GALS Technology to Improve Throughput of FIFO

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

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

ABSTRACT
An efficient high throughput FIFO (First-In-First-Out) system using GALS (Globally Asynchronous Locally Synchronous) technology is designed for data transfer from one domain to another domain with the development of a modeling and simulation framework whoseresults are obtained as RTL(Register-Transfer Level) Schematic. Integration of several of IP (Intellectual Property) cores into a single chip in order to fulfill the demand of latest applications, leads to various timing issues especially interfacing between the different clock domains. The GALS technology provides a clock distribution feature for the same. A general purpose 8-bit synchronous core designfavoringthe GALS technology is used for the designing. The model is implemented in VHDL (Very High Speed Integrated Circuits Hardware Description Language) with Xilinx ISE (Integrated Synthesis Environment) Design Suite 14.5 Version software and simulated using ISim tool. The synthesis results show improved throughput andreduced chip area usingGALS.

Keywords
FIFO (First-In-First-Out), GALS (Globally Asynchronous Locally Synchronous), RTL (Register-Transfer Level) Schematic, System-On-Chip (SoC), IC (Integrated Circuit), throughput, chip area.

1. INTRODUCTION
First-in-first-out (FIFO) is one of the basic methods to handle the incoming data in any task management system whether it is wired or wireless [1]. The synchronization of the internal components has the advantage that the reset presented to all functional flip-flops is fully synchronous to the clock present and will always meet the reset recovery time. On the other hand,in asynchronous system the circuit can be reset with or without a clock present. Moreover, high speeds can be achieved in asynchronous circuits as the data path is independent of reset signal.

The individual advantages of synchronous and asynchronous concept in different areas lead us to the design of the system partially synchronous and partially asynchronous.

2. GLOBALLY ASYNCHRONOUS LOCALLY SYNCHRONOUS
As the size of equipments of latest technology isshrinking, it is becoming difficult and expensive to distribute a global clock signal with low skew throughout a single processor [2]. Asynchronous processor designs overcome this trouble as they do not possess a global clock. It is difficult to design fully asynchronous system due to metastability issues. Therefore, Globally Asynchronous Locally Synchronous (orGALS) system is taken into account. The flexibility in independently controllable local clocks provides the effective use of another energy conservation technique like dynamic voltage scaling [3].For some designs of GALS like a 5-clock domain GALS processor, the power consumptionreduces by 10%.

The gap between fully synchronous and GALS implementations is decreased by designing of fine-grained voltage scaling along with enhanced power efficiency.Designs of conventional microprocessor are synchronous [4]. In these circuits, a common clock signal is there as a timing reference for the whole circuit for all tasks. Although, asynchronous system made up of self-timed circuits lacks any global timing reference[5].

GALS system consists multiple individual synchronous modules which work with their own local clocks and communicate asynchronously with another modules.The prominent feature of this system is the presence of global timing reference and use of various individual local clocks (or clock domains), possibly running at different frequencies. GALS design is preferred more because of its global clock distribution feature.

The increase in die sizes and increasing transistor counts would result in expensive distribution of high-frequency global clock signals with lower skew throughout a large die when design effort, die area, and power dissipation are considered. Requirement of careful design and fine-tuning of a global clock distribution network is somewhat removed while using GALS system. IP cores and system-on-chip (SoC) are gaining popularity among the designers now-a-days [6]. Multiple cores on single chip would not have possible with the with a single clock system; every core has different clock requirement and separate operating frequency. GALS systems with a particular asynchronous interface will ease the design reuse [7].

In the microprocessors field, global clock distribution issue is the prominent reason to study the GALS system design. The development of a modeling and simulation framework and its corresponding results as synthesized circuit using GALS is described in this paper.

3. DESIGN
Various timing issues are generated on integration of several of IP cores into a single chip in order to fulfill the increasing demand of latest applications, most prominently interfacing between the different clock domains. These issues are better managed by the GALS technique that divides a chip into several independent local subsystems working on different clock signals, along with keeping the system optimized.

4. RESULT
The RTL Schematic as a result of implementation program in Xilinx is as shown in Fig 1 and Fig. 4 using VHDL [8]. It is followed by their respective testbench waveforms obtained during simulation using ISim simulator [9]. The numbers of address bits are2-bit and latch of 4-bit for RTL Schematic as

shown in Fig.2, whereas for RTL Schematic shown in Fig. 5, (component names are also changed) address is of 8-bit and latchof 16-bit. It is clear from the RTL's that even on increasing the address bits number of components remains unchanged, which shows reduction in chip area.

