also implements AM but the input used is a series of discrete values. Its demodulation also uses peak detection and it works by assigning the output with a discrete value for a range of peak values wherein this range is bounded by threshold values. Thus, if there are eight discrete levels for the input in the modulator, there are nine threshold values in the demodulator.

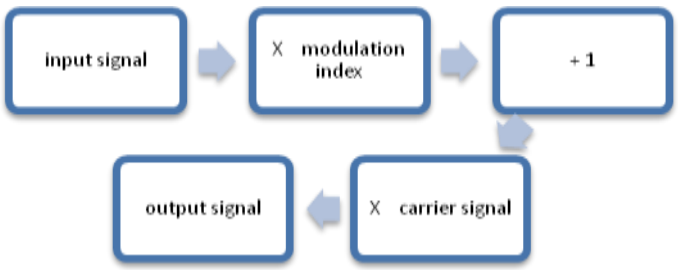
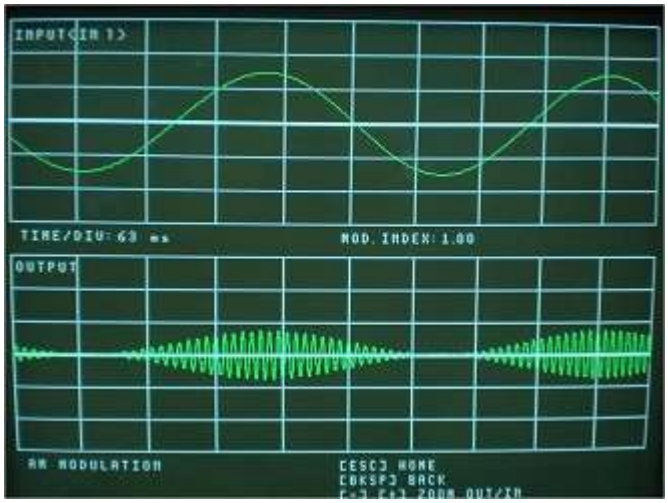
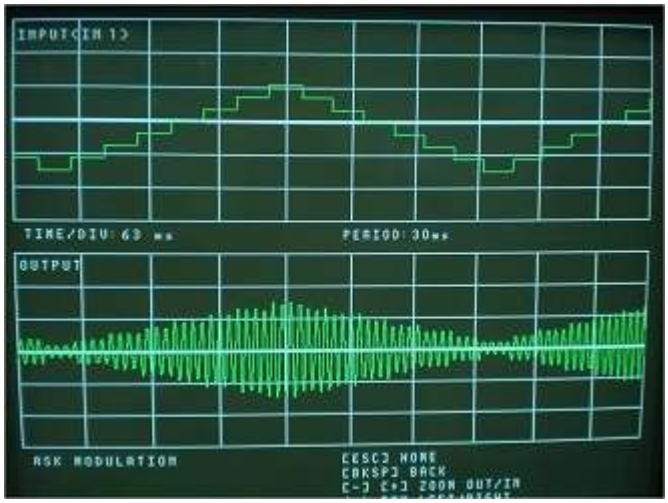


Fig.10: Block diagram of the implementation of amplitude modulation



(a)



(b)

Fig.11: Waveform outputs of (a) amplitude modulation and (b) amplitude shift keying

4.3.2 Frequency Modulation and Frequency Shift Keying

Frequency modulation (FM) modulates a carrier wave by changing its frequency such that the frequency change is proportional to the amplitude of an analog modulating signal. Again, the carrier wave is generated through DDS. Also, the input signal is multiplied by a register containing the desired frequency deviation. The product is then used to configure the address counter of the DDS which results to a varying frequency of the carrier signal as shown in Fig.12.

The demodulation for FM is implemented using a phase-locked loop. The input is multiplied with the output of a numerically-controlled oscillator which is basically a DDS. The product passes through a loop filter and a low pass filter to output the original message signal. The output of the loop filter serves as an input for the numerically-controlled oscillator as shown in Fig.13

Frequency shift keying simply employs FM but it uses discrete inputs. Since the possible values for the demodulated signal are finite, the demodulation is performed by using a counter that increments when the signal is positive. The counter already gives half of the period which is inversely related to the frequency. Thus, the demodulator outputs a discrete value depending on the range of half period counts. Similar to ASK, threshold values for the half period counts are assigned.

