Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Vol. 4, Issue 2, May 2020

Design of Functionalities of a Programmable High Speed I/O

Aruna Rao B P
Asst. Prof, Department of ECE, K.S Institute of Technology, Bangalore (affiliated to VTU), India,
arunaraobagre@gmail.com

Dr. Shanthi Prasad M J
Professor, Department of ECE, Cambridge Institute of Technology, North Campus, Bengaluru, India,
prasadmjs1952@gmail.com

Abstract: In this paper we have made an attempt to discover different functionalities of a programmable IO like data conversions (parallel, serial),a buffer and an arbiter depending upon the situation the IO designed should be able to act on the data accordingly and support the speed of data movement.

Keywords: Parallel Transmission; Serial Transmission; FIFO (First in first out buffer); Arbiter, input/output

I. INTRODUCTION

With respect to the present technology the i/o which are present can be programmable and the modules coded and verified in this paper are a part are sub modules. Most of the above indicated sub modules of the proposed High speed programmable I/O are designed, simulated & implemented to verify the functionality. Later these sub modules will be integrated at the top level to have a multi-functional & multi-dimensional Next generation High speed I/O.

Basic Input/output System functions are:

- Communicate between chip and external world
- Drive large capacitance off chip
- Operate at compatible voltage levels
- Provide adequate bandwidth
- Protect chip against electrostatic discharge
- Use small number of pins (low cost)

A. The serious I/O Challenges

There are 2 major challenges to continued scaling of high-speed I/Os: band-limited channels and timing uncertainty.

- *Challenge 1*: As data rate increases, channel bandwidth becomes limited by the frequency-dependent loss of the channel. Therefore, handling large data or highspeed data in limited band width is a challenge
- *Challenge 2*: As signal rates scale, the timing jitter of a high speed I/O must decrease to remain a constant fraction of a bit-time or unit interval. It is important to know the value of jitter so that loss of data is minimum

B. Key Research challenges

- I/O Specifications are normally Analog, but we have to implement & achieve Digital Functionality
- Hot swap of logic levels – implementing programmable pull up & pull downs
- Implementing & achieving GPIO & SPIO is very complex process
- Implementing high speed Gigabyte PIN TRANSCEIVER is very difficult

II. METHODS

A. Serial to parallel converter

a) Parallel Transmission*:*

Parallel transmission uses one line for communication for each bit of the message. The accompanying figure depicts the transmission for an eight-bit message using a parallel an eight-bit bus, as shown in Fig. 1.

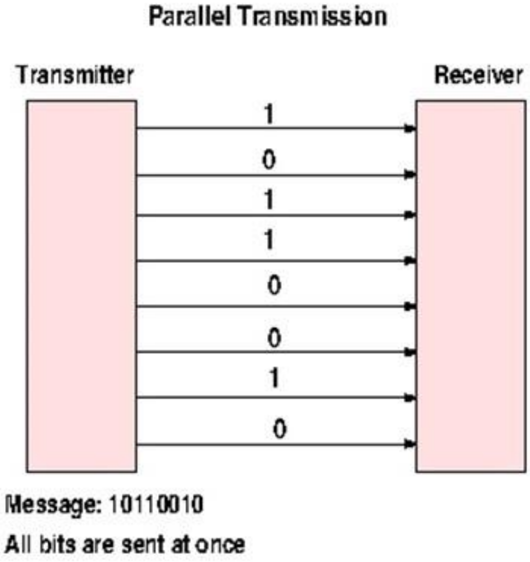


Fig 1. Parallel Transmission

b) Serial to parallel converter

To convert serial data to parallel data a set of flip-flops are required. The number of flip-flops is exactly the size of the serial data to be transmitted. For example, to transmit four-bit serial stream four flip-flops a required. A schematic of a four-bit converter is depicted.

