Design of FPGA Controller for Linear and Mixed Signal Test System Using VHDL

Tittu S L
Dept. Of Electronics and Communication Engineering
Sree Buddha College of Engineering, Pattoor, Alappuzha, Kerala, India

Abstract

The necessity for testing systems is quite versatile and they can reliably test a variety of different circuits. Nowadays, many of the linear and mixed signal devices such as ADC, DAC, MUX etc. are evaluated by traditional Automatic Test Equipment. But they have some limitations with respect to high speed data acquisition, high speed pattern generators etc. In order to overcome these limitations, a new high performance main controller is used in the test system architecture which commands the other modules through a high speed backplane bus. This paper presents the design of the controller using Xilinx FPGA, which accepts test commands from PC through PCI (Peripheral Component Interconnect) bus. The functions of PCI core are controlling and sequencing the various functional units such as Analog and Digital PMUs (Parametric Measurement Units) via SPI (Serial Peripheral Interface) bus, TDC (Time to Digital Converter), DDS (Direct Digital Synthesizer), ADC (Analog to Digital Converter), Test circuit interface etc., and then getting back the response from these units and sending them back to PC through PCI for data analysis by which DUT's pass/fail is ensured. USB (Universal Serial Bus) controller is also developed for the communication with Digital Signal Processor. The code is written using VHDL.

1. Introduction

Automatic or Automated Test Equipment (ATE) is any apparatus that performs tests on a device, known as Device Under Test (DUT) or Unit Under Test (UUT), using automation to quickly perform measurements and evaluate the test results. ATE is widely used in the electronic manufacturing industry to test electronic components and systems after being fabricated. ATE is also used to test avionics and the electronic modules in automobiles. It is used in military applications like radar and wireless communication. Nowadays, linear and mixed signal devices like ADCs, DACs, PLL, DC/DC converters, relay drivers, transistors, diodes, opamps, regulators, line drivers or receivers etc. are evaluated by traditional ATEs so that they have limitations in generation of precise input voltages/currents, high speed data acquisition, high speed pattern generators etc. So the test system configuration is introduced with a main controller which controls the overall functions of test system in a defined and programmed manner. All other sub consoles are interfaced to main processing console through a high speed parallel backplane bus interface. Here PCI (Peripheral Component Interconnect) bus has been chosen. The design of main controller is presented in this paper.

2. Linear and Mixed Signal Test System

This is a PC based test system with simple hardware and software subsystem modules for easy portability and testability. It is divided into three major modules as System module, Family board module and the DUT module.

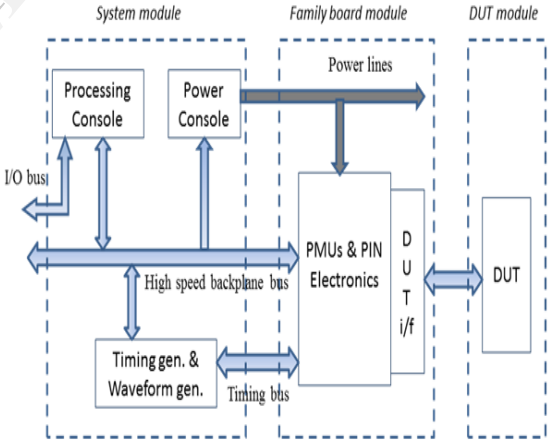

Figure 1. Block diagram of the test system

The system module contains three consoles as processing console, power supply console, timing and waveform generation units. These blocks are common to all class of semiconductor devices. In family board module, necessary PIN electronics with PMUs and specific test circuits are housed. So, for each category of device, one family board is to be designed only with required number of PIN electronics, PMUs and test circuits for characterizing it. DUT Module houses only the device which is to be tested. All the required input and output signals will be thoroughly handled or managed by test system through DUT interface module which will be associated with family board.