The behavioral model of simulation in the form of corresponding testbench waveforms in Fig. 3 and Fig. 6 shows the decrement in time period of waveform from nanoseconds to picoseconds, which indicates the improvement in throughput of FIFO design [10, 11].

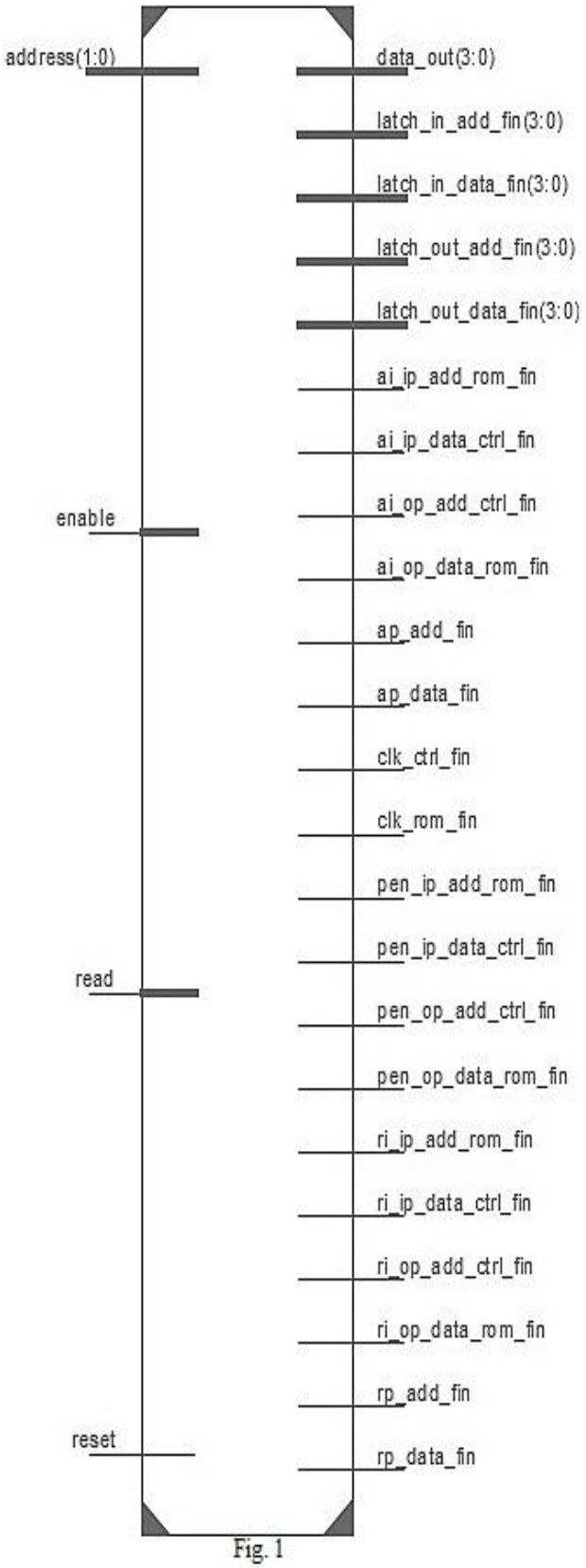

Fig. 1

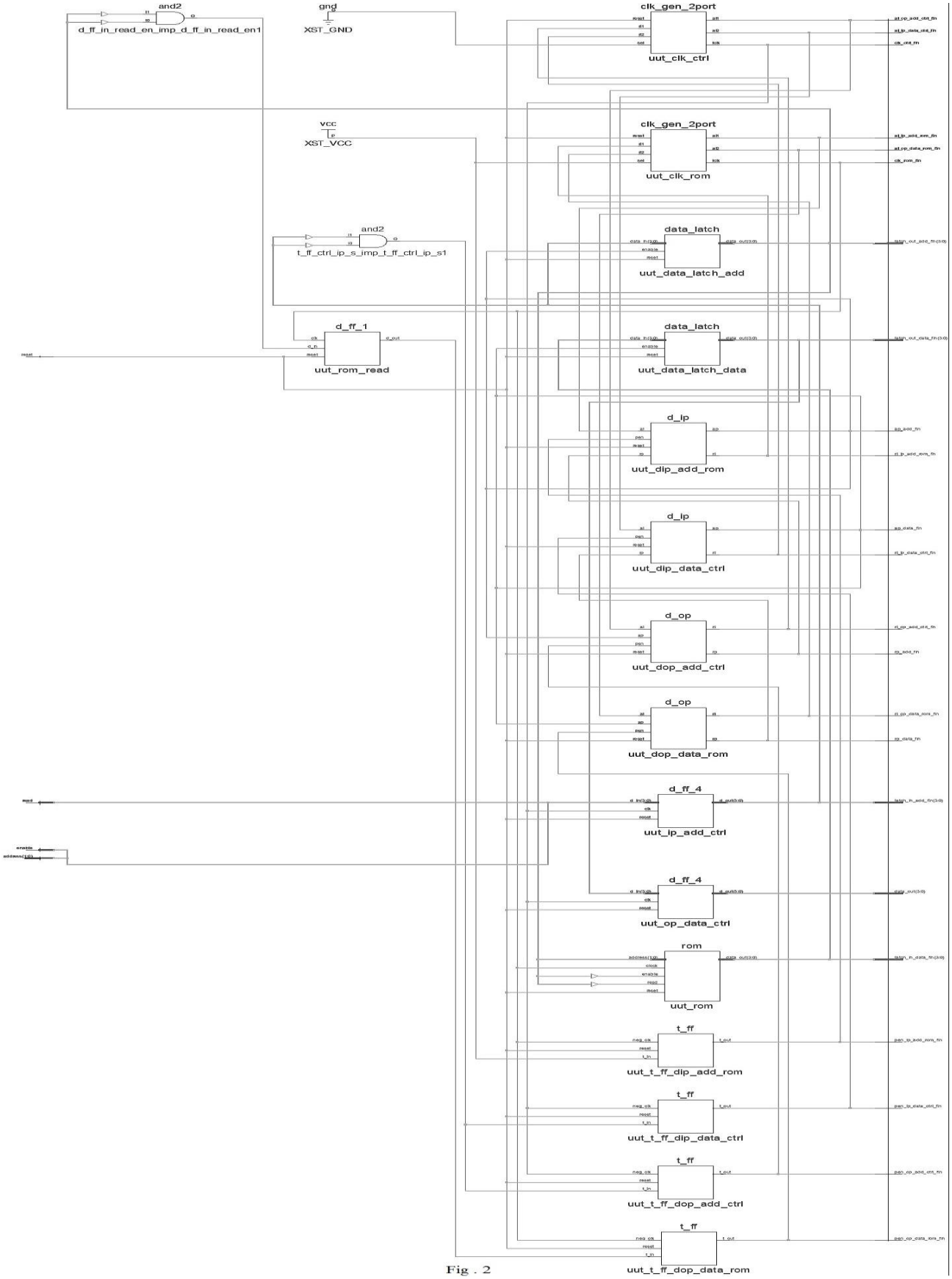
and2
d_ff_in_read_en_imp_d_ff_in_read_en1
gnd
XST_GND
clk_gen_2port
uut_clk_ctrl
vcc
XST_VCC
clk_gen_2port
uut_clk_rom
and2
t_ff_ctrl_ip_s_imp_t_ff_ctrl_ip_s1
data_latch
uut_data_latch_add
d_ff_1
uut_rom_read
data_latch
uut_data_latch_data
d_ip
uut_dip_add_rom
d_ip
uut_dip_data_ctrl
d_op
uut_dop_add_ctrl
d_op
uut_dop_data_rom
d_ff_4
uut_ip_add_ctrl
d_ff_4
uut_op_data_ctrl
rom
uut_rom
t_ff
uut_t_ff_dip_add_rom
t_ff
uut_t_ff_dip_data_ctrl
t_ff
uut_t_ff_dop_add_ctrl
t_ff
uut_t_ff_dop_data_rom

