RESEARCH ARTICLE OPEN ACCESS

Implementation of AXI Master IP for High-Speed Data Transfer Between PL and PS in Zynq Board

Mahammad Rafi Shaik

Student, Department of VLSI Engineering,The Veda Educational Society,MRP Towers, Vidyanagar 1st Lane, Guntur, Andhra Pradesh,India.

Rafi446.shaik@gmail.com

----------**********----------

Abstract:

Nowadays ZYNQ boards are widely using In Many applications like Aerospace, Defense and, Medical Electronics, Automotive, Broadcast, Security systems, Video & Image Processing, etc. Zynq 7010 board consists of Programmable logic (PL) and Processing System (PS). In the above applications, Communication logic is needed for data transfer between PS and PL. For that there is a basic method using AXI GPIO (General Purpose Input/Output), but it consumes more CPU time. Instead of that, if we create a custom AXI Master IP (intellectual property), we can transfer the data with high speed and less involvement of the CPU.

Keywords **—ZYNQ- 7010, AXI Master IP, high-Speed Data, Vivado, SDK, GPIO.**

----------**********----------

I. INTRODUCTION

The Zynq Board is having full of features and is easy to use. The Xillinxfamily Z-7010 board contains All Programmable System-on-Chip (AP SoC), which is Integrated by a dual-core ARM Cortex-A9 processor and Xilinx 7-series Field Programmable Gate Array (FPGA) logic [7]. And also have multimedia ports and I/O ports with this the Zynq Z-7010 can host a whole system design.

Vivado Design Suite is a software developed by Xillinx. The HDL (hardware description language) designs created on Zynq Board, are synthesized and analysed by Vivado Design Suite, this Suite contains ISE (Integrated Synthesis Environment) [10]. It supports both VHDL and Verilog languages.

AXI GPIO is a Soft IP core, used to transfer data between PL and PS. It contains AXI4-Lite interface, and it does not support Direct Memory Access (DMA) and IP Master Services [3]. In applications where bulk data transfer is not required there, AXI GPIOs are used.

In Vivado Software there is a feature with that we can create, a customized IP with AXI Master interface, which means it's having direct memory access support and IP Master Services. With this feature, we can transfer data with high speed between PL and PS with less involvement of the CPU. This paper will give you the performance comparison between AXI GPIO and AXI Master IP when transferring the data between PL and PS.

II. DATA TRANSFER USING AXI GPIO

Zynq board mainly containing two parts PL and PS. Here in the PL part small logic is implemented for generate some data and transfer it to the PS part. Two counters named counter1 and counter2, along with FIFO also implemented. Discussed about the components used in the GPIO based transfer method below.

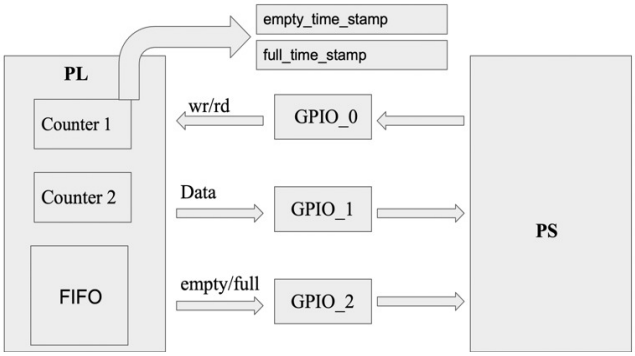

Fig. 1 Block Diagram of the Data transfer using AXI GPIO

- Counters: Counter1 and counter2 both are 32bit size counters
- FIFO: Size of the FIFO is 8192 samples with 4-Byte size of each sample
- GPIOs: Three GPIOs are used for communication logic between PL and PS. GPIO_1 used for Data transfer purpose and GPIO_0 and GPIO_2 is used for handshaking purpose.
- Latches: empty_time_stamp andfull_time_stamp are the two latches used to latch the value of counter1.

First, the data request initiated from PS to PL by sending WR signal through GPIO_0. Whenever WR signal received from PS the two counters in PL part are start counting, at every posedge of the clock the Counter2 value stored FIFO. After some time, the FIFO will be full at that time what is the value present in Counter1 latched into theFull_time_stampregister.

Whenever the FIFO is full then the FULL signal is sent to the PS through GPIO_2. After that PS will send the RD signal to the PL, then the data presented in the FIFO read sample by sample through GPIO_0. After some time, FIFO will be empty at that time what is the value present in the Counter2 latched into the Empty_time_stamp register and EMPTY signal sent to PS through GPIO_2.