3. FPGA Controller

This paper presents the design of an FPGA controller for linear and mixed signal test system. The functional configuration of FPGA controller is as shown in Figure 2. A Family board module consists of Time to Digital Converter (TDC), Direct Digital Synthesizer (DDS), Digital Parametric Measurement Unit (D_PMU), Analog Parametric Measurement Unit (A_PMU) and test circuit. All these units except the test circuit are common for all the Family boards. FPGA communicates with the PC or system console via Peripheral Component Interface (PCI) bus and PCI interfaces with D_PMU and A_PMU via Serial Peripheral Interface (SPI) bus. The interface to TDC and DDS are designed using PCI_TDC and PCI_DDS interface blocks.

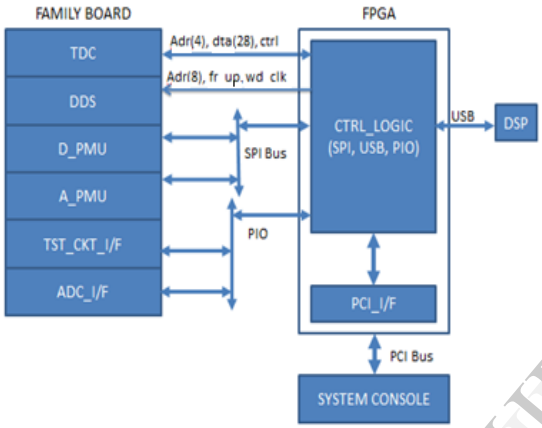

Figure 2. Functional configuration of controller

The FPGA controller consists of PCI core, SPI master controller, USB controller, PCI to DDS interface, PCI to TDC interface, PCI to ADC interface and parallel IO to control the proper switching of relays in the test circuit. System console sends data to the FPGA using PCI bus. PCI_SPI block is used to send this data to the SPI master controllers for PMUs. FPGA writes this data to A_PMU and D_PMU via SPI bus. FPGA writes to TDC or DDS or ADC or DSP using PCI_TDC, PCI_DDS, PCI_ADC, PCI_USB interface blocks respectively.

3.1. PCI Core

The Peripheral Component Interconnect (PCI) is a high-bandwidth, processor-independent bus that can function as a peripheral bus. PCI bus interface for 32 bit, 33 MHz is designed for a raw transfer rate of 133 MBytes/s. But it is not just a high speed that makes PCI attractive. Other features are:

- Processor independent
- Support for up to 256 PCI buses, each supporting 32 devices
- 64 bit extension possible
- Low power consumption
- Bursts can be performed in all read and write transfers
- Hidden bus arbitration
- Low pin count
- Transaction integrity check

A PCI bus transfer between the initiator and target may either be a read or a write. A bus transfer consists of one address phase followed by several number of data phases. For a PCI bus transaction to start, a valid address is placed in A/D line and FRAME signal is asserted by the initiator. Also in C/BE line the command telling what type of transfer is required is placed during the address phase (0010 for device read, 0011 for device write, 0110 for memory read, 0111 for memory write, 1010 for configuration read, 1011 for configuration write etc.). Next, data is transferred in bursts, which forms the data phases. During data phases C/BE lines show which bytes are enabled during the current transaction. For example, if C/BE(3) = 0 then only the last byte (8 bits) among the double word (32 bits) is valid so that only the valid byte is transferred. After the address and the command is transferred, the bus master uses the IRDY (Initiator Ready) signal to indicate readiness to receive or send data. The target of the transfer responds with DEVSEL (Device Select) to indicate that it has been addressed, and with TRDY (Target Ready) to indicate readiness to send or receive data. When both the initiator and the target are ready, one unit of data is transferred each clock cycle. If one of them is not ready, that data phase is delayed till both become ready.

