International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 12, December-2016

Throughput Improvement in Asynchronous FIFO Queue in Wired and Wireless Communication

Pragya Dour
Electronics and Communication Department
Sagar Institute of Research Technology and Science
RGPV University, Bhopal, India

Chhaya Kinkar
Electronics and Communication Department
Sagar Institute of Research Technology and Science
RGPV University, Bhopal, India

***Abstract*- Various technologies and algorithms are used for data transfer from one domain to another domain. First-In-First-Out (FIFO) method is the simplest among them but it requires multiple asynchronous clocks to access data. Advantage of favoring asynchronous is that the circuit can be reset with or without a clock present. The problem of multiple asynchronous clocks to access data in FIFO can be solved by the design of high throughput FIFO system which is briefly described in this paper. For achieving high throughput use of Globally Asynchronous Locally Synchronous (GALS) methodology is discussed in this paper.**

Keywords- First-In-First-Out (FIFO), Globally Asynchronous Locally Synchronous (GALS), System-On-Chip (SoC), asynchronous queue, throughput, latency.

I. INTRODUCTION

Efficient devices in every system have always been a requirement. Many factors are there which make a system well competent. In order to streamline the tasks handled by the system, a well organized system is to be designed that keeps speed, size, memory usage, amount of cache memory, bus speed and bus type optimized. For system optimization, throughput is one of the key factors, which is defined as a measure of number of units of information processed by a system in the given amount of time.

Throughput is the units of information a system can process in a specific time. It is broadly applied to computer and network systems. The productivity of a system is affected by the speed to handle workload, response time, time between a single interactive user request and the receipt of the response. The throughput may be affected by various factors, like the limitations of analog physical medium used, processing power of components used, and end-user behavior in a communication system. Throughput also measures the performance of hard drives and RAM, as well as Internet and networking connections.

In case of the Central Processing Unit of a computer system, the scheduler allows an enterprise to manage and track computer tasks [1]. It tries to maintain maximum throughput, minimize response time or minimizing latency or maximizing fairness (equal CPU time to each process, or proper time distribution according to the priority and workload of processes). Practically, some conflict arises (e.g. throughput versus latency), so compromise is considered. Many preferences are there according to the needs and objectives of the user.

The technique to increase overall system throughput is that the several programs are kept in memory meanwhile the processor run other process when another process is waiting for an input-output operation [2]. This is termed as multiprogramming. The next technique is to preempt a process and allowing another process run and doing this fast to give the illusion that many processes are executing at the same time. This is what termed as multitasking.

II. LITERATURE SURVEY

A. Clifford E. Cummings et al.

Synchronisation of FIFO pointers into the opposite clock domain accomplished by Gray code pointers, dual n-bit gray code counter for synchronisation & comparison , sampling & handshaking control signals of binary pointer [3].

Finding FIFO design errors typically require simulation of gate level FIFO design with back annotation of actual delays & include approximation.

B. Xiao Yong, Zhou Runde et al.

Description and analysis of circular architecture FIFO to deliver a new type of FIFO suitable for GALS systems [4]. This paper includes somewhat complicated design and algorithms.

C. Hosuk Han, Kenneth S. Stevens et al.

FIFO structures, first order equations are derived to model the capacity latency, & maximum throughput based on occupancy of designs [5].

For this design throughput is reduced due to delay in all cases except for FIFO of even length that are exactly half full.

D. Dadhania Prashant C

One method that is used to design, synthesise and analyze a safe FIFO between different clock domains using Gray code pointers that are synchronised into a different clock domain before testing for 'FIFO full' or "FIFO empty" conditions [6]. The fully coded, synthesised and analysed RTL Verilog is included.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 12, December-2016

In some cases, it is difficult to notice minute details from pointer generation technique to full & empty generation & finding FIFO error need simulation of gate level FIFO design.

E. Amit Kumar, Shankar, Neeraj Sharma

Data sent by Write clock domain to asynchronous FIFO read clock domain & 8-bit data are stored at the memory location pointed by the adder [7].

Metastability occurs in the multi clock domain when the output from one clock domain changes at the rising edge of the second clock domain which may lead to wrong results.

F. Hyoung-Kook Kim et al.

A test method for testing two-D-flip-flop synchronizers in an asynchronous first-in first-out (FIFO) interface [8].

Prior to implement the proposed test method, the channel delay compensator, delayed scan enable signal generator, launch clock generator and capture clock generator should be designed.

