"USB 3.0 Based Digital Data Pattern Generator and Data Formatter"

Harsidhdhi S. Kotila^{1*}, Vishal S.Vora²

¹Department of Electronics & Communication Engineering, Atmiya Institute of Technology & science, Rajkot, Gujarat, INDIA

Abstract - The project is concentrating on development of digital pattern generator and formatter for the high speed systems. As the technology advances the requirement of fast data transfer and processing arises. To meet the requirement the data should be read, write or processed in faster rates typically in Gbps. The proposed system is an effort to the same field and develops the high speed USB data pattern generator and formatter working in the Gbps speeds. USB is an interface that connects a device to a computer. With this connection, the computer sends or retrieves data from the device. USB gives developers a standard interface to use in many different types of applications. A USB device is easy to connect and use because of a systematic design process. Presently there are many systems available with USB 2.0 interface at 480 Mbps. The USB 3.0 specification, released in 2008, allows a maximum signaling rate of up to 5 Gbps (SuperSpeed), which is 10 times the signaling rate of High Speed. USB 3.0 based Digital Data Formatter card which acquires data from multiple data chains and feeds it to USB 3.0 module and it sends data to host PC over USB 3.0 cable, for storing the acquired data in the system memory of host PC for further evaluation.

Keywords: - USB 2.0, USB 3.0 CONTROLLER, SIMULATOR, FORMATTER, PATTERN GENRATOR, FPGA, PCIe, etc..

I. INTRODUCTION

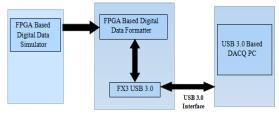
Aim of this project is development of digital data pattern generator and data formattar Using USB protocol to communicate between the nodes for SAC ground checkout systems. Pattern generator is a device, which can be programmed to generate any particular digital signal for testing purposes. It usually consists of several channels. These channels serve as the output of the generator. Each channel can individually be programmed for a desired signal. The programmed signal or the pattern is stored in a buffer memory and clocked out at a user defined sampling rate. The application for a pattern generator varies from generating a single channel square wave to digital protocols such as SPI, USB, and multi channel complex patterns. Digital Signal formatting is the process of transforming information from one format into another. This is often used in many digital devices and for communication processes. A digital system is a data technology that uses discrete (discontinuous) values. By contrast, non- digital (or analog) systems use a

continuous range of values to represent information. Although digital representations are discrete, the information represented can be either discrete, such as numbers, letters or icons, or continuous, such as sounds, images, and other measurements of continuous systems. In our proposed system, USB interface is used.

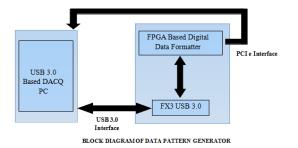
Motivation for USB $2.0^{[2]}$ was stem from the fact that PCs have increasingly higher performance and are capable of processing vast amounts of data. At the same time, PC peripherals have added more performance and functionality. User applications such as digital imaging demand a high performance connection between the PC and these increasingly sophisticated peripherals. The same demand has increased and motivated the development of FX3 to meet higher data rate demand up to 5 Gbps. For the testing of high speed devices the digital data pattern generator and formatter is used. The proposed solution meets the high speed data rate requirement for digital communication. demand up to 5 Gbps. For the testing of high speed devices the digital data pattern generator and formatter is used. The proposed solution meets the high speed data rate requirement for digital communication.

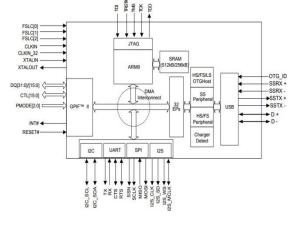
2. USB INTERFACE DESCRIPTION

USB 3.0 and USB 2.0 devices connect to and communication with the USB 3.0 host. The USB 3.0 interconnect inherits core architectural elements from USB 2.0, although several are augmented to accommodate the dual bus architecture. The baseline structural topology is the same as USB 2.0. It consists of a tiered star topology with a single host at tier 1 and hubs at lower tiers to provide bus connectivity to device.