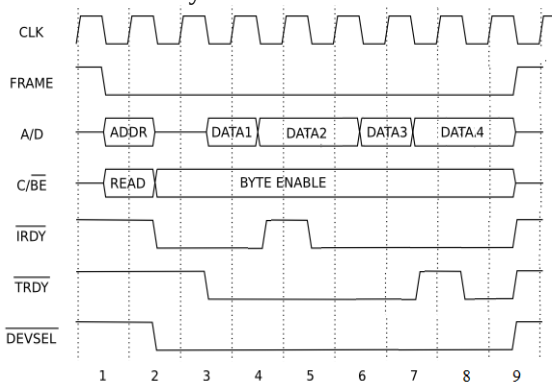

Figure 3. Timing diagram for PCI bus Read

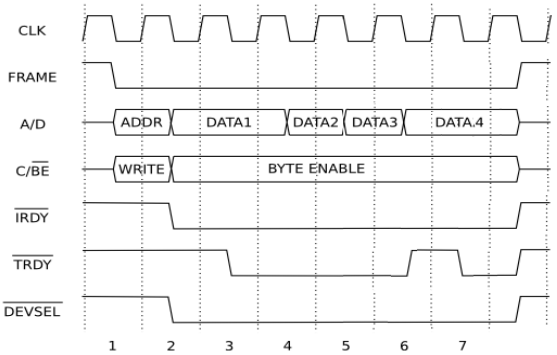

Figure 4. Timing diagram for PCI bus Write

Different devices request to use the PCI bus simultaneously. But one of them is permitted to use the bus at a time. For this, an arbiter is designed based on fairness rotation algorithm.

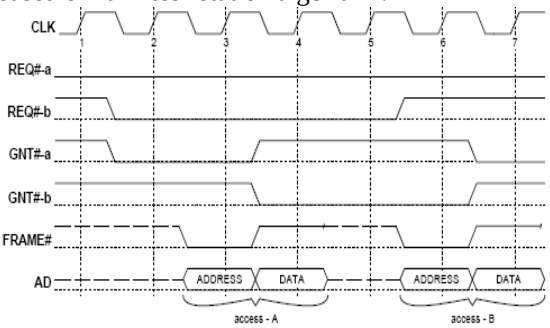

Figure 5. PCI bus arbitration for two devices

The bus arbitration for seven devices (TDC, DDS, APMU, DPMU, ADC, Test circuit interface, USB Controller) is designed here. When more than one device request to use the bus, the first transaction is done by highest priority device, the next transaction by next priority device.

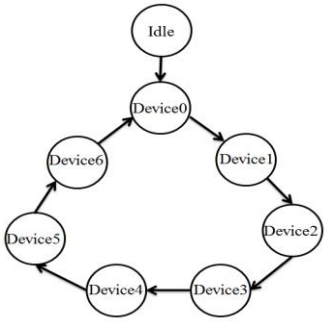

Figure 6. Rotation scheme used in PCI arbiter

The state diagram for PCI core is shown in Figure 7. A read cycle requires turn around state which is not required in a write cycle. Whenever term=1, stop signal is asserted low and it will go to idle state.

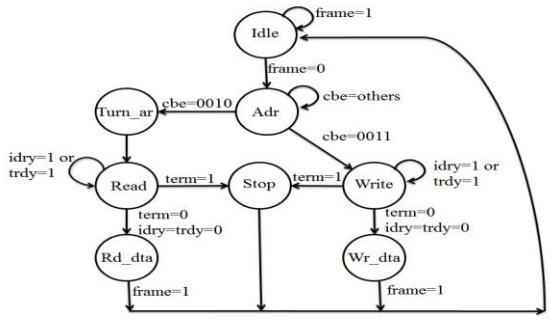

Figure 7. State diagram for PCI core

3.2. SPI Master Controller

One of the simplest serial bus protocols is the Serial Peripheral Interface bus (SPI) bus. It is a synchronous serial data transfer standard popularized by Motorola. Devices on the SPI bus transfer information in a full duplex mode and communicate in a master/slave configuration, where the master initiates the data transfer between itself and one or multiple slaves. The SPI bus uses four signals to affect the transfer of information between devices. They are Serial Clock (CLK), Serial Data Output (SDO), Serial Data Input (SDI) and Chip Select ($\overline{CS}$). Data is shifted out during write and data is being shifted in during read. Data is transferred on either the rising or falling edge of the CLK signal, depending on the settings of the Clock Polarity and Phase parameters.