III. FIFO TECHNOLOGY

For better task handling or management of incoming tasks for a system a particular strategic approach is required. Several techniques are used to perform this task. FIFO (First In First Out) is one of the basic approaches to manage the incoming data. FIFO is utilized to organize and manipulate the data buffer, where the oldest processed first. It is one of the scheduling algorithms in operating system of Central Processing Unit (CPU) performs various tasks as desired. Queueing theory includes FIFO, which process data structures, along with cooperation among various FIFO queues.

FIFO scheduling algorithm can be vastly used, for example in disk controllers to serve the requests in definite sequence. FIFO method is used in communication network bridges, routers and switches and to send data packets to their corresponding targets. In some devices multiple FIFOs are used simultaneously in order to queue various information.

Generally, FIFOs are used in electronics in order to buffer and control the data flow among hardware and software components. A hardware FIFO consists of control logic, storage and read and write pointers. Storage can be latches, static random access memory (SRAM), flip-flops etc. Dual-port SRAM with writing and reading ports is used for FIFOs of bigger size.

A synchronous FIFO uses same clock for reading and writing, whereas asynchronous FIFO uses different clocks for reading and writing purposes [9]. The reliable flag generation is possible with implementation of an asynchronous FIFO using Gray code for the read and write pointers. It's noteworthy to use pointer arithmetic in order to generate flags in asynchronous FIFO implementation purposes. Moreover, leaky bucket approach or pointer arithmetic is used to generate flags in synchronous FIFO implementation. FIFO status flag's examples are as: full, empty, almost full, almost empty, etc. As shown in Fig.1.

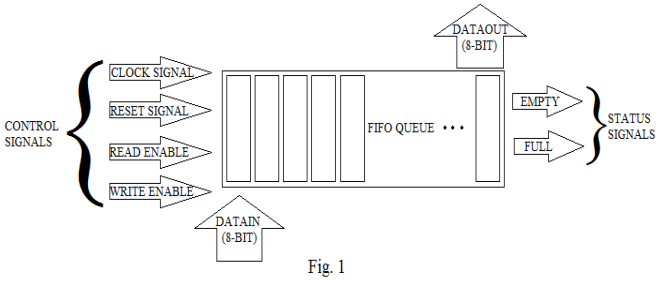

Fig. 1

IV. GALS TECHNOLOGY

Globally asynchronous locally synchronous (GALS) is a Model of Computation (MoC) which utilizes both synchronous programming as well as asynchronous programming as per requirement [10]. It eases to get rid of the synchrony assumption partially through modeling and designing computer systems consisting of several synchronous islands (the program of each such island obeying the synchronous programming) communicating with each other asynchronously, e.g., FIFOs. Asynchronous wrappers are there in GALS circuit for communication of locally synchronous modules [11].

An independent clock is provided in every synchronous subsystem (clock domain), in order to run it. Low electromagnetic interference (EMI) is one of the advantages of this system [12]. While changing the state from 0 to 1 the CMOS circuit or any logic gate requires huge supply current. As maximum changes are started by an active clock edge, the changes get accumulated. Therefore, at active clock edges larger spikes in supply current appears. Circuit malfunction is generated as spikes cause large electromagnetic interference. Decoupling capacitors are one of the solutions to control the spikes. One more way to solve the above problem is to use a GALS technology, that is design locally synchronous (easier than asynchronous one) but globally asynchronous, i.e. there are different (for example phase shifted, rising and falling active edges) clock signal regimes thereby spikes of supply current do not get accumulated eventually. Therefore, GALS design style is used in system-on-a-chip (SoC).

V. DESIGNING PLATFORM

XC3S50 device of Spartan-3 family is used in this asynchronous FIFO queue designing. This is shown in Table.I.

The Spartan-3 Generation of FPGAs offers a choice of various platforms, each delivering a unique cost-optimized balance of programmable logic, connectivity, and dedicated hard IP for our low-cost applications. Spartan-3 is provides Highest Density and Pin-Count Applications. For it both high logic density and high I/O count are important. It is ideal for highly-integrated data-processing applications.

