IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 1, Issue 6, July 2016

FPGA Implementation Of Real Time Controller Based Flight Termination System

[1] Rashmi.V.R, [2]Mrs,Anooja.B
[1] Mtech Student Department Of ECE, Malabar College Of Eng & TECH,
[2]Assistant Profosser, Malabar College Of Eng & TECH

Abstract: ***This paper proposes a demonstration of FPGA platform based digitally implemented Flight Termination System (FTS).The applied design procedure replaces a multiple platform based system with a single platform. It also guarantees reconfigurable, interoperable, portable and handy FTS and maintains errorless, bug free and reliable implementation. Real-time flight termination operation demands a very highly reliable and ruggedized platform. Hence, the FTS is implemented in FPGA. In order to minimize hardware resources and to enable future up-gradation, efficient optimization technique has been applied. Xilinx ISE, a software tool is used for synthesis and analysis of VHDL design. Modelsim, a multi-level high definition language simulation environment is used to simulate and enables system level testing. Validation and simulation is done at various intermediate stages of processing and is found to be utilized in real time environment with further upgradation.***

Keywords: FPGA, Flight Termination, Real Time Controller (RTC),UART

I. INTRODUCTION

For satisfying all expected performance regardless of many uncertainties, a newly developed Flight Vehicle (FV) needs rigorous testing. In some cases, while under test, termination of Flight Vehicle (FV) is mandatory if the high speed vehicle deviates from its preset trajectory and takes different course due to unpredictable failures of on- board system. Hence a Flight Termination System (FTS) is utilized in order to secure the property and human life.

The design of FTS involves signal generation, transmission, reception and analysis of transmitted and received signal. In such cases specific commands are transmitted from a Command Transmission System (CTS) to Command Reception System (CRS) by Remote Control Unit (RCU). Mostly, controlling the FV to make it in correct track is more important than termination of Flight Vehicle (FV) unless crucial situation occurs.

Flight Termination System (FTS) basically involves Real Time Controller (RTC), transmission and reception units. High robustness and high reliable flight termination operations are strongly desired. High degree of versatility and reconfigurable architecture with real-time response is achievable through the combination of highly reliable software with wide range of capabilities and suitable hardware platform i.e. FPGA.

The FPGAs are broadly used now-a-days for many applications. Because of efficient computational power, parallel data processing, low power consumption and technologies include as many as dedicated digital signal processor (DSP) cores to fit several order complex

algorithms to be implemented. Further it is necessary to optimize the design algorithms for optimal resource utilization. In order to ensure future up gradation and versatility a suitable hardware platform ie; Xilinx Spartan 3 FPGA is used for analysis and implementation of FTS algorithm. The waveforms and their portability for transmission and reception are realized through basic functions and libraries provided with VHDL coding and Modelsim 5.7 is the simulation environment.

Universal Asynchronous Receiver Transmitter (UART) is a kind of serial communication protocol. The UART implemented with VHDL language can be integrated into the FPGA to achieve compact, stable and reliable data transmission.

The structure of this paper is as follows. Section 2 shows the architectural details of each work done for the proposed FTS. Section III explains the analysis as well as simulated results. Section IV Finally, concluded the paper with a scope for future works.

II. ARCHITECTURE

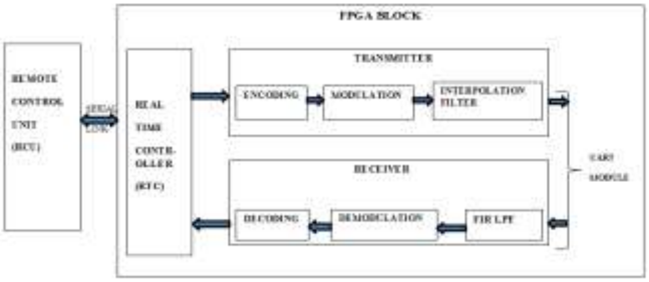


Fig: proposed method block diagram

The RTC offers a communication interface between RCU and host computer. RCU is used for remote operation of the CTS. Host computer is used for configuration, local mode operation and analysis of intermediate signal processing of the CTS system. The RCU is connected with the CTS through reliable serial communication links. The commands are given from the RCU to CTS. The commands are generated and transmitted from the CTS, received by on-board CRS required operation is done accordingly. Single platform implementation of transmitter-receiver and signal generation in controller is done here in FPGA.