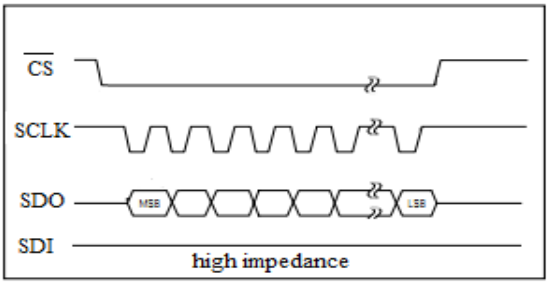

Figure 8. Timing diagram for SPI bus Write

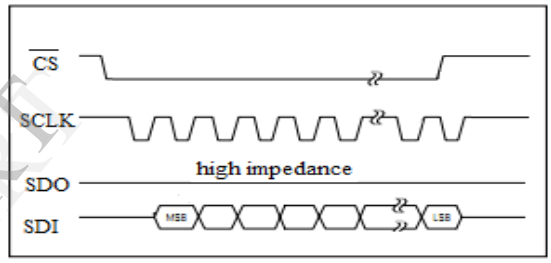

Figure 9. Timing diagram for SPI bus Read

Two SPI Master controllers are designed for APMU (AD5522) and DPMU (ADATE318) using a simple state machine as shown in Figure 10. When initiate=1, SPI controller block becomes ready for data transfer to or from the PMUs (Ready state). The input rd_wrbar determines whether it is a read or a write. For a write operation, the data input given to data_in is serially shifted to serial output line. For a read operation, the input is given to serial input line. It is shifted into a temporary register and finally when bitcount reaches 'n', it is given to data_out as output. Here n=29 for APMU and n=27 for DPMU.

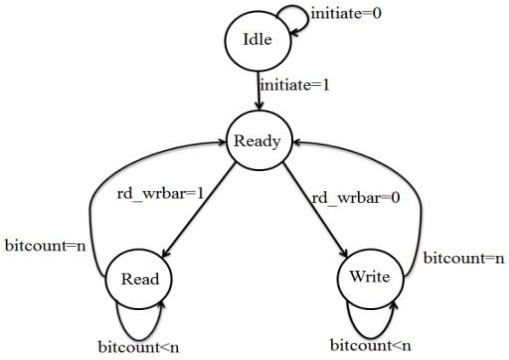

Figure 10. State diagram for SPI Controller

3.3. USB Controller

USB communication consists of a series of frames. Each frame contains one or more transactions. Each transaction is made up of a series of packets. A packet is enclosed within a SYNC pattern and an End of Packet (EOP) pattern. Each transaction consists of a token packet, data packet and handshake packet.

Here, USB Controller is designed to communicate between the PCI core and the Digital Signal Processor board Blackfin ADSP506 via USB port. The design is based on the USB protocol. The test system communicates with DSP processor which is considered as a FS (Full Speed) USB 1.1 device and it supports 12 Mbps. In USB data transactions are in packet format and each packet is enclosed within a SYNC pattern and a EOP (End of Packet). For a Full Speed device, SYNC pattern is K-J-K-J-K-J-K-K and EOP is SE0-SE0-J. Sequence detectors have been design for SYNC detection and EOP detection. A counter is used to count the bits in between SYNC and EOP. The count determines whether it is a token packet or data packet or handshake packet.

In USB the address/endpoint information in token packet is eleven bits wide, and is secured by CRC of 5 bits (generator polynomial is $G(X) = X^5+X^2+1$). If PID error or CRC error occurs for a token packet, then it is discarded and it waits for next token. If there are no such errors, that token is decoded by identifying the PID and determines whether it is a read (IN) or a write (OUT) transaction. In data packet, data is eight bits wide and is secured by CRC of 16 bits ($G(X) = X^{16}+X^{15}+X^2+1$). If there are no PID and CRC errors, that packet is successfully received/sent.