Fig . 2

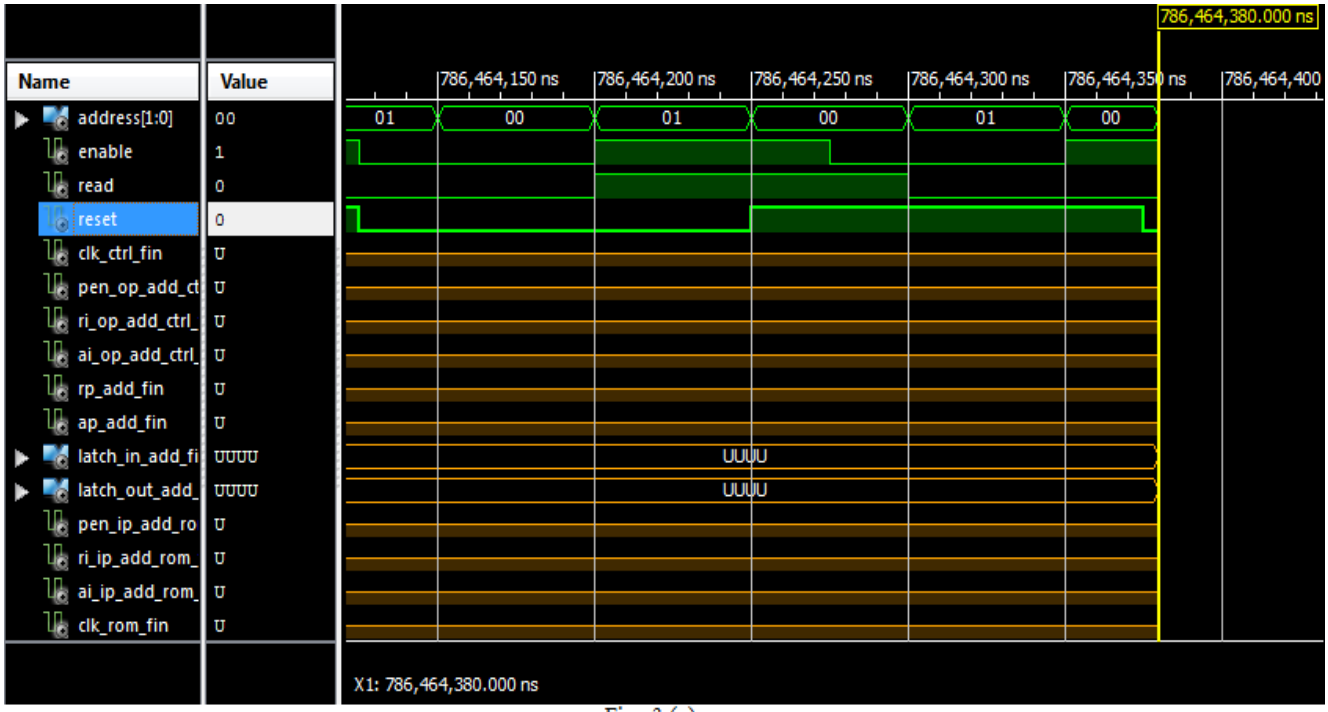
Name
Value
address[1:0]
00
enable
1
read
0
reset
0
clk_ctrl_fin
U
pen_op_add_ct
U
ri_op_add_ctrl_
U
ai_op_add_ctrl_
U
rp_add_fin
U
ap_add_fin
U
latch_in_add_fi
UUUU
latch_out_add_
UUUU
pen_ip_add_ro
U
ri_ip_add_rom_
U
ai_ip_add_rom_
U
clk_rom_fin
U
786,464,380.000 ns
786,464,150 ns
786,464,200 ns
786,464,250 ns
786,464,300 ns
786,464,350 ns
786,464,400
01
00
01
00
01
00
UUUU
UUUU
X1: 786,464,380.000 ns

Fig . 3 (a)

