ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA079

# TIMING SYSTEM USING FPGA FOR MEDICAL LINEAR ACCELERATOR PROTOTYPE AT SLRI

P. Koonpong†, R. Rujanakraikarn,

Synchrotron Light Research Institute, Nakhon Ratchasima, Thailand

*Abstract*

A prototype of medical linear accelerator is under dvelopment at Synchrotron Light Research Institute (SLRI). In order to maintain the proper operation of the machine, the pulse signal is used to synchronize the various subsystems such as electron gun, RF trigger, and magnetron trigger subsystems. In this project, we design the timing system using a Xilinx Spartan-3 FPGA development board with VHDL in order to achieve the desired characteristics and sequences of the timing signals for those subsystems. A LabVIEW GUI is designed to interface with the timing system in order to control the time delay and pulse width via RS-232 serial interface. The result of the system design is achieved with the pulse resolution of a 20 nsec per step for four timing channels. The time delay and pulse width for each channel can be set independently based on the SYNC reference signal.

## INTRODUCTION

Synchrotron Light Research Institute (SLRI) in Thailand has development a prototype of the 6 MeV medical linear accelerator (medical linac) for cancer treatment. This project has been proposed to help increase the availability of low-cost radiotherapy machines in Thailand. Reverse engineering approach has been employed in this project via a donated machine to develop a prototype of this machine. The linear accelerating structure has operating frequency at 2,998 MHz, a 3.1 MW magnetron driven by a solid-state modulator, and a hot-cathode electron gun. A drive stand and a gantry of the prototype provide housing for the modulator cabinet for magnetron and electron gun and automatic frequency control (AFC) system in order to provide resonant frequency tuning for the magnetron.

A linac treatment head consists of an X-ray target attached to the end of the accelerating structure. A dosimetry control system is designed to measure and control the X-ray beam dose by processing the signals coming from an ionization chamber installed in the treatment head. Three-stage treatment beam collimators are used to adjust the shape and size of the beam. The collimators are 1) fixed primary collimators, 2) secondary collimators, and 3) a multi-leaf collimator (MLC). A timing system is used to link various subsystems of the machine in order to provide proper synchronization of the X-ray beam generation.

A central control system software is designed to give access and control all subsystems. It is run on the Main Control System with a display monitor and GUI. All of the subsystems are connected to the Main Control System in a private network as shown in Figure 1 [1].

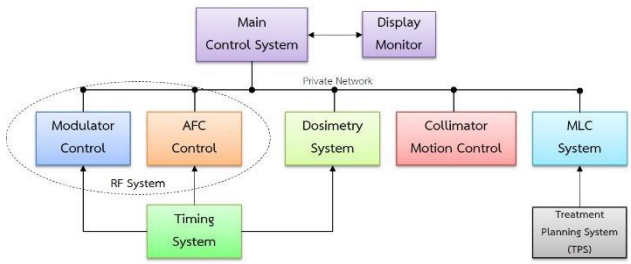


Figure 1: Network diagram of the machine prototype [1].

This paper describes a timing system using FPGA for the medical linac project. System design, both hardware and software, is explained in the next section. The following section discusses some result and discussion. Conclusion is presented in the final section.

## SYSTEM DESIGN

In order to produce the required X-ray beam efficiently, the timing system that provides synchronization of the machine subsystems is indispensable. The accuracy of the timing signals connecting the subsystems is very crucial. Each signal has its own specific timing requirement. This section describes the main design and implementation, both hardware and software, of the timing system developed in this project.

### *Hardware*

For this project, Xilinx's Spartan 3 FPGA development board [2-3] is chosen to be the main hardware for the timing system. The main clock of the system runs at 50 MHz which is the primary reference clock signal of the FPGA. This contributes to the resolution of a 20 nanoseconds (nsec) in the time domain characteristics of the timing signals synthesized by the digital circuit in the FPGA. Since synchronization of all subsystems of the medical linac depends on the pulse timing signals connecting them, we can choose to set up a pulse width and a delay time of the individual signals independently at the outputs.