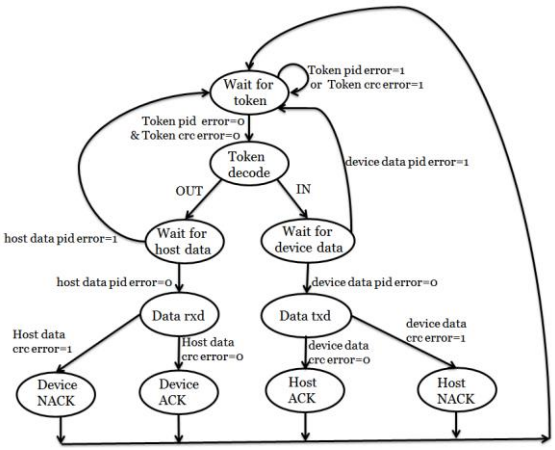

Figure 11. State diagram for USB Controller

3.4 Interfacing to TDC, DDS and ADC

Here TDC–GPX of acam-mess electronic is used to measure a time interval between two events and to convert to digital form. A valid address is placed in ADR pin after asserting CSN low. Then WRN is asserted low for a write and RDN is asserted for a read. Then valid data is sent to/ received from DATA pin. After completing the operation, WRN or RDN and CSN are asserted high.

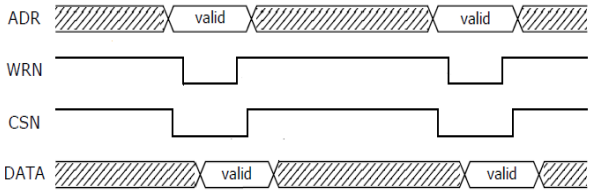

Figure 12. TDC Write operation

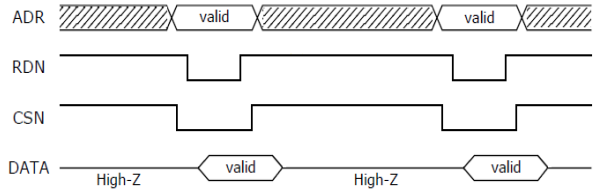

Figure 13. TDC Read operation

AD9851 is used as DDS. 40 dit data has to be written to its 40 bit internal register. Parallel load is used so that 40 bits are written in 5 cycles of system clock and each cycle transmitting a byte at the rising edge of word clock. After transmitting 8 words, frequency update signal is set high so that the internal register gets updated by this 40 bit value.

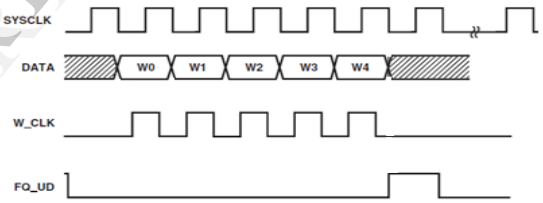

Figure 14. Parallel load phase/frequency update

ADC used here is AD7870/75/76 (12 bit). Conversion is initiated by a rising edge on CONVST pulse. INT is normally high and goes low at the end of conversion. A read operation from the ADC accesses the data from DATA line when CS and RD are set low. INT is again set high during the falling edges of CS and RD. CONVST input must be high when CS and RD are brought low for the ADC to operate correctly.

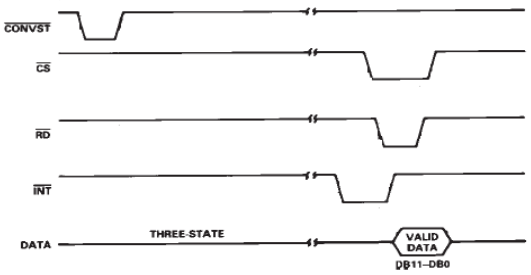

Figure 15. Parallel Read (12 bits) from ADC

4. Results and Discussions

The system console communicates with FPGA through PCI core. PCI core can read or write data

of different lengths as 1 byte or a word or a double word with the help of byte enable signal.