BLOCK DIAGRAM OF DATA FORMATTER





3. LOGICAL STRUCTURE OF THE INTERFACE A. FPGA/ASIC Interfaced to FX3

Several applications connect an FPGA or ASIC to FX3 over a high-speed parallel GPIF II interface. Another device, which is the source or sink of data is connected to the FPGA or ASIC. Examples of such applications are data acquisition devices, printers, scanners, and imaging devices. In such applications, FX3 serves as a fast data pipe to and from a USB host. For these applications, the FX3-to FPGA/ASIC interface is typically the synchronous Slave FIFO interface.

In applications where FX3 connects to an FPGA, it may be preferred to load the FPGA bit file through FX3. In this case, the FPGA configuration file can be transferred from a USB host through FX3 to the FPGA. This saves the cost of an FPGA boot memory and enables FPGA reconfiguration (including updates) from the PC.

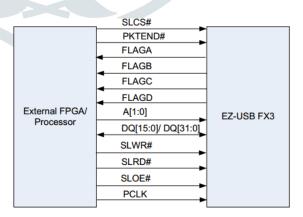


Fig 4. FPGA/ASIC Interfaced to FX3

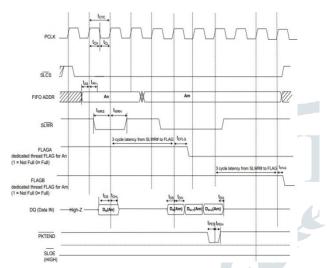
Fig.3 Block diagram of FX3

FX3 is a USB 3.0 peripheral controller with an integrated ARM9 processor. Parallel and serial interfaces provide high-speed connectivity with other devices in the user system. The main function of the FX3 device in a system is to transfer high- bandwidth data between a USB host and a peripheral device, such as a camera or scanner. The presence of a powerful ARM9 processor on-chip also allows FX3 to access the data stream and efficiently process data. In systems where FX3 is not required to perform data processing, the ARM9 firmware only initializes and manages data transfers between two interfaces—USB and a data consuming/providing device.

FX3 has a highly flexible, programmable interface known as the General Programmable Interface Generation 2 (GPIF II) in addition to I2C, SPI, UART, and I2S serial interfaces. GPIF II programmability allows FX3 to be connected to various types of devices including FPGAs, image sensors, ADCs, and application processors. This makes FX3 a good controller choice in a wide range of high-performance USB 3.0 applications.

B. Synchronous Slave FIFO Write Sequence

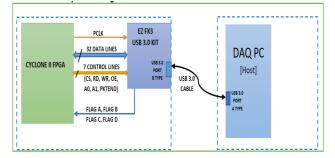
The sequence for performing writes to the synchronous Slave FIFO interface is - 1.FIFO address is stable and the signal SLCS is asserted. 2. External master/peripheral outputs the data onto the data bus. 3.SLWR is asserted. 4.While the SLWR is asserted, data is written to the FIFO; on the rising edge of the PCLK, the FIFO pointer is incremented. 5.The FIFO flag is updated after a delay of Tc flag from the rising edge of the clock.



C. Synchronous Slave FIFO Read Sequence

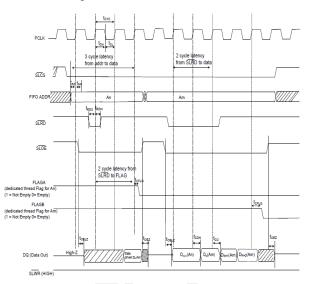
The sequence for performing reads from the synchronous Slave FIFO interface is:1. FIFO address is stable and SLCS is asserted.2.SLOE is asserted. SLOE is an output enable only whose sole function is to drive the data bus. 3.SLRD is asserted.

The FIFO pointer is updated on the rising edge of the PCLK while SLRD is asserted. This action starts the propagation of data from the newly addressed FIFO to the data bus. After a propagation delay of Tco (measured from the rising edge of PCLK), the new data value is present. N is the first data value read from the FIFO. To drive the data bus, SLOE must also be asserted.