In the first version of this system, there are four independent output channels, SYN, Electron Gun, RF Trigger, and Magnetron Trigger. For interface between the FPGA board and PC, the serial communication through RS-232

† pratya@slri.or.th

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA079



is designed to transfer the required parameters of the timing signals. These parameters are master clock, pulse width, and delay time, which independent among all four signals. In fact, they depend solely on the main FPGA clock. The voltage level of the output signals coming out of the FPGA board is adjustable by the use of high speed opto-isolator. This can prevent the FPGA from any damages from electrical signals connected to the hardware of other subsystems.

### *Software*

There are two main parts of the software in the system design, digital circuit design part and graphic user interface or GUI part. The digital circuit design part is developed using VHDL [4]. There are three main digital circuit blocks generated in the FPGA fabric, UART transceiver, data splitter, and pulse width and delay generator. Figure 2 illustrates the block diagram of the designed system

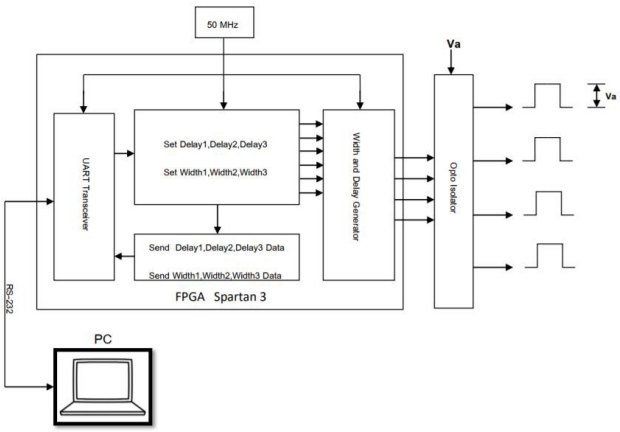


Figure 2: Block diagram of the timing system.

There are four parts in the UART transceiver, receiver, transmitter, baud generator, and buffer. To follow a standard practice for asynchronous communication, the sampling rate of UART transceiver is chosen to be 16. Since the baud rate of 19,200 is selected for the transmission, the UART sampling rate is equal to 19,200 x 16 = 307,200. As the main clock of the Spartan 3 FPGA runs at 50 MHz, the baud rate is therefore obtained from the mod-163 counter circuit ($\frac{50\ MHz}{307200\ baud} = 162.7604$). The data splitter functions as a data block separator in order to define constraints for pulse width and delay calculation of each of the timing signals. The constraints are then passed to the pulse width and delay generator in order to generate a required pulse signal whose timing characteristics are defined by counting a number of pulses of the main clock. Note that the master clock, whose frequency is at 200 Hz, is used as a reference for constraints determination from the data splitter. The output voltage of the pulse timing signal is equal to an external supply voltage, $V_a$.

In serial communication via RS-232 with the chosen sampling rate of 16, each serial bit is sampled 16 times. Suppose there are N data bits with a number of M stop times in the communication. The following steps explain how the UART transceiver works:

1. Wait for an incoming start bit whose logic level is low or '0'. Then, the system allows the counter to start counting.
2. When the count reaches the middle of the data bits where the counter value is 7, the system clears the counter value and restart the counting process.
3. When the counter value is 15 the system stores all the data bits into a data register and restart the counting process.
4. Repeat part 3 again N-1 times to receive the other N-1 data bits.
5. If parity check is implemented, repeat part 3 again to receive the parity bit.
6. Repeat part 3 again M times to receive a number of M stops data

Figure 3 shows a flow diagram of the UART transceiver in the FPGA. A detailed diagram of the FPGA logic blocks and a standard one-byte serial data transmission are shown in Figure 4 and Figure 5, respectively.