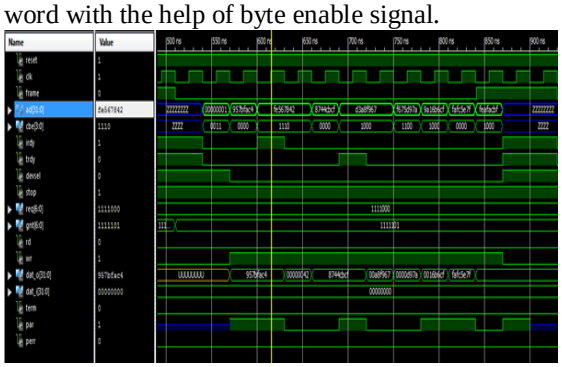

Figure 16. Simulated waveform for PCI Write

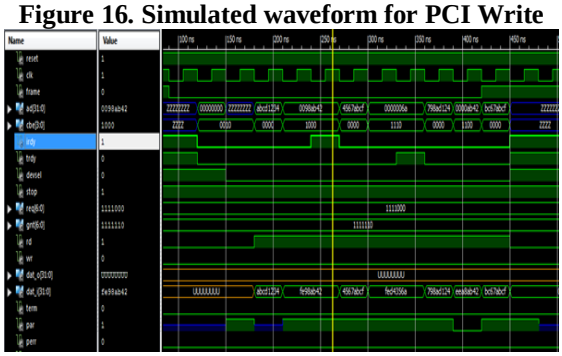

Figure 17. Simulated waveform for PCI Read

PCI arbiter accepts the requests from five PCI devices and permits only one device to access the bus at a time. Fairness rotation algorithm is followed here. From idle state, the arbiter gives the first priority to device 0 (Here TDC). Then the priority rotates.

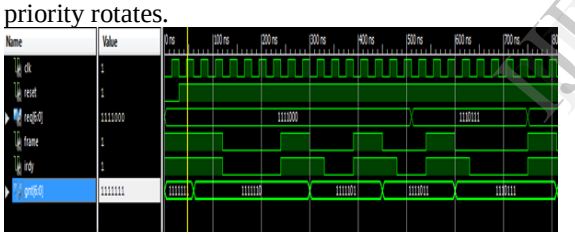

Figure 18. Simulated waveform for PCI arbiter

SPI master controllers are designed to accept data from PCI bus and to send it to A_PMU and D_PMU chips and also to send back the data from these blocks to the PCI bus. During SPI write transaction, data is sampled at the rising edge of clock and during read operation, data is sampled at the falling edge.

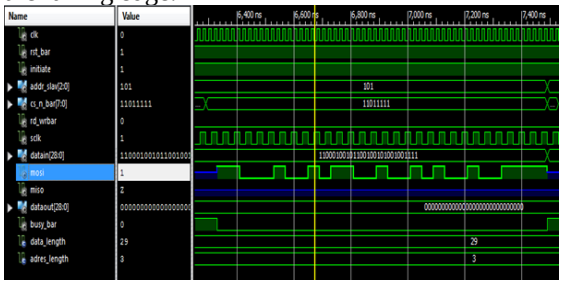

Figure 19. Simulated waveform for SPI Write

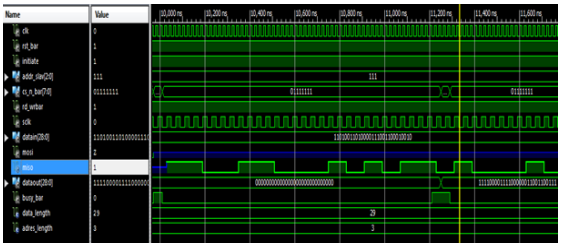

Figure 20. Simulated waveform for SPI Read

USB Controller is designed to interface PCI core with Digital Signal Processor board Blackfin ADSP506 via USB port. All the packets are sent in NRZI encoded form with bit stuffing.