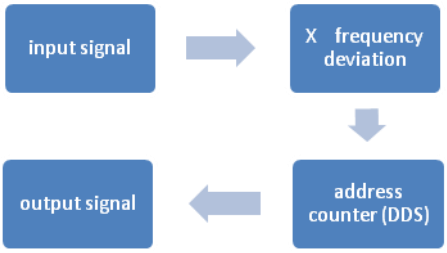


Fig.12: Block diagram of the implementation of frequency modulation

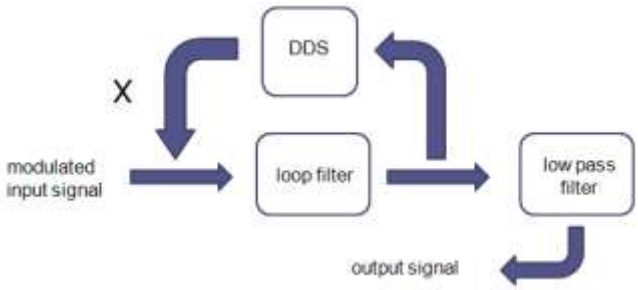
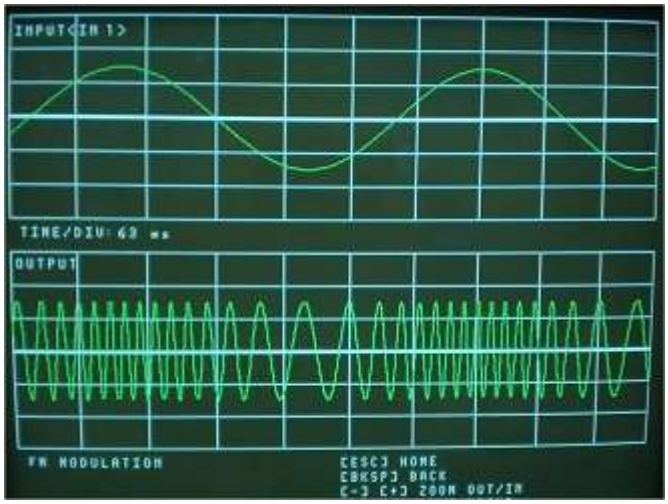
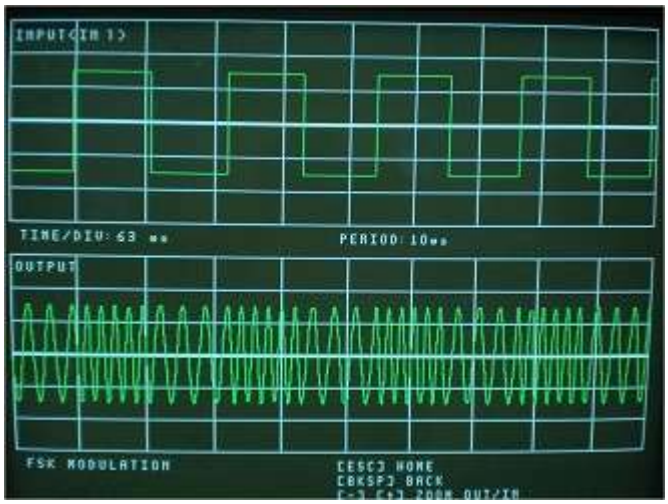


Fig.13: Block diagram of the implementation of frequency demodulation



(a)



(b)

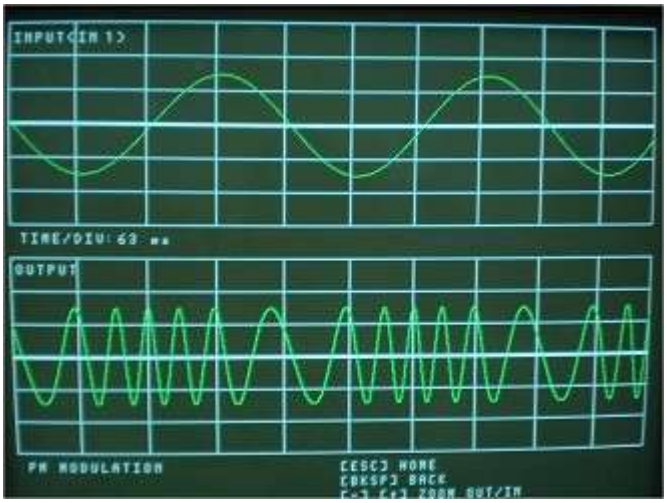
Fig.14: Waveform outputs of (a) frequency modulation and (b) frequency shift keying