The serial data is delivered at the input of the first flip-flop, and bits are successfully transferred to the next flip-flop on the rising (or falling) edge of the clock. The next figure shows an actual circuit for a four-bit converter, where four bits (0, 0, 0, and 1) are stored at the input of the first flip-flop.

With the first rising edge (i.e. tick) of the clock, the first bit (1 in this case) is transferred to the input of the second flip-flop. Successive ticks move the bits to the next flip-flop, until all four bits are stored at the output of each flip-flop. Once all the outputs are filled, the converter releases all the bits at once. For this process to happen, the converter is disabled (by means of one or more control lines) during the transfer process and enabled once all the bits are at the output bus. This is summarized by stating that the conversion is carried out in three stages:

1. Disable the output bus. The converter cannot send output data.
2. Load all the bits into the outputs of the flip-flops by moving them one bit at a time using the clock.
3. Once all the bits are loaded (all the flip-flops have one bit stored in the Q pin), then enable the bus operation. The bits are sent at once.

The working of the same can be witnessed in Fig. 2.

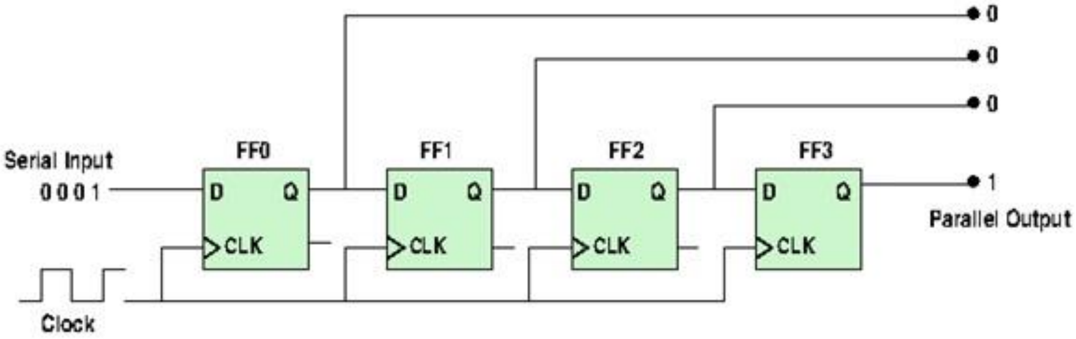


Fig 2. Serial to Parallel Conversion of Data

B. Parallel to Serial Converter

a) Serial Transmission

Each bit of the message is sent to the receiver one bit at a time using one communication line. Serial Data available on the I/O pins to can be used parallelly for many applications, as shown in Fig. 3.

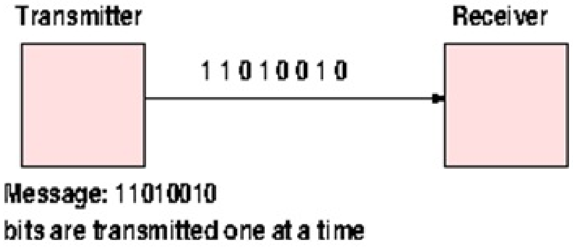


Fig 3. Serial Transmission

b) Parallel to Serial Converter

Parallel Data available on the I/O pins to can be used serially for many applications. Once this is achieved, with the help of the clock, data is shifted one-bit a time from the last flip-flop. This two-step process is schematically illustrated in the Fig. 4.

The multiplexer will force the parallel data to be shifted one bit at a time through the last (most significant bit) flip-flop. The following figure is the diagram of a four-bit converter. There are four flip-flops and three multiplexers. Each flip-flop is the output of a multiplexer, with the exception of the first flip-flop, which will represent the least significant bit (LSB) of the output serial data. Each multiplexer has two inputs and one output. The inputs are one bit of the parallel data and one input from the previous flip-flop.