The generated commands are a frame of binary bits. This frame comprises of 8 bit binary data, headed by start bit and trailed by stop bit. Transmission of each bit of data and their reception are provided by UART module.

FPGAs are presently taking over legacy processor based applications. Current development in FPGA is number of stop bits. So the length of the frame is N= (n+h+s). The whole frame is encoded with Manchester encoding scheme and total number of bits in a frame becomes double i.e.2N .

The frame is converted to binary data wave form d(t) with the logic d(t) = +1 or -1 corresponding to the binary bits 1or 0. Hence, the binary data waveform d(t)can be expressed as:

$$d(t)= \sum bi\ P(t - i.Tb) \quad \text{where } i= 0 \text{ to } 2N-1 \qquad (1)$$

Here bi is the ith number of bit of the frame. The bit bi takes on value 1 and -1 for binary bit 1 and 0 respectively. P (t) is a rectangular pulse which can be represented as P(t)=rect(t/Tb). Here Tb is the bit duration.

This binary waveform d(t) modulates a carrier using binary frequency shift keying (BFSK) modulation scheme. In this scheme sinusoidal signal with higher frequency ($\grave{\omega}H$) and lower frequency ($\grave{\omega}L$) are generated for bit 1 and 0 respectively. Thus, the BFSK modulated signal can be represented as

$$BFSK=A \cos (\grave{\omega}0t+d(t).\Omega.t) \qquad (2)$$

Here Ω is a constant offset from the carrier frequency (w0). Equation (2) can be written as (3):

$$BFSK=A \cos (\grave{\omega}0t+d (t).\Omega.t) =m (t) \qquad (3)$$

Here for bit 1, d (t) =+1 and its corresponding higher frequency ($\grave{\omega}H=\grave{\omega}0+ \Omega$) and for bit 0, and its corresponding lower frequency ($\grave{\omega}L- \Omega$).

BFSK modulated signal is interpolated in order to increase the sampling rate. The data is then transmitted using UART module of the FPGA. The UART serial communication module is divided into three sub-modules: the baud rate generator, receiver module and transmitter module

A. Baud Rate Generator

Baud Rate Generator is actually a kind of frequency divider. The baud rate frequency factor can be calculated according to a given system clock frequency and the required baud rate. The calculated baud rate frequency factor is used as the divider factor. the system clock is 50MHz, baud rate is 9600bps, and then the output clock frequency of baud rate generator should be 1* 9600Hz. Therefore the frequency coefficient (M) i.e. counts value of the baud rate generator is: M =50MHz/1*9600Hz=5208 When the UART receives serial data, it is very critical to determine where to sample the data information. The ideal time for sampling is at the middle point of each serial data bit.

B. Receiver Module

During the UART reception, the serial data and the receiving clock are asynchronous, so it is very important to correctly determine the start bit of a frame data. The receiver module receives data from RXD pin. RXD jumps into logic 0 from logic 1 can be regarded as the beginning of a data frame. When the UART receiver module is reset, it has been waiting the RXD level to jump. As we know, the ideal time for sampling is at the middle point of each serial data bit. Hence, RXD low level lasts at least half of receiving clock cycles is considered start bit arrives. Once the start bit been identified, from the next bit, begin to count the rising edge of the baud clock, and sample RXD when counting. Each sampled value of the logic level is deposited in the register parallel_data_signal by order. When the count equals 10, all the data bits are surely received, also the 10 serial bits are converted into a byte parallel data and deposited in the resister parallel_data.

C. Transmit Module:

The function of transmit module is to convert the sending 8-bit parallel data into serial data, adds start bit at the head of the data as well as stop bits at the end of the data. When the UART transmit module is reset by the reset signal, the transmit module immediately enters the ready state to send. In this state, the 8-bit parallel data is read into the Shift register. The transmitter only needs to output 1 bit every bit count.