4.3.3 Phase Modulation and Phase Shift Keying

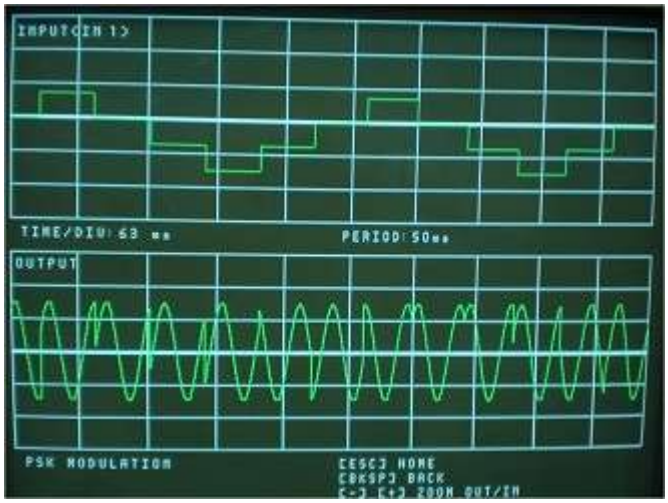
Phase modulation (PM) sends an analog message by adjusting the phase of the carrier signal. The phase of the carrier signal is adjusted by modifying the address counter of the DDS that produces the carrier signal.

The demodulation for PM is implemented by using consecutive values of the input signal to identify the address counter in the modulator. Then, using a phase reference supplied to the demodulator, the phase difference is determined.

Phase shift keying is the same with PM but it uses discrete inputs. Its demodulation is also the same but it uses threshold values of the phase difference to produce the discrete output.



(a)



(b)

Fig.15: Waveform outputs of (a) phase modulation and (b) phase shift keying

4.3.4 Pulse Amplitude Modulation

Pulse amplitude modulation (PAM) transmits data by changing the amplitude of a succession of pulses according to the input. The modulator samples the input, stores the value in a register for a period of time called aperture time, and then returns the value of the register to zero as shown in Fig.16.

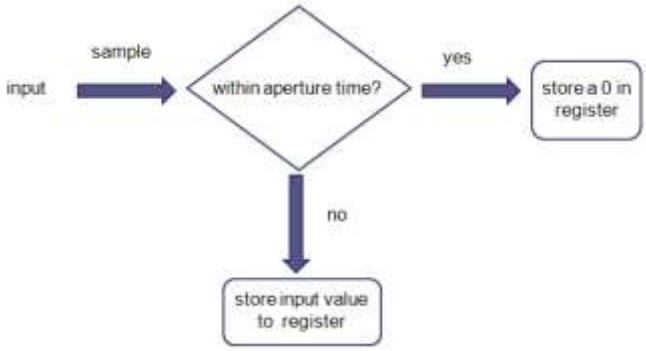


Fig.16: Block diagram of the implementation of pulse amplitude modulation

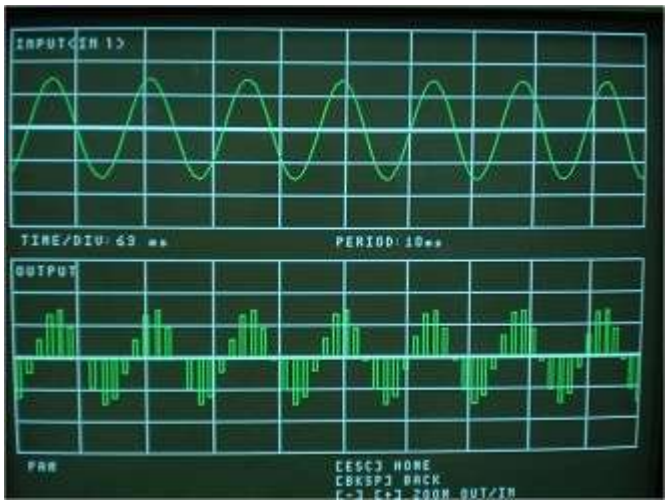


Fig.17: Waveform output of pulse amplitude modulation

4.3.5 Pulse Code Modulation

Pulse code modulation (PCM) is a digital representation of an analog signal wherein the magnitude of the signal is sampled regularly at uniform intervals and quantized to a series of symbols in a digital code (binary in this case). The modulator samples an input and stores it in a register and then, the 1-bit output gets the value of each bit (most significant bit first) in the register. The modulator is configured as to how many bits in the register determine the output. This way the number of bits that represents an input magnitude is modified. For example, if the PCM represents a discrete level with 4 bits, the output reads the 4 most significant bits. Also, the modulator is configured to pad bits before the actual data bits are sent.