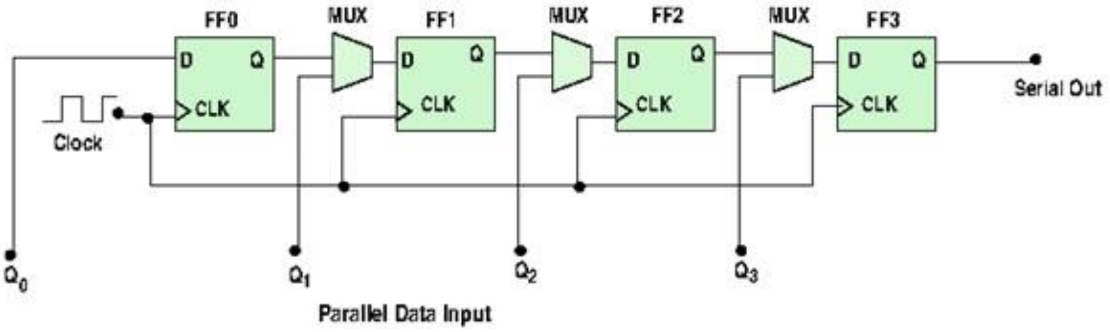


Fig 4. Parallel to Serial Converter

C. FIFO Buffer

This module is a First-in-First-Out (FIFO) Buffer Module commonly used to buffer variable-rate data transfers or to hold/buffer data used in digital communication and signal processing algorithms. The FIFO also has flags for empty, full and error. There is an output port for reading out the data count. As the port name suggest, this tells the world (outside the module) how many words are currently stored between the read and write pointers within the RAM.

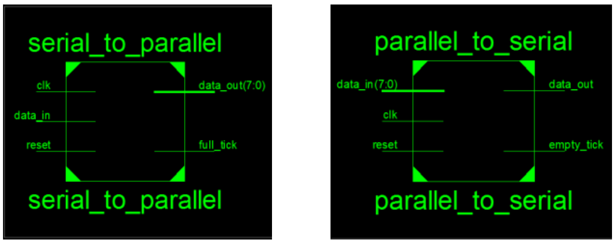
- The general working is listed as follows:
- Flip flop Update, rising edge clock sensitive
- Almost Full/Empty Flag Control, combinational
- Read and Write Pointer Control, combinational
- Memory Array Read/Write Control, rising edge clock sensitive
- Counter with Control Flags, combinational control
- Output Register Connections

D. Arbiter

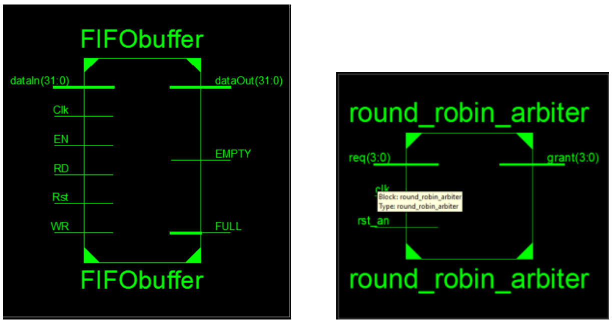
An arbiter is a logical element serving to select the order of access to a shared resource. An arbiter would typically employ a scheduling algorithm to decide which one on several requestors would be serviced. The round robin arbitration, in its basic form, is a simple time slice scheduling, allowing each requestor an equal share of the time in accessing a memory or a limited processing resource in a circular order using a fixed time slice for each requestor is inefficient as the processing time of each data element, impacts the fairness of the arbitration.

III. RESULTS

The Sub modules like Serial to parallel converter, parallel to serial converter, FIFO buffer and arbiter are coded using Verilog language & simulated in Xilinx ISE 14.5 to verify the functionality. The following figures showcase the simulation results of the same. The RTL view of these modules are shown in Fig. 5. Fig. 6 showcases the simulation waveforms of these modules.