D. Test Setup

Block diagram of test setup implementation is shown. Here 32 bit data bus and all control signals of EZ USB FX3 module are connected to FPGA. This FPGA is used to generate digital data pattern and control signal to provide FX3 module. Quartus II software and VHSIC Hardware



Description Language (VHDL) is used for programming the FPGA.FX3 USB module is a FIFO based device, FIFO write operation is done to communicate with this module. FX3 module is connected to host PC through USB 3.0 interface. Stored data in FIFO is transfer through USB 3.0 interface when host is ready to receive data. This system can be utilized as high speed communication system working on Gbps speed.

4. RESULT AND DISCUSSION

Output Testing in two different types of application.

A. Vendor Application – USB Control Center and C++ Streamer

Using USB control centre, different data pattern can be given to the FX3 card [refer figure 7]. The USB control centre can be divided into two modules, i.e. USB control centre and C++ streamer. To test the input sequence to FX3 card, USB control centre is used and for testing the output bit rate of read and write sequence the USB streamer is used which can simulate output speed of FX3 interface for given test sequence.

		🚱 USB3 Data Acquisition Application
😙 USB Control Center		File Compare View Plot Settings
File Program Help		Sample AcQ: No Of Lines: 10000 🕏 No Of Iterations: 50 🕏 Acq Mode: UNPACK 💌 M/R 💌 Acquire
🖛 🗉 🖸 🖂 🕅 🍰 🕨	URB Stat Abort Pipe Reset Pipe 🗶 🌡 🥲 🌮	Real Time AcQ: Data Recording
	Descriptor Info Data Transfers Device Class Selection	MSG PLOT DATA DUMP
BOS Gonfiguration 1	Transfer parameters	=> 1.Recording Status: 658 chunks written out of 667 chunks => 1.Recording Status: 659 chunks written out of 667 chunks
- Control endpoint (0x00)	Text to send: Data to send (Hex):	=> UnpackLIB->Counter values:- 0, -19 => 1.Recording Status: 660 chunks written out of 667 chunks
- Interface 0		I => UnpackLIB->Time taken in unpacking 3 => 1.Recording Status: 661 chunks written out of 667 chunks
Alternate Setting 0 - Bulk out endpoint (0x01)	Bytes to transfer.	=> 1.Recording Status: 662 chunks written out of 667 chunks => 1.Recording Status: 663 chunks written out of 667 chunks
Bulk in endpoint (0x81)	1048576 PktMode	=> 1.Recording Status: 664 chunks written out of 667 chunks
		=> 1.Recording Status: 665 chunks written out of 667 chunks => 1.Recording Status: 666 chunks written out of 667 chunks
		=> 1.Recording Status: 667 chunks written out of 667 chunks => Stopping the acquisition process, acquired required no. of chunks
		dataRecordingComplete Current software state is: ACQ COMPLETE SUBSTATE
		=> UnpackLIB->Stop Acquisition commands sent to all day threads => UnpackLIB->Stop Acquisition commands sent to all unpacking threads
	Transfer Data-IN Transfer File-IN Clear Box	=> USB FX3 LIB->Dev1->Finish DataXfer :2090 2070 2090
		Current software state is: NO_ACQ_SUBSTATE DataAcquisitionComplete
	FFF20 CB AF FF FF CC AF FF FF CD AF FF FF CE AF FF FF FFF30 CF AF FF FF D0 AF FF FF D1 AF FF FF D2 AF FF FF	Current software state is: WAIT_STATE => UnpackLIB->Dag threads have exited
	FFF40 D3 AF FF FF D4 AF FF FF D5 AF FF FF D6 AF FF FF	S => UnpackLIB->Retreival thread has exited => UnpackLIB->Unpacking threads have exited =
	FFF50 D7 AF FF FF D8 AF FF FF D9 AF FF FF DA AF FF FF FFF60 DB AF FF FF DC AF FF FF DD AF FF FF DE AF FF FF	
	FFF70 DF AF FF FF E0 AF FF FF E1 AF FF FF E2 AF FF FF	
	FFF80 E3 AF FF FF E4 AF FF FF E5 AF FF FF E6 AF FF FF	
C++ Streamer		Comparision Successful for Iteration:40 Recorded Filesize:
Connected Devices	(0x04B4 - 0x00F1) Cypress FX3 USB StreamerExample	
Endpoint	BULK IN, 16384 Bytes, 15 MaxBurst, (0 - 0x81)	B ISRO DEVELOP APPLICATION- USB3 DATA
Packets per Xfer	16 v Successes 10368	ACQUISITION APPLICATION
Xfers to Queue	64 Failures 0	
Timeout Per Xfer (ms)	1500 Stop	
Transfer Data (KD/a)		
Transfer Rate (KB/s)		
	39000	
Show Transfered Dat	8	
		USB3 Data Acquisition Application
		File Compare View Plot Settings
		Sample AcQ: No Of Lines: 1 🚔 No Of Iterations: 1 🚔 Acq Mode : UNPACK 🕥 M/R M 🕤 🛛 Acquire
		Real Time AcQ : 💽 Line Plot 🝷 🗆 🗆 Data Recording
		MSG PLOT DATA DUMP
		0 00001000 00001001 00001002 00003003 00001004 00001005
		6 00003006 00001007 00001008 00001009 0000300a 0000300b
	Fig.7 USB Control Center	12 0000100c 0000300d 0000100e 0000100f 00001010 00003011 18 00001012 00003013 00001014 00003015 00003016 00001017
	C	18 00001012 00003013 00001014 00003015 00003016 00001017 24 00001018 00003019 0000301a 0000301b 0000301c 0000101d
		30 0000101e 0000101f 00003020 00001021 00001022 00003023
		36 00003024 00001025 00003026 00001027 00003028 00001029
		42 0000302a 0000302b 0000302c 0000302d 0000102e 0000102f
		48 00003030 00003031 00001032 00003033 00003034 00003035 54 00003036 00001037 00003038 00003039 0000303a 0000303b
		60 0000103c 0000103d 0000103e 00003039 0000303a 0000303B
		66 00001042 00003043 00001044 00001045 00003046 00001047
		72 00001048 00001049 0000304a 0000304b 0000104c 0000304d
		78 0000104e 0000104f 00001050 00003051 00001052 00003053
		84 00001054 00003055 00003056 00001057 00001058 00003059 90 0000305a 0000305b 0000305c 0000105d 0000105e 0000105f
		Line No : 1 🖶 Start Addr : 0 🖶 HOME END PREV NEXT
		Find What(in Hex): Find Next Reset Search
		Offline Viewing : C:/Users/USB/Desktop/usb_acq_vnir_M.ref Recorded Filesize:

Fig.9 USB 3 Data Acquisition Application

There are well-establish solutions for connection with desktop processors, like multi-gigabit optical fibre or external PCI-Express, the same is not true for portable detection application that are often used connected to laptops through the USB 2.0 bus or the Ethernet bus. However, the bandwidth of the USB 2.0 bus is too small for very high data rates, both in case of fast sampling and in presence of a high number of channel. So, USB 3.0 Protocol is used.

6. CONCLUSION

In the proposed system, Digital data Formatter is generated for testing the bit rate of USB 3.0. The system of FX3 card is done on FPGA simulation card USB FX3 card is acting as data formatter card and FPGA is acting as simulation card for the proposed system. With this concern, higher bit rate has been achieved. The USB 3.0 is having maximum bandwidth of 5 Gbps. So, future scope is to achieve the highest bit rate near to 5 Gbps with the same system. This System can be used in high speed data interfacing in modern system, imaging applications in satellites, data storage and transfer etc.

7. REFERENCES

- [1] EZ-USB® FX3: SuperSpeed USB Controller www.cypress.com/products/ezusb-fx3-superspeed- usb- 30peripheral-controller
- [2] Universal serial bus specification revision 2.0

www.poweredusb.org/pdf/usb20.pdf

[3] Available on:

http://www.cypress.com/?id=3526

- [4] Universal serial bus specification
- revision 1.0, 1998

[5] Implementation of the USB Interface in the Instrumentation for Sound and Vibration Measurements and Analysis IEEE international Workshop on intelligent Data Acquisition and Advanced Computing System: Technology and Applications S- 10SeptemberZW3,LvivU,L June.

[6] S. Gandhi, "Designing with the EZ-USB FX3 Slave FIFO Interface," Cypress, 2013. Implementation of a High-Rate USB Data Acquisition System for PET and SPECT Imaging 2006 IEEE Nuclear Science Symposium Conference Record