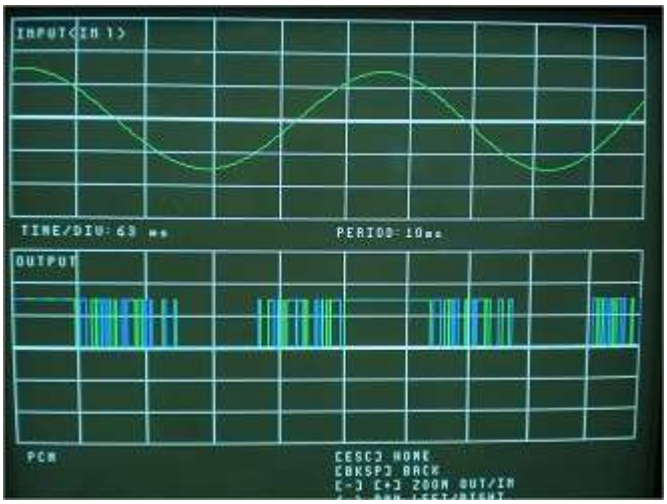


Fig.18: Waveform output of pulse code modulation

4.3.6 Pulse Position Modulation

Pulse position modulation (PPM) encodes a message by changing the position of the pulse. This is simply implemented by running a counter and depending on the discrete input, a pulse is outputted at a particular range of the counter. Its demodulation is performed by having a synchronization signal so that the counter in the demodulator matches with the one in the modulator.

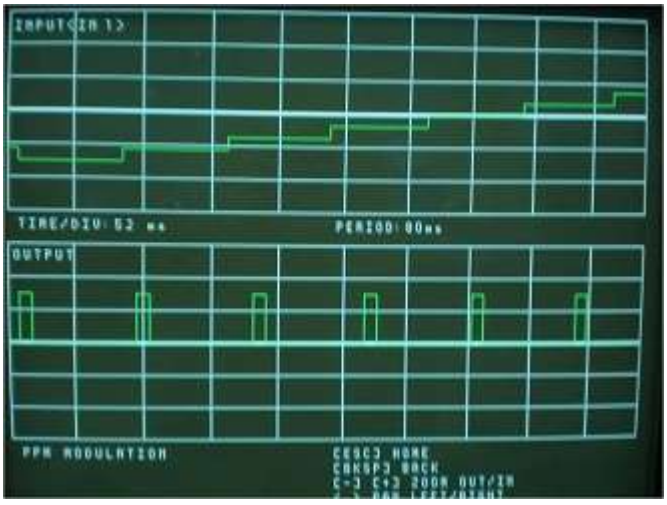


Fig.19: Waveform output of pulse position modulation

4.3.7 Pulse Width Modulation

Pulse width modulation (PWM) stores information by

adjusting the duty cycle of the pulse. The same with PPM, this is simply implemented by running a counter and depending on the discrete input, a pulse is outputted at a particular range of the counter. However, in this case, the lower bound is always zero or the start of the counter. Thus, the duration of the pulse is varied. Demodulation for PWM is implemented by having a synchronization signal.