At the receiver side a 4 tapped FIR low pass filtering gives a filtered output to the demodulator. The filter response is multiplied and delayed with co-efficient calculated with hamming window. The demodulation by BFSK gives and output with small decrease in operating frequency compared to modulation. It is then decoded where the Manchester only provides balancing the DC level.

The real time controller module determines whether to transmit, receive or to be in the idle state. Commands or controls are generated whenever needed from remote control unit (RCU).

III. ANALYSIS AND SIMULATION RESULTS

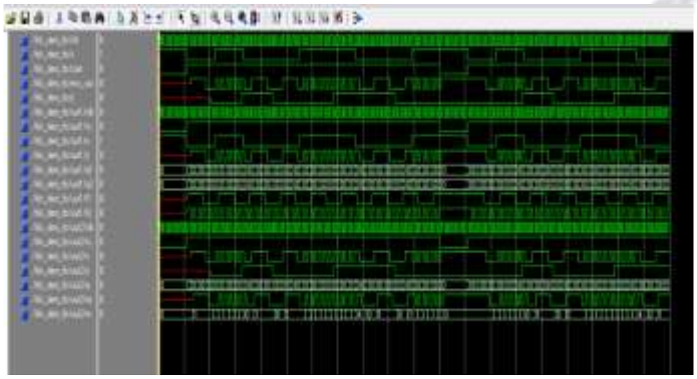


Fig :modulation and demodulation

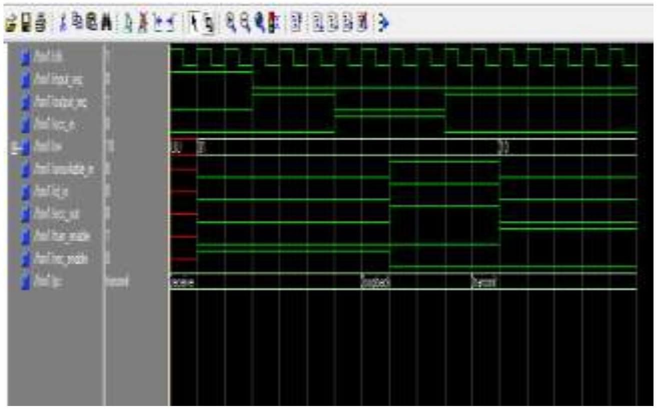


Fig RTC

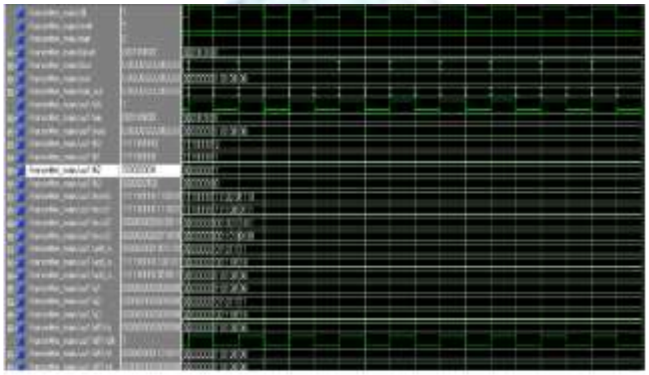


Fig Transmission

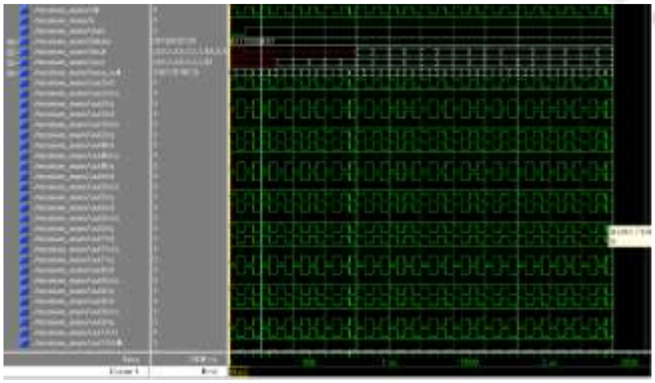


Fig: Receiver

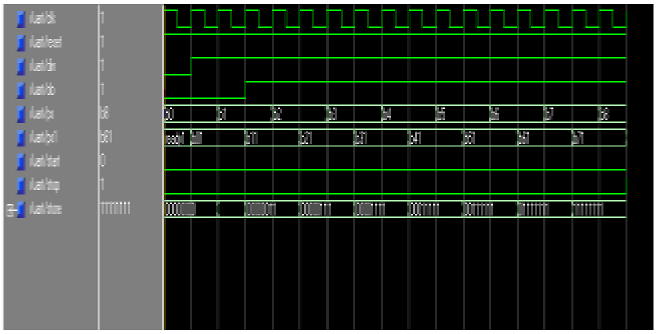
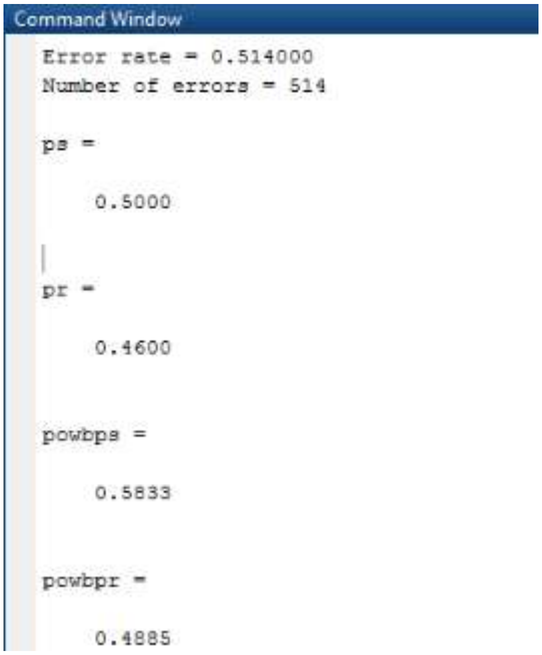


Fig UART

Fig: filter



```
Error rate = 0.514000
Number of errors = 514

ps =

    0.5000


pr =

    0.4600


powbps =

    0.5833


powbpr =

    0.4885
```

Figures :Error calculated with Matlab

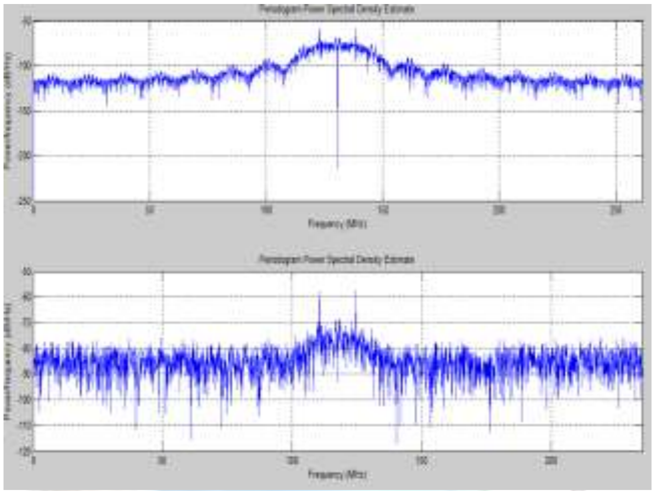
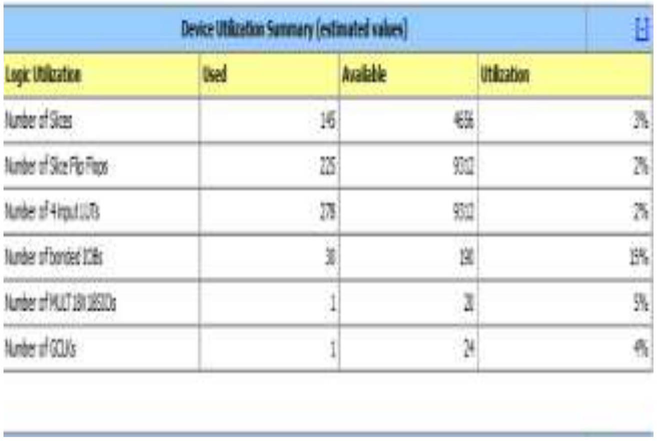