(a) (b)

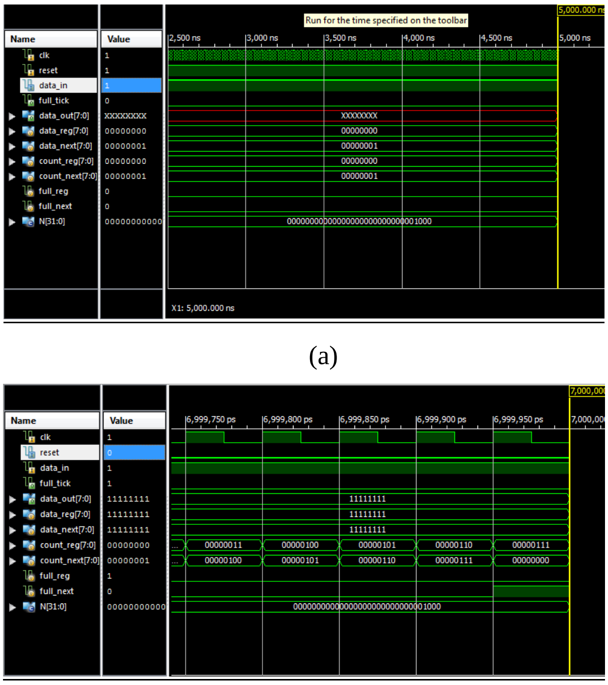


(c) (d)

Fig 5. RTL View of (a) Serial to Parallel Data Converter, (b) Parallel to Serial Data Converter, (c) FIFO Implementation and (d) Arbiter

IV. CONCLUSION

In this paper an attempt is made to design, simulate & implement the I/O functionalities like serial to parallel conversion, parallel to serial conversion, FIFO buffer and an arbiter - Xilinx Spartan. Results of each module is indicated & explained along with micro level digital design. All these functionalities implemented along with a few more proposed modules put together will be aimed towards the prototype development of Next generation High Speed programmable I/O.



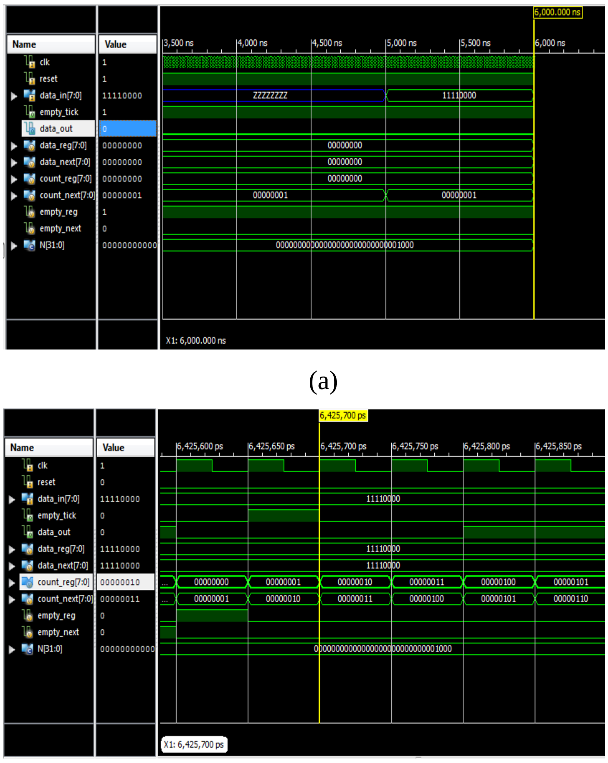
Name	Value
clk	1
reset	1
data_in	1
full_tick	0
data_out[7:0]	XXXXXXXX
data_reg[7:0]	00000000
data_next[7:0]	00000001
count_reg[7:0]	00000000
count_next[7:0]	00000001
full_reg	0
full_next	0
N[31:0]	00000000000

(a)

Name	Value
clk	1
reset	0
data_in	1
full_tick	1
data_out[7:0]	11111111
data_reg[7:0]	11111111
data_next[7:0]	11111111
count_reg[7:0]	00000000
count_next[7:0]	00000001
full_reg	1
full_next	0
N[31:0]	00000000000