The specifications are as follows:-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 12, December-2016

TABLE . I

Property Name	Value
Family	Spartan3
Device	XC3S50
Package	PQ208
Speed	-5
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Enhanced Design Summary	Enabled
Property Specification in Project File	Store all values
VHDL Source Analysis Standard	VHDL-93

VI. METHODOLOGY USED

The working of the system is represented in the following flowcharts:-

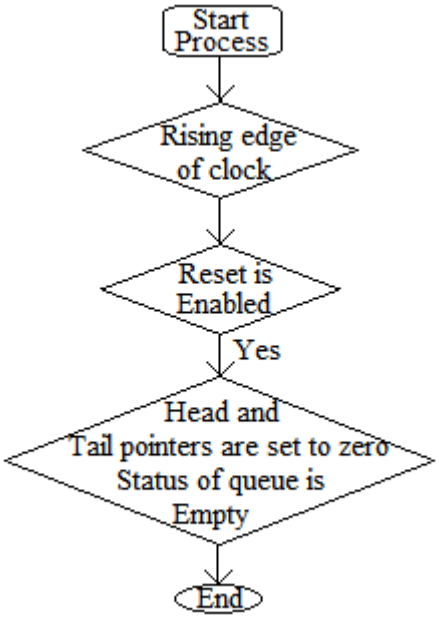

Fig. 2

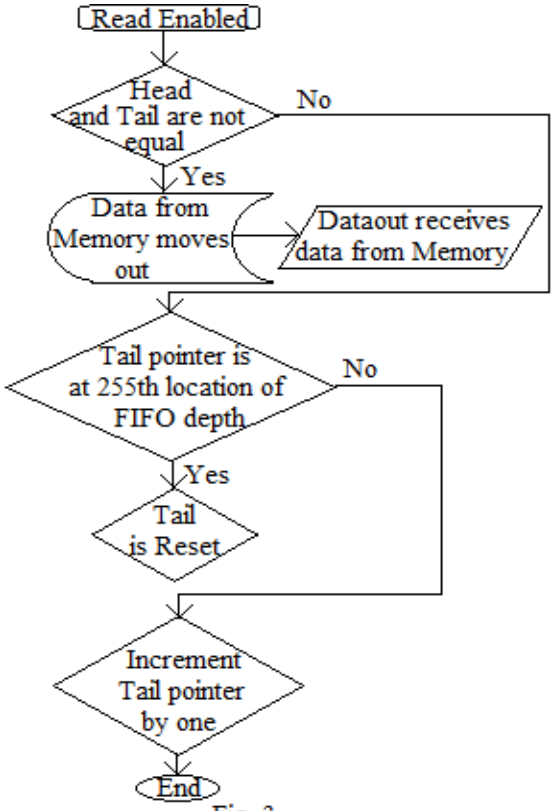

Fig. 3

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 12, December-2016

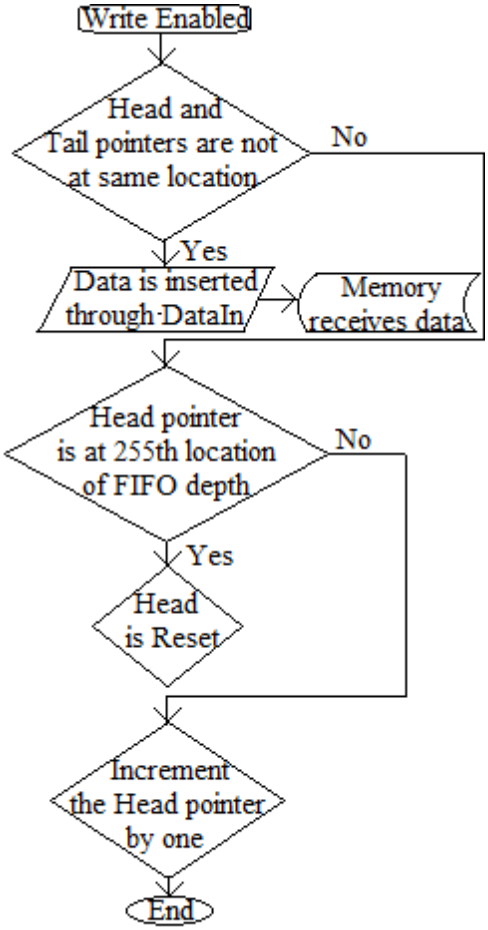

Fig. 4

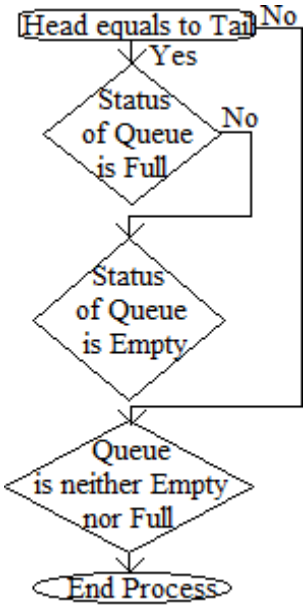

Fig. 5

VII. RESULT

In the RTL obtained the last element shown ranges from Memory_0_and0000_imp to Memory_225_and0000_imp.