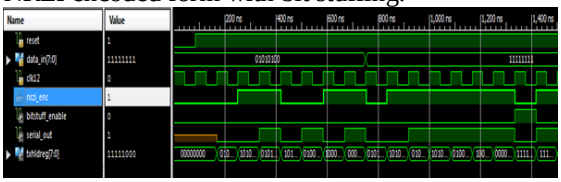

Figure 21. NRZI encoding and bit stuffing

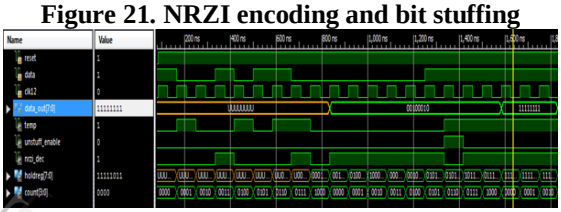

Figure 22. NRZI decoding and bit unstuffing

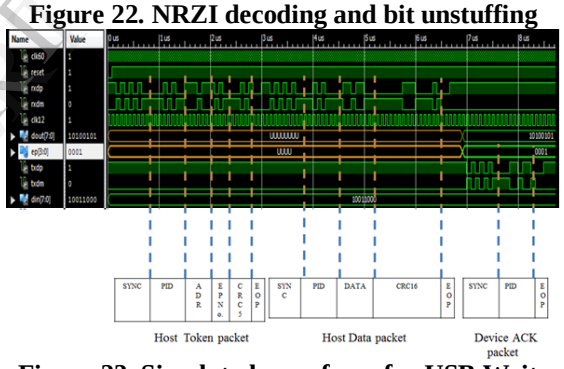

Figure 23. Simulated waveform for USB Write

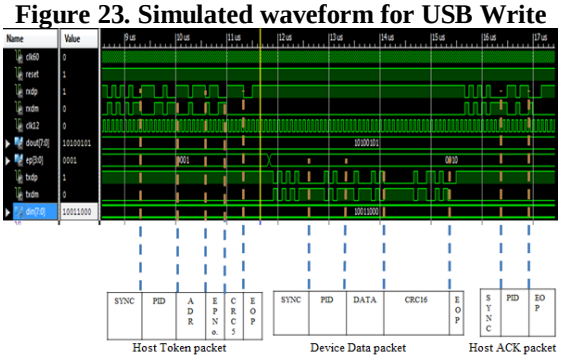

Figure 24. Simulated waveform for USB Read

DDS has a data width of 40 bit. So 2 cycles of PCI transaction are needed. These 40 bits are written to DDS as parallel load of 5 words, each consisting of 8 bits. Data is sampled at the rising edge of clock. After transmission, frequency update signal is set high for 1 clock cycle time.

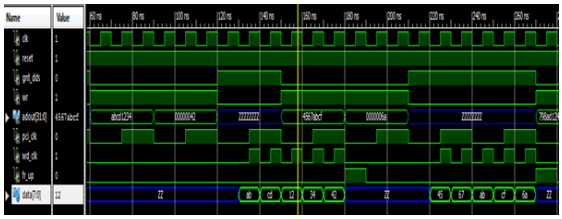

Figure 25. Simulated waveform for DDS Write

For starting conversion, 1 is sent via PCI bus (dat_i[0]=1) after asserting wr high. This creates a rising edge of adc_convst and the conversion begins. When adc_int is set low, the conversion ends. When adc_cs and adc_rd are made low, the data will be available in adc_data pin.

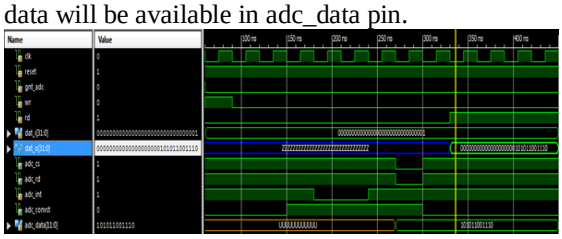

Figure 26. Simulated waveform for ADC Read

TDC has a 28 bit width data bus and a 4 bit width address bus. Read and write operations of TDC are shown below.