(b)

Fig 6. Simulation of Serial to Parallel Data Converter when (a) reset is 1, (b) reset is 0



(a)

(b)

Fig 7. Simulation of Parallel to Serial Data Converter when (a) reset is 1, (b) reset is 0

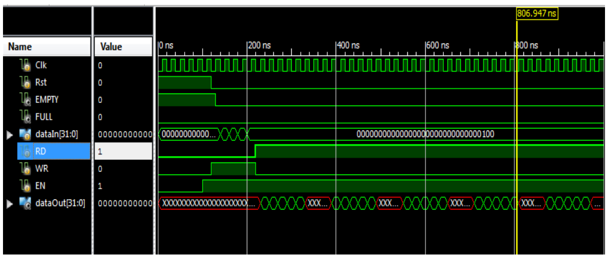


Fig 8. Simulation result of FIFO buffer

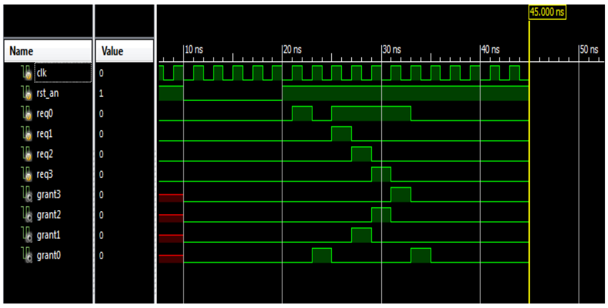


Fig 9. Simulation of Arbiter

V. FUTURE SCOPE

I/O functionalities related to the Processor IO & the related functionalities like – Programmed IO, Interrupt IO, Memory mapped IO, DMA IO, IO controller etc. need to be designed, simulated & experimented.

Also, contemporary & futuristic High-Speed Memories work on Double Data Rate [DDR] technology with specialized & complex IOs to deliver data at highest rated speeds.

Today & in future too, all the wired applications are getting itself upgraded into wireless applications & hence the IOs also need to interface with wireless devices or be a part of wireless device like Wireless Modem or Router or Switch or hub or Gateway.

REFERENCES

[1] Aruna rao B P, Shanthi Prasad M J," Overview of Differential - ended I/O Logic Families", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 6 Issue 11, November – 2017

[2] Aruna rao B P, Shanthi Prasad M J," A Technical Survey of Single-Ended Input/output (I/O) Logic Families & Key Performance Parameters", International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, UGC Approved List of Recommended Journal, Volume 8, Issue 5, May 2018).

[3] Aruna rao B P, Shanthi Prasad M J," A Comprehensive study of Input Output [IO] functionalities of Contemporary IO", at international journal of engineering development and research IJEDR 2019 | Volume 7, Issue 3 | ISSN: 2321-9939

[4] M Rabiee, "Parallel-to-Serial and Serial-to-Parallel Converters", Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition, American Society for Engineering Education, Session 1532, pp. 7.910.1-7.910.17

[5] Shilpa Kodgire, Dr. U. D. Shiurkar'" Implementation and Performance Evaluation of FIFO Buffer Design of NOC Routers", International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 5 Issue VI, June 2017 IC Value: 45.98 ISSN: 2321-9653

[6] FIFO Architecture, Functions and Applications, Texas Instruments, SCAA042A, November 1999

[7] K. Tharun Teja , PonnadaVenkata Sai Siva ," Implementation of Round Robin Arbiter using Verilog", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 4, Issue 10, October 2015, ISSN (Print) : 2320 – 3765 ISSN (Online): 2278 – 8875

[8] Mamdudul haque khan, prajoy podder, m.muktadir rahman, tatiur rahman, tanvir zaman," design of a round robin arbiter on resource sharing", International Journal Of Electrical, Electronics And Data Communication, Volume-5, Issue-4, Aprl.-2017, ISSN: 2320-2084