The RTL Schematic is as shown in Fig.6.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 12, December-2016

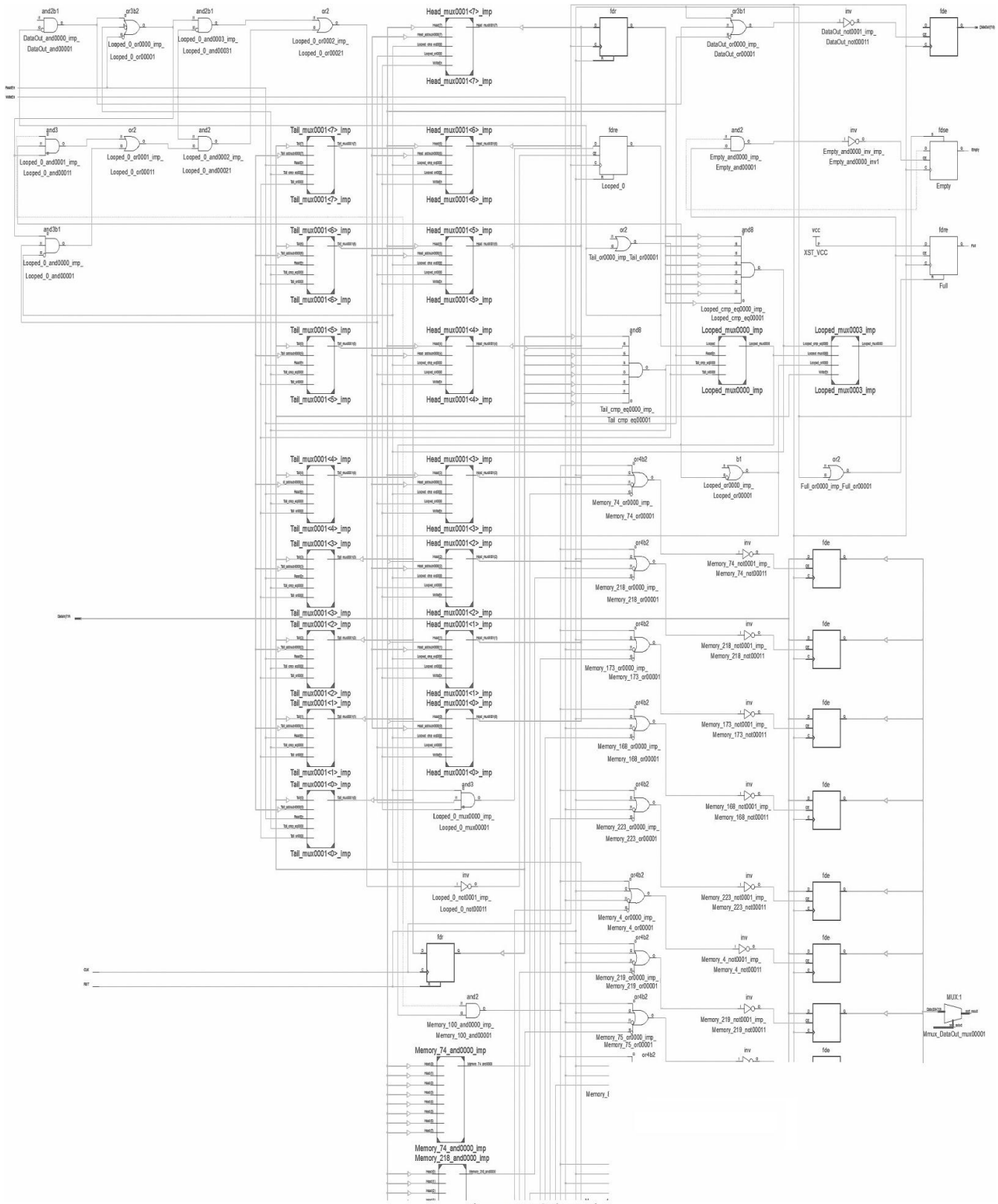

Fig. 6 RTL Schematic

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 12, December-2016

The Testbench Waveform is shown in Fig. 7.

