Multiple Channel Serial I/O Interfacing Using Fpga Kit
Krupa Patel1 Sneha Shah2
1P.G Student, Department of Electronics & Communication
2Assistant Professor, Department of Electronics & Communication
L.J. Institute of Engineering and Technology Ahmedabad, Gujarat, India.

Abstract— The Digital input and output interface could be a
important part for a machine needed to be interacted with its
setting. Even though a main processing unit (MPU) utilized
within the machine has its own Digital input output channel,
for some cases, the machine should use plenty of input
output channel than those the MPU can offer. To avoid this
downside, a process board interfaced on to the MPU via
serial communication is intended. If the MPU connect with
multiple process boards directly, it’s going to waste
sometime to process of communication data. The inessential
measure of the MPU and likelihood to have many boards
connected, a serial interface unit supported FPGA is
developed to help the MPU to talk with process boards. The
FPGA unit automatically browses serial signals from each
process boards and save understood data to share registers
for MPU to scan. The FPGA unit put together impromptu
reads registers written by MPU and send registers values to
process boards automatically.

Keywords: - Process, MPU, Interface, FPGA

I. INTRODUCTION

In general, a machine has a minimum of one digital input or
output unit to move with alternative machines or its
atmosphere. Most the time, MPU features a limit range of
input and output channel, that is often but the machine
wants. A method to avoid this downside is to possess a
process board connecting to the MPU via serial
communication. Rather than victimization thirty two pins
for thirty two input channels, solely four pins for a serial
communication for required for MPU to interface with
process board.

A communication exploitation serial protocol is
time overwhelming. The communication between MPU and
process boards in signal hand checking before receiving or
transmitting information generally consumes plenty of our
time from the MPU. To avoid these issues, a middleware
unit has been enforced on FPGA to handle communication
tasks between the MPU and process boards. Whereas the
FPGA unit communicates with process boards via serial
interfaces, share registers area unit want to link between the
FPGA unit and MPU.

FPGA is wide used to implement a middleware
unit. It’s enforced to be a SPI master for a microcontroller
unit (MCU) to speak with SPI Slave. By exploitation SPI
protocol, it links between DSP and wireless process. It’s
want to browse four channels ADC using four completely
different protocols that are parallel, SPI, and one-wire
protocol, and communicates with computer by USB
controller. Flexray controller on FPGA for intravehicular
communication is enforced.

In this paper, I introduce a development of FPGA
unit that automatically write and browse information from
multiple process boards via serial communications and also
the MPU via share registers. Having the FPGA unit, it helps
to cut back the burden of a serial communications between
the MPU and process boards and additionally increase
variety of process boards that the MPU will connect with. I
selected to implement the FPGA unit and program it by
VHDL language.

In this article, I will explain protocol to
communicate with process board and will introduce
implementation of automatic multi channel serial I/O
interface on FPGA.

II. INTERFACING

The interfacing requirements for a serial I/O peripheral are
the same as for a parallel I/O device. The microprocessor
identifies the peripheral through a port address and enables
it using the read or write control signals. The primary
difference between parallel I/O and serial I/O is in the
number of lines used for data transfer. The parallel I/O uses
the entire data bus and the serial I/O uses one data line. Fig.
1 shows a typical configuration of serial I/O transmission.
The MPU chooses the peripheral through chip select and
uses the management signals scan to receive information
and write to transmit information. The address secret writing
is either peripheral I/O or memory mapped I/O. Similarly, a
serial peripheral can be interfaced under either program
control or interrupt control. Serial communication uses
Synchronous Data Transmission as well as asynchronous
Data Transmission. In Synchronous Data Transmission, the
transmitter and receiver are synchronized whereas
asynchronous Data Transmission occurs at any time.

Fig. 1: Block diagram of Serial Input Output Interfacing

III. FLOWCHART AND ERROR CHECKING

Various types of errors may occur during transmission. To
allow checking for these errors, extra data is transmitted
with the info. Error checking techniques are parity checking
and verification. But in this article checksum (CS) method is
used when larger blocks of data are being transmitted
combined flowchart for error checking and in transmission
and Reception of serial data is shown in figure 2 and 3.
Multiple Channel Serial I/O Interfacing Using FPGA Kit

Fig. 2: Flowchart for whole process

IV. INCREASING INPUT-OUTPUT CHANNEL BY USING SERIAL INTERFACE

In industrial machine, most of MPU pins are used to communicate with several peripheral elements. Therefore, there’s not abundant MPU pin left for different use. To manage machine with several input-output (I/O) channels, the quantity of I/O to regulate I/O device is required to be sufficiently enhanced. Rather than directly connecting between MPU pins and I/O pins, process boards can communicate with MPU via FPGA by using serial protocol as showed in Fig. 4.

Signals:
1. X: Input
2. tx_decoded
3. rx_decoded
4. Clk
5. Addr

Fig. 3: Experimental Setup

Fig. 4: Detailed Block Diagram

In Fig. 4 the Dir-signal defines a direction of a communication signal. Dir is 1 as FPGA send data to process board. Dir is 0 as FPGA receive data from process board. CLK-signal defines a synchronize of communication and works at rising edge. Whilst Dout-signal is used to send data from FPGA to process boards, Din-signal is used to receive data from process boards to FPGA. Addr-signals represent an address of process board that the FPGA unit wants to communicate with.

Input and signaling diagrams are shown in Fig 5. To send out information (Data Out), a most significant bit (MSB) is first sent. 0-bit to 63-bit are sent from the FPGA unit to process board whereas 64-bit to 67-bit area unit checking bits (Check Sum-CS) for data correction. To get information (Data In), MSB is received before a low significant bit (LSB). 0-bit to 7-bit are Check sum, whereas 8-bit to 71-bit are information sent from process board to the FPGA unit. Addr-signals shows sequence of communication in different process boards.

Fig. 5: I/O signal Diagram

V. EXPERIMENTAL RESULTS

We use Cyclone EP12Q240C8 to develop the FPGA unit in our experiment. The FPGA unit is connected between a MPU and one processes. The frequency CLK is set 150MHz for the serial protocol. It takes 7.612 nanoseconds to receive and send data to a single process. The experimental result
VI. CONCLUSION

We have developed the FPGA unit for linking between the MPU and process boards. It communicates with process boards via the serial protocol. Whereas share registers is employed to speak with the MPU the FPGA unit mechanically scans serial signals from every process boards and save taken information to share registers for MPU to read. The FPGA unit additionally spontaneously reads registers written by MPU and send registers values to process board mechanically. Time interval for handling the serial protocol within the MPU is reduced and additionally the MPU will hook up with multiple process boards at a similar time.

REFERENCES

"Design and Implementation of the Configuration Circuit for FDP FPGA” 978-1-4244-2342-2/08/$25.00
©2008 IEEE.


Spartan-3E Family”.

[14] David P. Schultz; Steven P. Young; Lawrence C. Hung.
“Method and Architecture for reading, modifying and writing selected configuration memory cells of FPGA”,

[15] technique for online transient error recovery of FPGA configuration”,in Proc FPGA 2001, February, 11-13,
Monterey, CA.,USA.


“Method and Structure of Configuring FPGAs”,


[19] Markh Enderson, I Lin Icklessr, I Cks Tevens. “ A Scalable High-Performance 1/0 System” 0-8186-
56802394 $03.00 © 1994 IEEE.