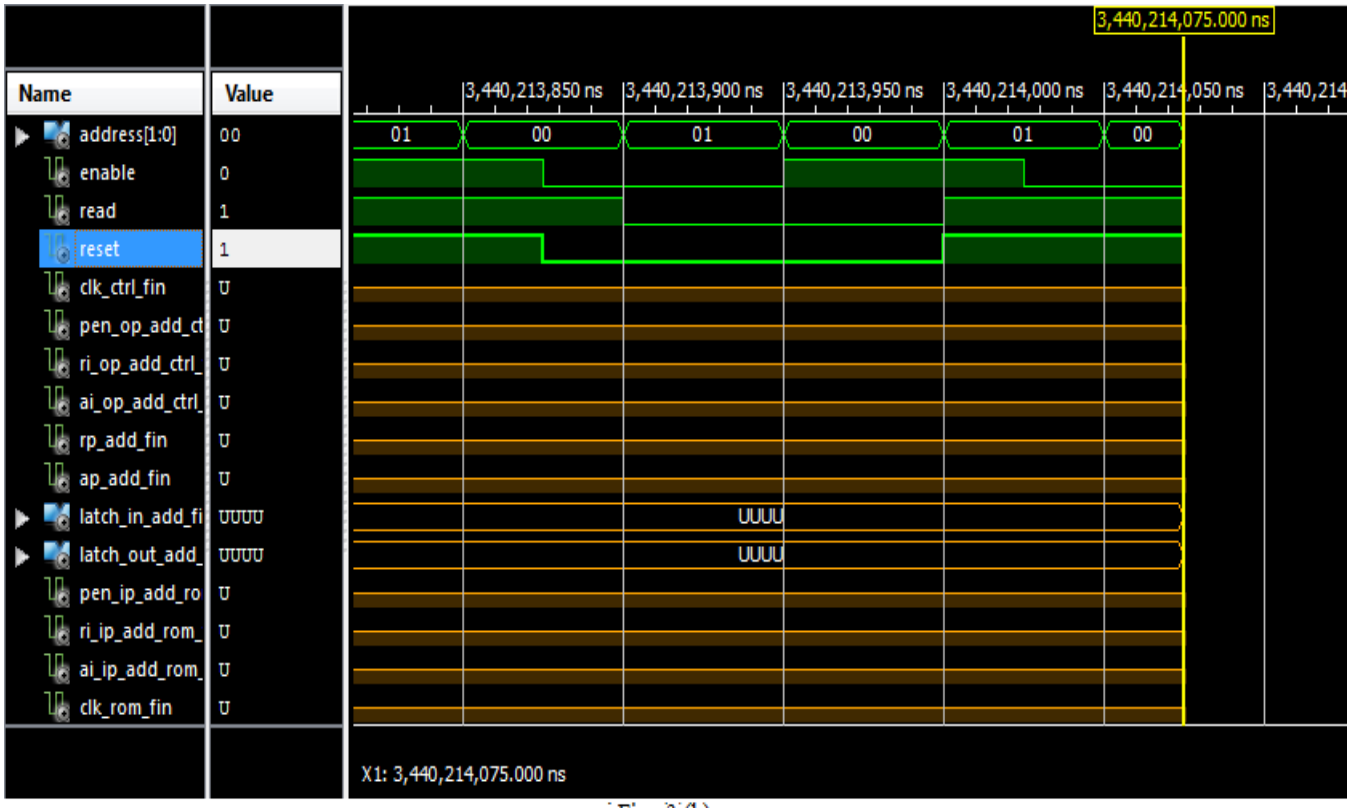
Name
Value
address[1:0]
00
enable
0
read
1
reset
1
clk_ctrl_fin
U
pen_op_add_ct
U
ri_op_add_ctrl_
U
ai_op_add_ctrl_
U
rp_add_fin
U
ap_add_fin
U
latch_in_add_fi
UUUU
latch_out_add_
UUUU
pen_ip_add_ro
U
ri_ip_add_rom_
U
ai_ip_add_rom_
U
clk_rom_fin
U
3,440,214,075.000 ns
3,440,213,850 ns
3,440,213,900 ns
3,440,213,950 ns
3,440,214,000 ns
3,440,214,050 ns
3,440,214
01
00
01
00
01
00
UUUU
UUUU
X1: 3,440,214,075.000 ns

Fig . 3 (b)