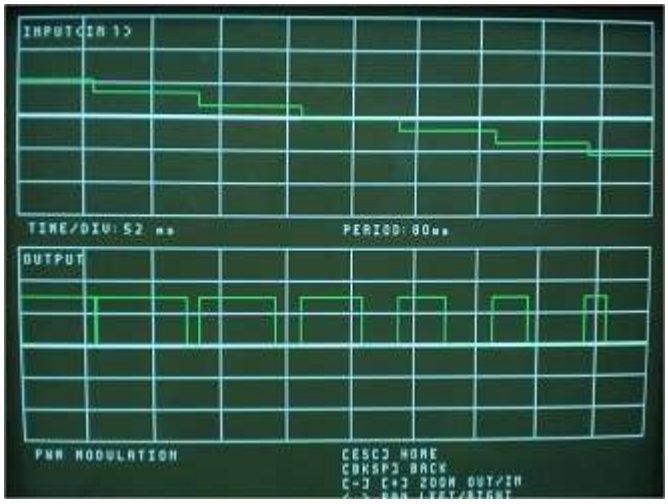


Fig.20: Waveform output of pulse width modulation

4.3.8 Delta Modulation

Delta modulation (DM) is an analog-to-digital conversion wherein the output is not the quantized value of the analog signal (such as pulse code modulation) but only the difference between the previous and current step. Periodically, the modulator gets an input signal and compares it with a reference signal. If the input signal is greater than the reference, the modulator outputs a one and increments the reference signal with a constant called delta. Otherwise, the modulator outputs a zero and decrements the reference signal with the delta. This is shown in Fig.21.

For demodulation, synchronization also takes place. The initial reference value and the delta of the modulator are the same with those in the demodulator. The demodulator outputs the reference plus the delta if the input is one; otherwise, it outputs the reference minus delta.

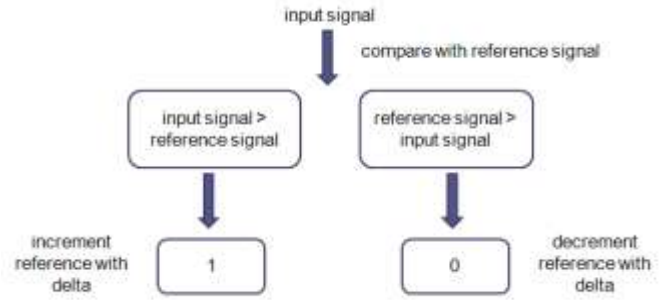


Fig.21: Block diagram of the implementation of delta modulation

4.3.9 Line Codes

Line codes describe how a bit is represented in a single digital transmission link [12]. Its implementation is straightforward from their definition. A multi-bit register is used for the output in order to represent three states in bipolar encoding.

Alternate Mark Inversion (AMI) is a bipolar encoding that represents a “0” with a ground and represents a one bit by alternating positive and negative values having the same magnitude.

Bipolar Return-to-Zero (BRZ) encodes a bit by representing the two possible values with a pulse comprising of a non-zero value (bit equal to one is a positive voltage while a zero is a negative signal) followed by a zero value or ground signal.

In Manchester Line Encoding (MAN), a value of one has a 180° phase shift pulse. In short, a one is represented by a low-high sequence while a zero is represented by a high-low sequence.

Non-Return-to-Zero Inversion (NRZI) encodes a bit by representing a “1” with a transition from one level to another while a no transition means a “0”. Its difference with Non-Return-to-Zero Mark is that the transition happens during the leading edge instead of the trailing edge of the message bit's clock.

Non-Return-to-Zero Mark (NRZM) encodes a bit by representing a “1” with a transition from one level to another while a no transition means a “0”. The transition happens during the trailing edge of the message bit's clock.

Unipolar Return-to-Zero (URZ) encodes a bit by representing the two possible values with a pulse comprising of a non-zero value and a zero value. However, the non-zero value is always followed by a zero value to represent a bit. Usually, a “1” is a positive voltage followed by a zero value while a “0” is a zero

value or ground all throughout.

To ensure that one bit of input is differentiated from another in the encoding and decoding, the counter in the input side that determines the duration of the data is synchronized with the counter used in the encoder or decoder.