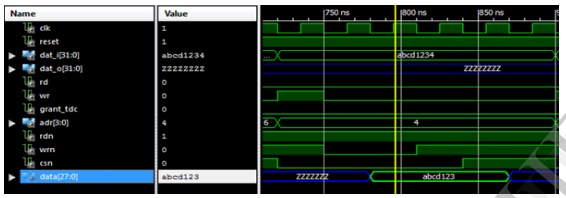

Figure 26. Simulated waveform for TDC Write

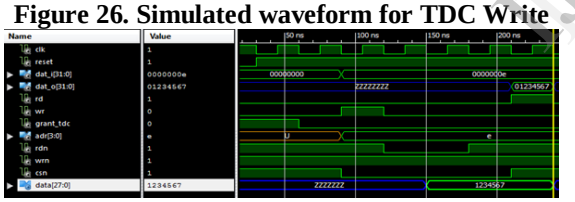

Figure 27. Simulated waveform for TDC Read

32 bit parallel data is given to the test circuit control bits when the grant signal for test circuit interface is enabled.

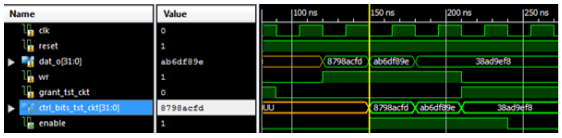

Figure 28. Test circuit interface

5. Conclusion

In this paper the design and development of a FPGA controller which is meant for controlling the various units in the linear and mixed signal test system is considered. Its functions are accepting the test commands from PC through PCI, then controlling and sequencing the various functional units of the test system like DDS, TDC, DPMU, APMU, ADC etc. via SPI bus, and then getting back the response from DUT and sends back the test parameter to PC through PCI bus for data analysis. A USB Controller is also developed for further expansion of the test system where this FPGA Controller is used. The work is designed using VHDL language and then the implementation part is done using Xilinx FPGA.

6. References

[1] Yanping Li, Jia Lei, "Design of Analog test Hardware platform based on IEEE 1149.4", *The Ninth IEEE International Conference on Electronic Measurement and Instrumentation*, 2009, pp. 873-876.

[2] Michael G Dimopoulos, Dimitris K Papakostas, Alkis A Hatzopoulos, "Design and Development of a Versatile Testing System for analog and mixed signal circuits", *6th IEEE International Conference on Electronic Circuits and Systems*, 2007, pp.846-849.

[3] Ronal H. Troy, Charles .F, "Fundamentals of Testability- A Tutorial", *IEEE Transactions On Industrial Electronics*, vol.36, no.2, May 1989.

[4] Venkatanarayanan H V, Bushnell M L, "An area efficient mixed signal test architecture for system on chip", *IEEE Proc. of 19th International Conference on VLSI Design (VLSID'06)*, Jan 2006.

[5] R Mehtani, B Atzema, M De Jonghe, R Morren, G Seuren, T Zwemstra, "*Mix Test*: A Mixed-Signal Extension to a Digital Test System", *International Test Conference, 1993*, paper 42.4, pp.945 - 953.

[6] J. A. Starzyk, Z. H. Liu, and J. Zou, "An Organization of the Test Bus for Analog and Mixed-Signal Systems", *IEEE 1994*, pp. 247 - 251.

[7] Haruyasu Hayasaka, Hiroaki Haramiishi, Naohiko Shimizu, "The Design of PCI Bus Interface", *IEEE 2003*, pp. 579 - 580.

[8] "Designing PCI bus interfaces with programmable logic",*Proc. of the 8th Annual IEEE International ASIC Conference*, September,1995, pp. 321 – 324.

[9] Brueggen. C, Pottinger H.J, "A novel FPGA-based master/slave PCI interface core", *Proc. of the 43rd IEEE Midwest Symposium on Circuits and Systems*, vol.1, 2000, pp.162 – 163.

[10] Guoyong Zhen, Yanhu Shan, "The Design and Realization of Data Asynchronous Exchange Protocol Based on USB and FPGA", *Third IEEE International Conference on Information and Computing (ICIC)*, vol. 1, 2010, pp. 196 – 198.

[11] Leens F, "An Introduction to I2C and SPI protocols", *Instrumentation and Measurements Magazine*, vol.12, Feb., 2009, pp. 8-13.