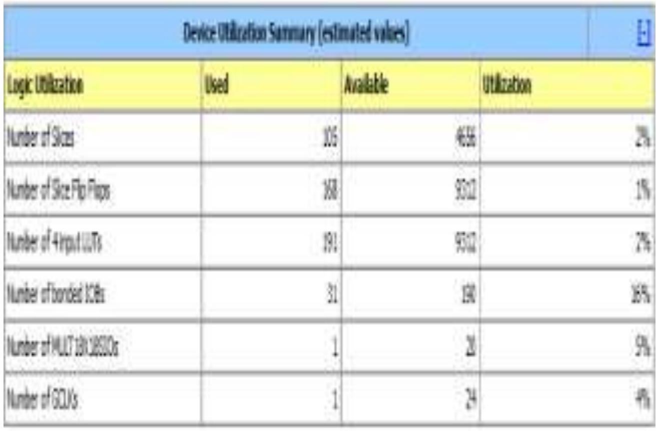
Fig: power spectrum

TABLE:

OPERATING FREQUENCY	
TRANSMISSION UNIT	261.438MHZ
RECEPTION UNIT	141.091M
UART MODULE	17.360MHZ



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	145	4656	3%
Number of Slice Flip Flops	225	9312	2%
Number of 4 input LUTs	278	9312	2%
Number of bonded IOBs	36	190	15%
Number of MULT18X18SIOs	1	20	5%
Number of GCLKs	1	24	4%



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	105	4656	2%
Number of Slice Flip Flops	168	9312	1%
Number of 4 input LUTs	191	9312	2%
Number of bonded IOBs	31	190	16%
Number of MULT18X18SIOs	1	20	5%
Number of GCLKs	1	24	4%

Figures:device utilization(transmitter,receiver)

Analysis is done at each intermediate stages of processing and simulated waveforms are obtained. Modelsim 5.7 is the simulation environment.xilinx ISE provides a platform for analysis of data to find the device utilization and operating frequency. Finally, transmission as well as receiver blocks are combined to get a final analysis.

At modulation stage matlab platform is utilized to fin the power spectrum and error rate.
Transmission and reception is done using UART protocol which is a reliable module for FPGA communication which is done serially.

Standard frequency for Flight communication system is 121.5-123.5MHZ. Here the operating frequency of transmission module is 261.438MHZ . For the sake of easiness the propsed methoduses UART for communication. The whole programme is burned to FPGA which provides a viewability for control.

IV. CONCLUSION

A Flight Termination System(FTS) is a fully redundant turn key range safety and test system for remote control and flight termination of airborne test vehicles. The system is fully programmable and is flexible to meet the changing requirements with Spartan 3 FPGA.

Here complete implementation of each block is done digitally taking the concept of SDR(Software Defined Radio).
As a future work if UART module is replaced with antenna complete utilization of operating frequency can be made. Also if BFSK is replaced with BPSK eror rate can be reduced upto 5 times with reduced delay and device utilization compared to BFSK.

REFERENCES

1. FPGA Implementation of Software Defined Radio Based Flight Termination System Amiya Ranjan Panda, *Student Member, IEEE*, Debahuti Mishra, and Hare Krishna Ratha, *Member, IEEE* 2014

2. W.H.W. Tuttlebee, “Software-defined radio: facets of a developing technology,” *IEEE Pers. Commun.*, Vol. 6, No. 2, pp. 38-44, 1999.

3. An Autonomous Flight Safety System James B. Bull1 and Raymond J. Lanzi2 NASA/Goddard Space Flight Center/Wallops Flight Facility, Wallops Island, VA, 23337