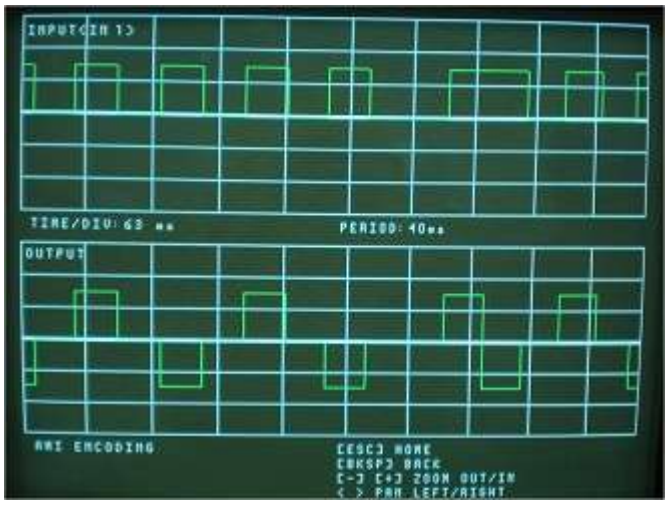


Fig.22: Waveform output of alternate mark inversion

4.3.10 Time Division Multiplexing

Time division multiplexing is a method of sending multiple information or signals in one physical channel [13]. This is done by assigning a fixed time slot for each signal and all signals are taking turns in the channel. One frame is composed of one time slot for each signal. In this method, the different signals are sent simultaneously which is shown in the upper display of the trainer's output device while at the bottom display, it shows the signals taking turns in one channel. Each signal source is represented by a distinct color thus, in the bottom display, different colors are shown corresponding to the color of its signal source as shown in Fig.23.

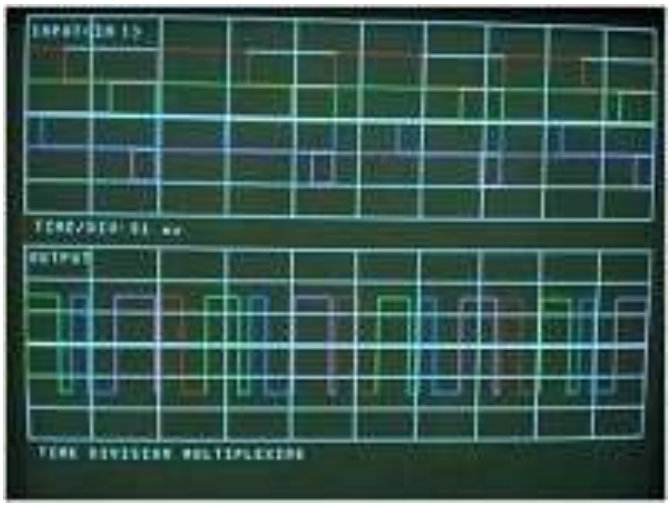


Fig.23: Waveform output of time division multiplexing

4.4 Trainer I/O

The output of this telecommunications trainer is shown in a display device via the VGA port or it can be acquired using the output port (GPIO 0 port) of the trainer. In addition, the trainer is also capable of accepting an external input from another trainer using the input (GPIO 1) port of the trainer. These GPIO ports are simply interconnected using a common IDE cable.

4.4.1 VGA Display

This trainer is capable of displaying the input signals and the output signal of the different modulation and encoding schemes to an output device through the VGA port. There are five signals needed to display an image using the VGA port. Two signals are used to synchronize the video. These are the horizontal sync and vertical sync. Three signals are used to control the color. These are the red signal, green signal, and blue signal [14].

Data for the red, green, and blue (RGB) signals corresponds to the pixel data. This data is saved in an external memory (e.g. 512Kbyte SRAM) connected to the FPGA. Then, data is retrieved from the RAM, 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.

The display of the trainer is a grid-like display having two divisions. The upper grid displays the input signal while the lower grid displays the output signal of the selected modulation shown at the bottom left corner of the output device.

For the input signal, there may be two sources for the signal: internal or external. This is indicated at the upper right of the output device. INT refers to an internal input signal while EXT refers to an external input signal.

The time per division (TIME/DIV) in milliseconds (ms) is also indicated for all the modulation/demodulation techniques. This is displayed at the center right of the output device. Moreover, for some of the modulation/demodulation techniques, the period of the signal in milliseconds (ms) is also indicated.

4.4.2 Using the Output (GPIO 0) Port

The trainer has the option to output the modulated or demodulated signal through the output port (GPIO 0 port) of the trainer. The output signal is in digital form of 8 bits corresponding to pins 0 to 7 with the 8^{th} pin having the most significant bit.

4.4.3 Using the Input (GPIO 1) Port

The trainer is also capable of accepting external input from another trainer. The signal should be in digital form and assigned to the first 8 pins (pin0 to pin7) of the port with the 8^{th} pin having the most significant bit.