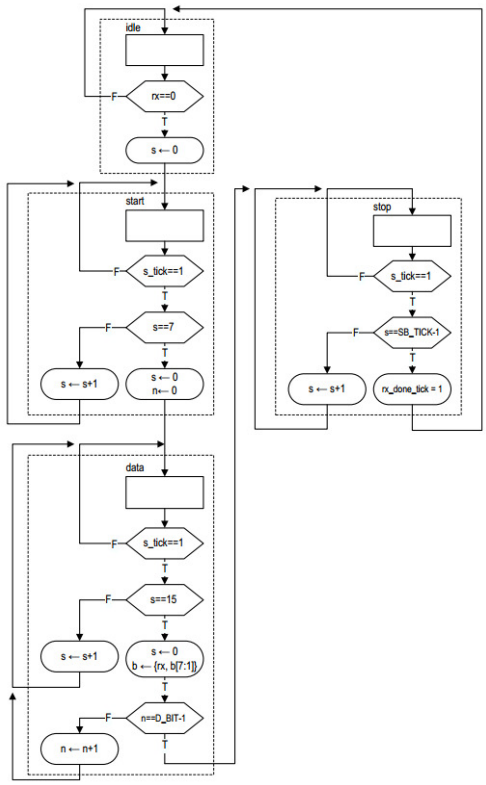


Figure 3: Flow diagram of the UART transceiver.

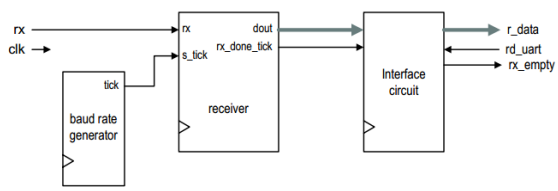


Figure 4: FPGA logic blocks of UART transceiver.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA079

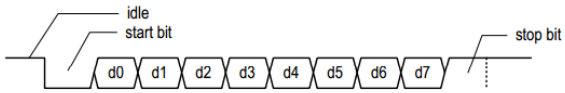


Figure 5: Standard one-byte serial data transmission.

The graphic user interface on PC is developed using LabVIEW 2015. The main purpose is to send the pulse width and the delay time of each timing channel independently to the FPGA via RS-232. The first version of the main GUI is shown in Figure 6.

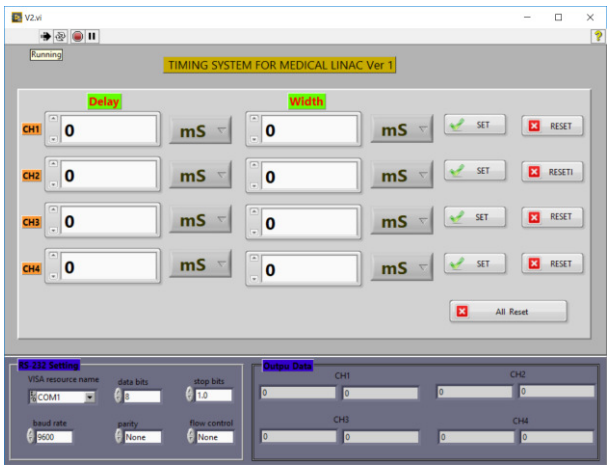


Figure 6: Graphic user interface of the timing system.

## EXPERIMENTAL RESULTS

Time domain characteristics of each of the timing signals is the main specification that needs to be verified. Figure 7 shows how the experiment was set up. All four output channels of the Spartan 3 FPGA development board were connected to the high speed opto-isolator circuits on the custom-made board. DC power supply was used to adjust the required output voltage of the timing signals. All the timing signals were connected directly to an oscilloscope. The pulse width of the reference clock was also verified. Its oscilloscope capture was taken and is shown in Figure 8.

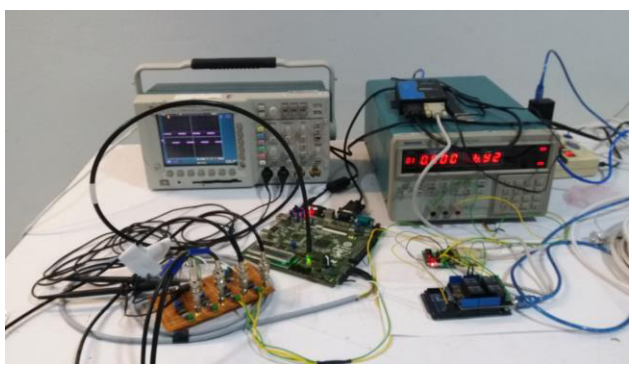

Figure 7: Experiment set up for the timing system.