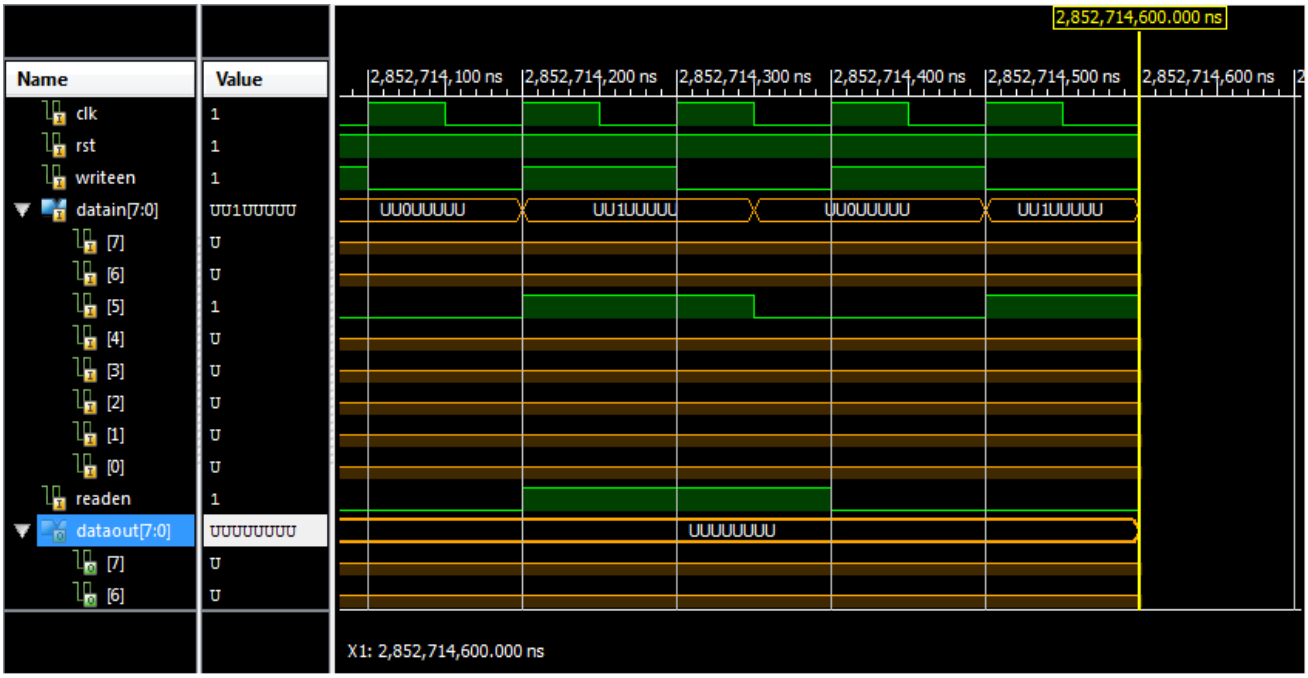

Fig. 7 (a) Testbench Waveform

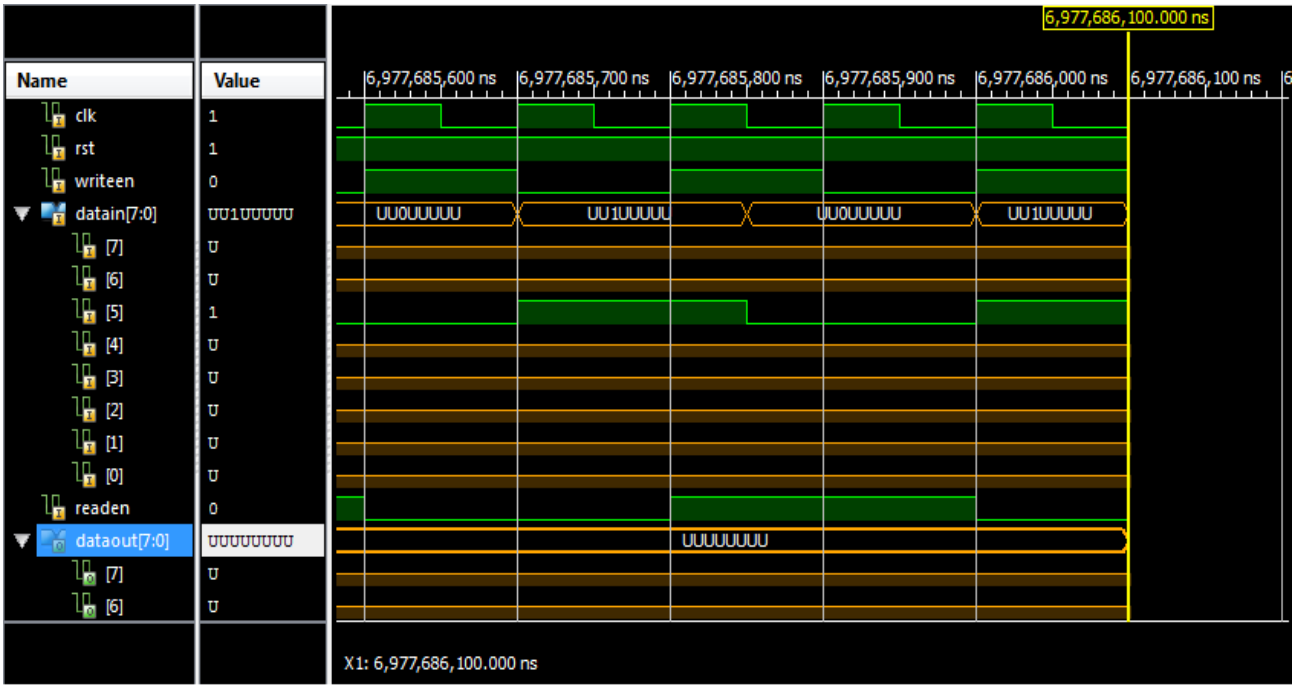

Fig. 7 (b) Testbench Waveform

VII. CONCLUSION

The execution of the queue program is successfully done in Xilinx ISE Design Suite Software with syntax checked. The control signals using programming logic performed the task of managing multiple tasks assigned to queue system. Status signals described the status of the queue as full and empty respectively. Respective test bench waveforms are simulated for high throughput.

REFERENCES

[1] Silberschatz Galvin, Operating System Concepts.
[2] Dr.Guy Tel-Zur, Ben-Gurion University, Israel, The Importance of High-Performance and High-Throughput Computing, HPC Advisory Council, Israel Supercomputing Conference, February 7, 2012.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 12, December-2016

[3] Clifford E. Cummings, Peter Alfke, Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons, SNUG-2002, San Jose, CA, 2002.
[4] Xiao Yong, Zhou Runde , IEEE, Low Latency High Throughput Circular Asynchronous FIFO, Institute of Microelectronics, Tsinghua University, 2008: 812-816.
[5] HoSuk Han Kenneth S.Stevens, IEEE, Clocked & Asynchronous FIFO Characterization & Comparison, 2009.