In addition, for pulse modulation, a synchronization clock is also needed as an input and it is assigned to the next pin (pin8). However, for PSK and PM demodulation, the reference signal should also be received. The PM reference signal is assigned to the next 8 pins (pin9 to pin16) while the PSK reference signal is then assigned to the next 8 pins (pin17 to pin24). Lastly, to determine if the input to the pins is already valid, another synchronization clock is necessary and it is assigned to pin35.

5 Summary

The FPGA-based telecommunications trainer was implemented using a VGA port to connect to any output device for its display and using a PS/2 port to connect to a keyboard for its control. This trainer includes the basic digital modulation and demodulation techniques like amplitude modulation and demodulation, frequency modulation and demodulation, phase modulation and demodulation, pulse amplitude modulation, pulse code modulation, pulse position modulation and demodulation, pulse width modulation and demodulation, delta modulation and demodulation, line code encoding and decoding (non-return-to zero line code, non-return-to zero mark line code, non-return-to zero inversion line code, Unipolar return-to zero line code, bipolar return-to zero line code, alternate mark inversion line code, Manchester line code), amplitude shift keying, frequency shift keying, phase shift keying, and time division multiplexing.

In addition, this trainer does not need a function generator for its input because the input source may be internally generated or it can come from an external input. This trainer also has zooming and panning capabilities and a pause or run option similar to an oscilloscope. Overall, this FPGA-based telecommunications trainer has a number of capabilities as compared to other existing telecommunications trainer with the added benefit of being inexpensive. Thus, this trainer is an adequate resource material in teaching the basic concepts of telecommunications.

6 Acknowledgements

This trainer was developed by Blue Chip Designs, Inc. and it was funded by the Department of Science and Technology – Philippine Council for Advanced Science and Technology Research and Development (DOST-PCASTRD).

References:

[1] Clive Maxfield, *The Design Warrior's Guide to FPGAs: Devices, Tools, and Flows*, Elsevier, 2004.
[2] Jerry C. Whitaker, *The Electronics Handbook*, CRC Press, 1996.
[3] Bob Zeidman, *Designing with FPGAs and CPLDs*, Focal Press, 2002.
[4] Deming Chen, Jason Cong, and Peichen Pan, *FPGA Design Automation: A Survey*. Now Publishers Inc., 2006.
[5] Stephen Brown and Zvonko Vranesic, *Fundamentals of Digital Logic with Verilog Design*, McGraw-Hill, 2008.
[6] Zainalabedin Navabi, *Digital Design and Implementation with Field Programmable Devices*, Springer, 2004.

[7] Omar Al-Ayasrah and Ayman Al-Lawama, DSP-Based Two-Synchronized Motors Control System Using External FPGA Design, *WSEAS Transactions on Systems and Control*, Vol.3, No.4, 2008, pp. 279-288.
[8] Shabal Islam, Nowshad Amin, M.S. Bhuyan, Mukter Zaman, Bakri Madon, and Masuri Othman, FPGA Realization of Fuzzy Temperature Controller for Industrial Application, *WSEAS Transactions on Systems and Control*, Vol.2, No.7, 2007, pp. 484-490.
[9] Esma Alaer, Ali Tangel, and Mehmet Yakut., "MIB-16" FPGA Based Design and Implementation of a 16-Bit Microprocessor for Educational Use, *WSEAS Transactions on Advances in Engineering Education*, Vol.5, No.5, 2008, pp. 326-330.
[10] Antonio J. Araujo and Jose C. Alves, A Project Based Methodology to Teach a Course on Advanced Digital Systems Design. *WSEAS Transactions on Advances in Engineering Education*, Vol.5, No.6, 2008, pp. 437-446.
[11] Philippe Dondon, J.M. Micouleau, J. Legall, P. Kadionik Enseirb, and A. Schweitzer, Design of a Low Cost BPSK Modulator/Demodulator for a Practical Teaching of Digital Modulation Techniques. In *Proceedings of the 4th WSEAS/IASME International Conference on Engineering Education*, Agios Nikolaos, Crete Island, Greece, July 24-26, 2007.
[12] Robert J. Schoenbeck, *Electronic Communications: Modulation and Transmission*, Macmillan Publishing Company, 1992.
[13] A. Bruce Carlson, *Communication Systems*, McGraw-Hill, 2002.
[14] James O. Hamblen, Tyson S. Hall, and Michael D. Furman, *Rapid Prototyping of Digital Systems: Quartus® II Edition*, Springer, 2006.