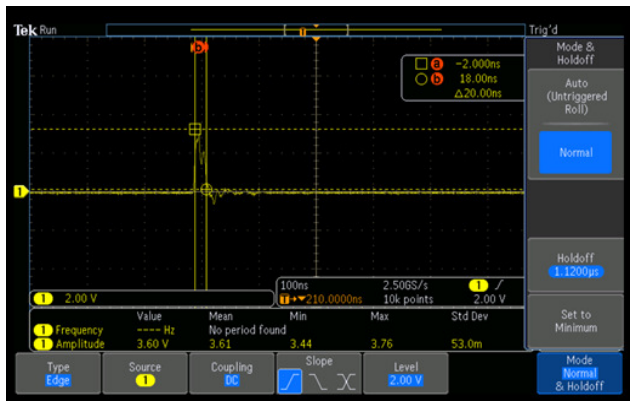


Figure 8: Oscilloscope capture of the pulse width of the reference clock.

Figure 8 shows that the reference clock shape is nearly a perfect rectangle as expected. This means that it can be used as an input to the pulse width and delay generator in order to calculate and generate the correct pulse width and delay time of the required timing signals. The following test was performed to generate four timing signals. All settings can be entered via the GUI which are listed in Table 1.

Table 1: Parameter Settings of the Timing Signals in the Experiment

| Channel | Required Values | |
|---|---|---|
| | Width (nsec) | Delay (nsec) |
| CH1 | 100 | 0 |
| CH2 | 200 | 50 |
| CH3 | 300 | 100 |
| CH4 | 400 | 150 |


Figure 9: Result of the timing signal generation.

From the test, all four outputs from the timing system are shown to be as expected. The oscilloscope capture of the four timing signals is shown in Figure 9. Their shape are nearly perfect rectangular pulses. This shows that they can be used to trigger the operation of various subsystems of the medical linac. In fact, three different timing signals from this system were already used in the machine tests in order to try to generate the beam. As shown in Figure 1, two of the timing signals were connected to modulator

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA079



cabinet to trigger magnetron and electron gun modulators, and the other signal was connected to the AFC system to help sample the needed signals. The result of the multiple tests was satisfactory as it helped trigger the operation of the modulators and AFC. The detail of the machine test result will be discussed in the next technical papers.

## CONCLUSION

The timing system using FPGA is designed and developed using VHDL in order to generate the triggering timing signals for various subsystems of the medical linac machine. The Xilinx's Spartan 3 FPGA development board is used as the main controller board. The system clock runs at 50 MHz, generating the timing resolution of a 20 nanoseconds for each step size of the generated pulse timing signals. As a result of implementing the system with FPGA which allows concurrent operation of multiple timing channels, the pulse width and delay time of all four timing signals can be set independently. Three main digital circuit blocks, UART transceiver, data splitter, and pulse width and delay generator, are designed to generate and control the required timing signals. The main LabVIEW GUI is used to send the values of pulse width and the delay time of each timing channel via RS-232 to the FPGA.

From multiple system tests, the reference clock is found to have the required characteristics. With proper design of the three main digital circuit blocks, all of the generated timing signals have good characteristics as required and expected. This proves that the designed timing system is suitable to provide synchronization of the various subsystems of the medical linac prototype being developed.

## REFERENCES

[1] R. Rujanakraikarn, "FPGA-Based Motion Control System for Medical Linear Accelerator Development at SLRI", presented at the 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS 2017), Barcelona, Spain, Oct. 2017, paper TUPHA069, this conference.

[2] http://www.xilinx.com

[3] http://store.digilentinc.com

[4] K. L. Short, *VHDL for Engineers*, Pearson Ed. 2009.