[6] Dadhania Prashant C, Designing Asynchronous FIFO, Journal of Information, Knowledge and Research in Electronics and Communication Engineering, ISSN: 0975 – 6779, Nov 12 To Oct 13, Volume – 02, Issue – 02, 2002:561-563.
[7] Amit Kumar, Shankar, Neeraj Sharma, Verification of Asynchronous FIFO using System Verilog, International Journal of Computer Applications (0975 –8887)Volume 86– No 11, January 2014:16-20.
[8] Hyoung-Kook Kim, Laung-Terng Wang, Yu-Liang, Wen-Ben Jone, Testing of Synchronizers in Asynchronous FIFO, Journal of Electronic Testing 29(1), February 2013:49-72.
[9] Jiann S. Yuan, Senior Member, IEEE, and Weidong Kuang, Teaching Asynchronous Design in Digital Integrated Circuits, 2004:397-404.
[10] Lin Shijun, Su Li , JIN Depeng , Zeng Lieguang, Universal GALS Platform and Evaluation Methodology for Networks-on-Chip, 2009: 176-182.
[11] James S. Guido, Graduate Student Member, IEEE, and Alexandre Yakovlev, Senior Member, IEEE, Design of Self-Timed Reconfigurable Controllers for Parallel Synchronization via Wagging, 2011.
[12] Frances Ann Hill, Eric Vincent Heubel, Philip Ponce de Leon, and Luis Fernando Velásquez-García,Senior Member, IEEE, High-Throughput Ionic Liquid Ion Sources Using Arrays of Microfabricated Electrospray Emitters With Integrated Extractor Grid and Carbon Nanotube Flow Control Structures, 2014:1237-1247.