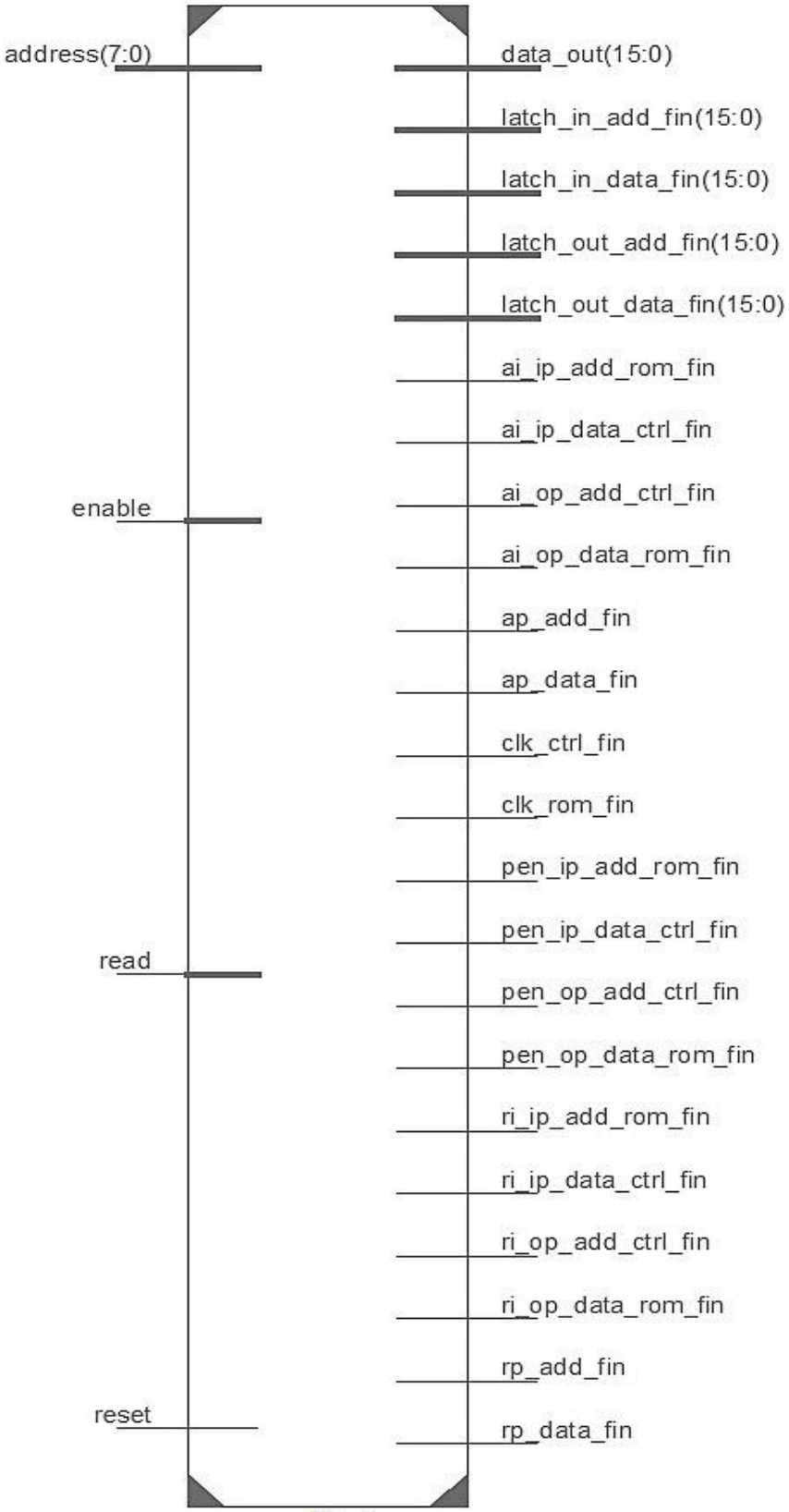
address(7:0)
enable
read
reset
data_out(15:0)
latch_in_add_fin(15:0)
latch_in_data_fin(15:0)
latch_out_add_fin(15:0)
latch_out_data_fin(15:0)
ai_ip_add_rom_fin
ai_ip_data_ctrl_fin
ai_op_add_ctrl_fin
ai_op_data_rom_fin
ap_add_fin
ap_data_fin
clk_ctrl_fin
clk_rom_fin
pen_ip_add_rom_fin
pen_ip_data_ctrl_fin
pen_op_add_ctrl_fin
pen_op_data_rom_fin
ri_ip_add_rom_fin
ri_ip_data_ctrl_fin
ri_op_add_ctrl_fin
ri_op_data_rom_fin
rp_add_fin
rp_data_fin

Fig . 4

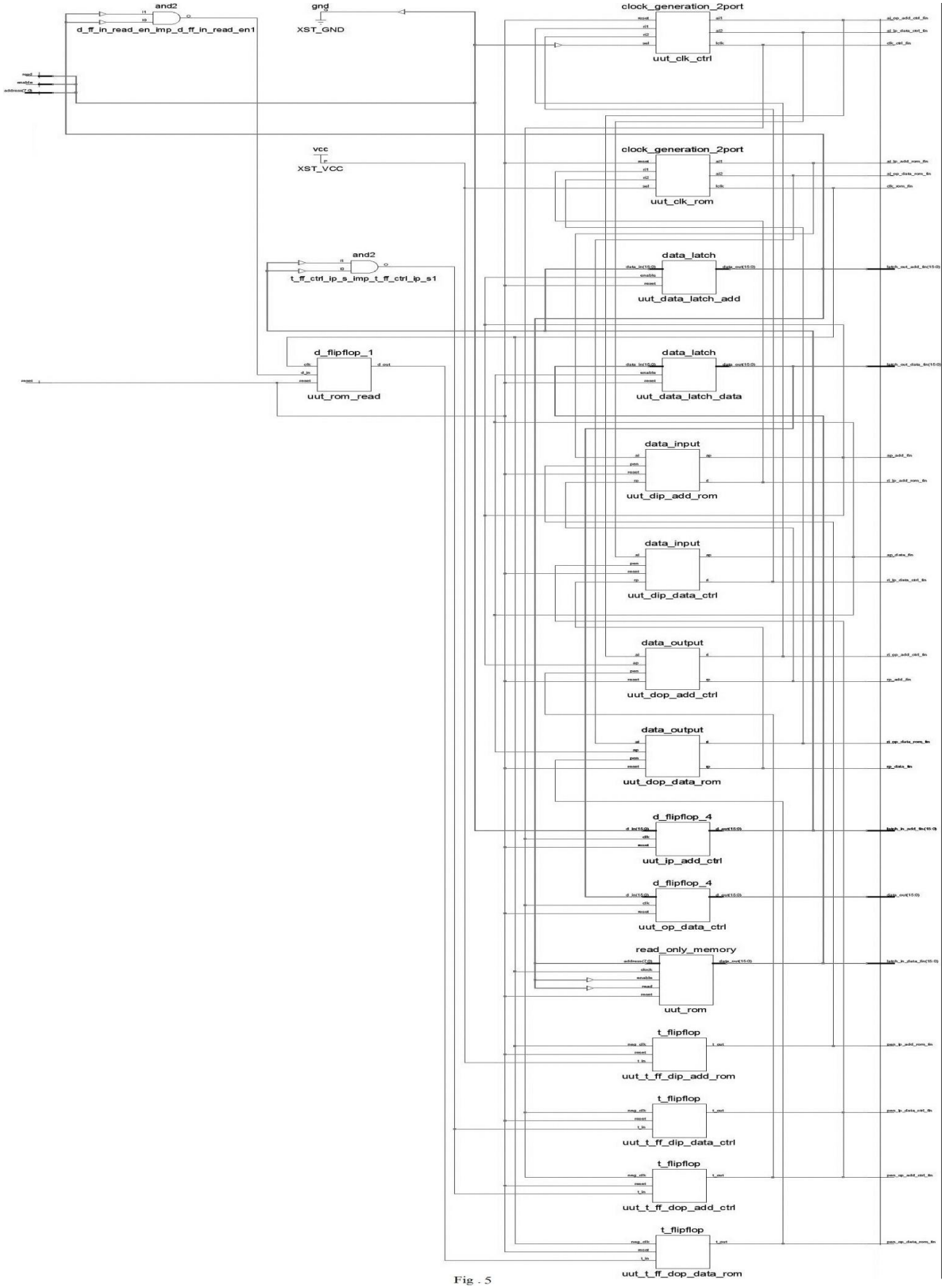
and2
d_ff_in_read_en_imp_d_ff_in_read_en1
gnd
XST_GND
vcc
XST_VCC
and2
t_ff_ctrl_ip_s_imp_t_ff_ctrl_ip_s1
d_flipflop_1
uut_rom_read
clock_generation_2port
uut_clk_ctrl
clock_generation_2port
uut_clk_rom
data_latch
uut_data_latch_add
data_latch
uut_data_latch_data
data_input
uut_dip_add_rom
data_input
uut_dip_data_ctrl
data_output
uut_dop_add_ctrl
data_output
uut_dop_data_rom
d_flipflop_4
uut_ip_add_ctrl
d_flipflop_4
uut_op_data_ctrl
read_only_memory
uut_rom
t_flipflop
uut_t_ff_dip_add_rom
t_flipflop
uut_t_ff_dip_data_ctrl
t_flipflop
uut_t_ff_dop_add_ctrl
t_flipflop
uut_t_ff_dop_data_rom

Fig . 5

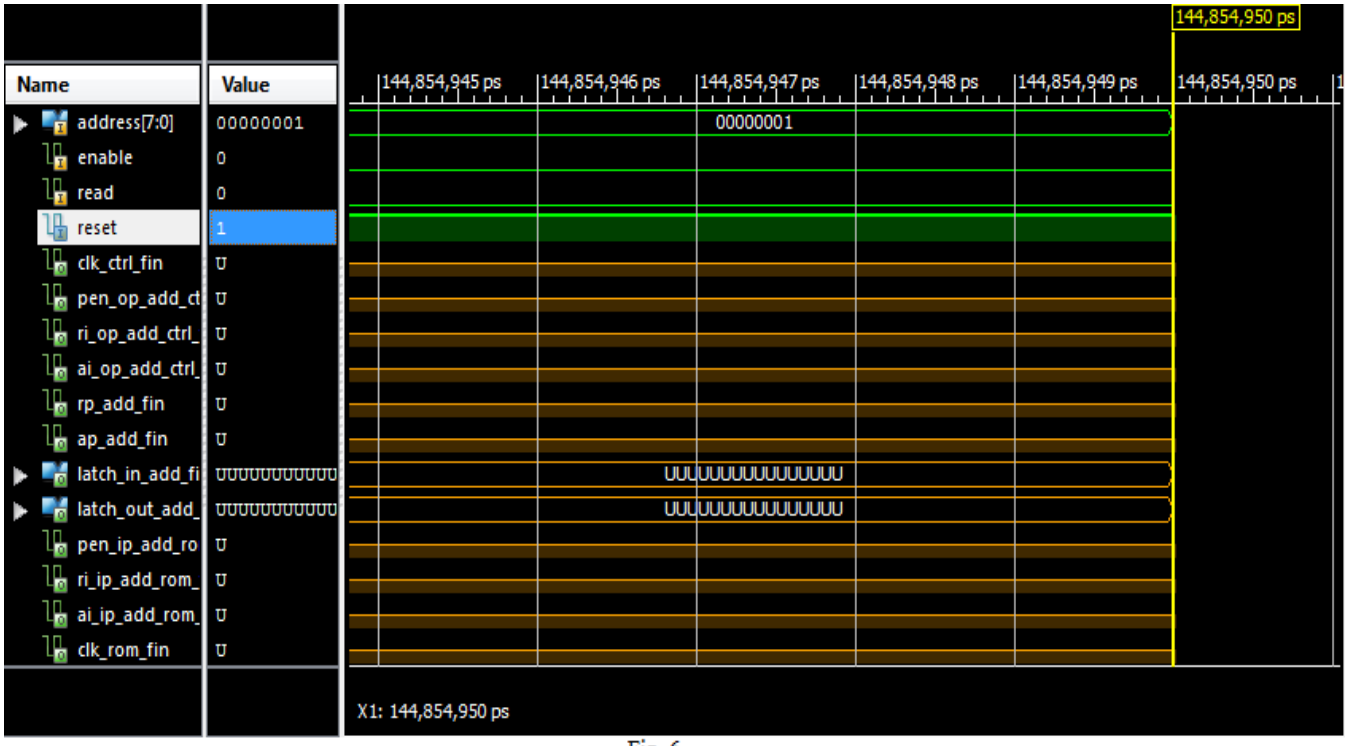

Fig. 6

5. CONCLUSION

A general purpose 8-bit synchronous core isdesigned firstly and then converted into GALS core to improve the throughput of the FIFO system design. The model thus obtained is implemented in VHDL using Xilinx ISE 14.5 Version software and simulated using ISim tool. The RTL Schematic shows the improved throughput and less area using GALS technology [12]. This core can be integrated into a single (IC) Integrated Chip [13] to generate a multi core system for further designing in future.

6. REFERENCES

[1] Silberschatz Galvin, *Operating System Concepts*.

[2] HoSuk Han Kenneth S.Stevens, IEEE, *Clocked & Asynchronous FIFO Characterization & Comparison*, 2009.

[3] Abbas Rahimi, Mostafa E. Salehi, SiamakMohammadi, Sied Mehdi Fakhraie, Microelectronics Journal 42, Elsevier, *Low-Energy Gals Noc with FIFO—Monitoring Dynamic Voltage Scaling*, , 2011: 889-896.

[4] Hyoung-Kook Kim, Laung-Terng Wang, Yu-Liang, Wen-Ben Jone,Journal of Electronic Testing 29(1), *Testing of Synchronizers in Asynchronous FIFO*, February 2013:49-72.

[5] 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.

[6] Lin Shijun, Su Li , JIN Depeng , Zeng Lieguang, *Universal GALS Platform and Evaluation Methodology for Networks-on-Chip*, 2009: 176-182.

[7] Dadhania Prashant C, Journal of Information, Knowledge and Research in Electronics and Communication Engineering, ISSN: 0975 – 6779, Nov 12 To Oct 13, Volume – 02, Issue – 02, *Designing Asynchronous FIFO*, 2002:561-563.

[8] Amit Kumar, Shankar, Neeraj Sharma, International Journal of Computer Applications (0975 –8887)Volume 86–No 11, *Verification of Asynchronous FIFO using System Verilog*, January 2014:16-20.

[9] Clifford E. Cummings, Peter Alfke, SNUG-2002, San Jose, CA,*Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons*,2002.

[10] 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 Micro fabricatedElectro spray Emitters With Integrated Extractor Grid and Carbon Nanotube Flow Control Structures*, 2014:1237-1247.

[11] 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.

[12] Xiao Yong, Zhou Runde, IEEE, *Low Latency High Throughput Circular Asynchronous FIFO*, Institute of Microelectronics, Tsinghua University, 2008: 812-816.

[13]Jiann S. Yuan, Senior Member, IEEE, and WeidongKuang, *Teaching Asynchronous Design in Digital Integrated Circuits*, 2004:397-404.