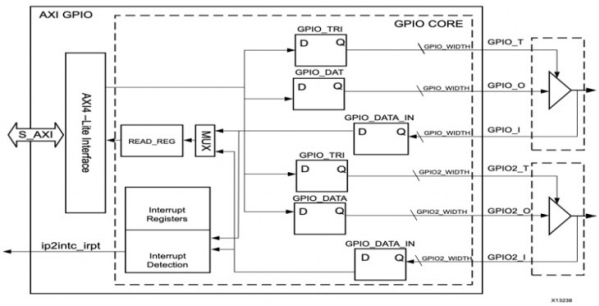

Fig. 2 AXI GPIO Block diagram

The AXI GPIO core used as a general-purpose input/output interface to the AXI interface. This 32-bit IP core is designed to interface with the AXI4-Lite interface [3].

- Supports the AXI4-Lite interface specification
- configurable single or dual GPIO channel(s)
- configurable channel width for GPIO pins from 1 to 32 bits
- dynamic programming of every GPIO bit as input or output
- individual configuration of every channel
- independent reset values for every bit of all registers
- optional interrupt request generation.

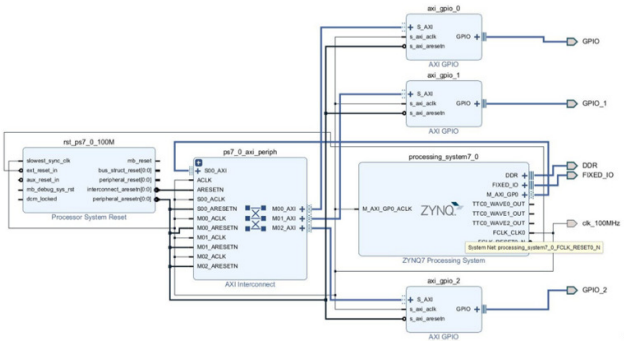

Fig. 3 Architectural View of data transfer using AXI GPIO

Fig 3 shows the architecture view of the data transfer using AXI GPIO. The design flow of the Vivado integrated design environment (IDE) is explained in the below step by step

- Block design creation: in this process block design will be created using Processor System Reset, AXI interconnect and Zynq processing system, etc.
- Simulation: Behavioural and structural logic simulation performed in simulation step.
- RTL analysis: RTL schematic nothing but gate-level schematic, representation of pre-optimized design in the form of adders, multipliers, counters, AND gates, and OR gates, etc.
- Synthesis: It is the process of converting RTL code into the gate-level netlist.
- Implementation: means place and route the netlist on the FPGA resources.
- Bit stream: It's a file contains programming information for FPGA.

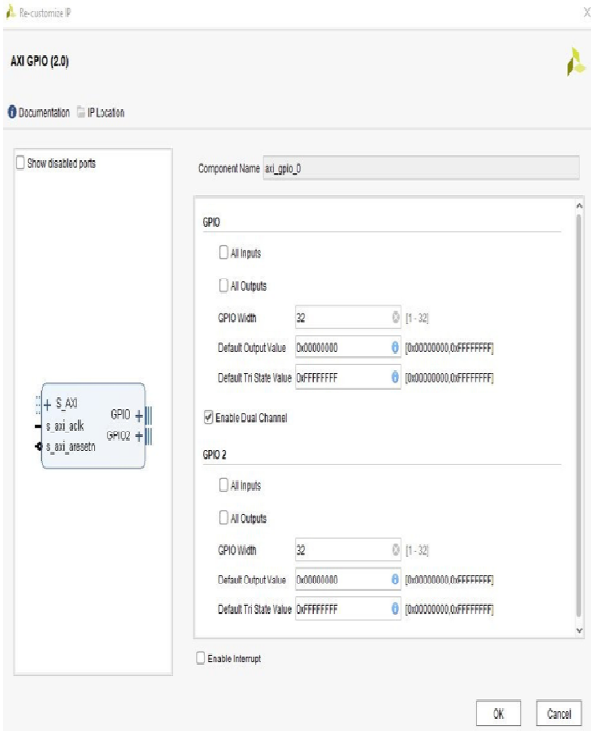

Fig. 4 AXI GPIO customization

- There are separate windows for two channels customization
- All Inputs: with this option, we can GPIO channel bits in INPUT mode.
- All Outputs: with this option, we can set GPIO channel bits in OUTPUT mode.
- GPIO Width: with this option, we can define the width of the GPIO channels. It varies from 1 to 32, and the default value is 32 [3].
- Default Output Value: with this option we can set the default value of all the enabled bits of this channel. By default, the value is set to 0x0.
- Default Tri-State Value: with this option we can configure the input or output mode of each bit of the GPIO channel. default value is 0xFFFFFFFF.
- Enable Dual Channel: with this option, we can enable dual channels in GPIO.
- Enable Interrupt: this option will enable interrupt mode and interrupt registers in the GPIO module, by default interrupts are not enabled.

Fig 5 shows the PL fabric clocks customizations in Vivado software, when we double click the Zynq processing system there is a clock configuration option, there PL fabric clocks are available.

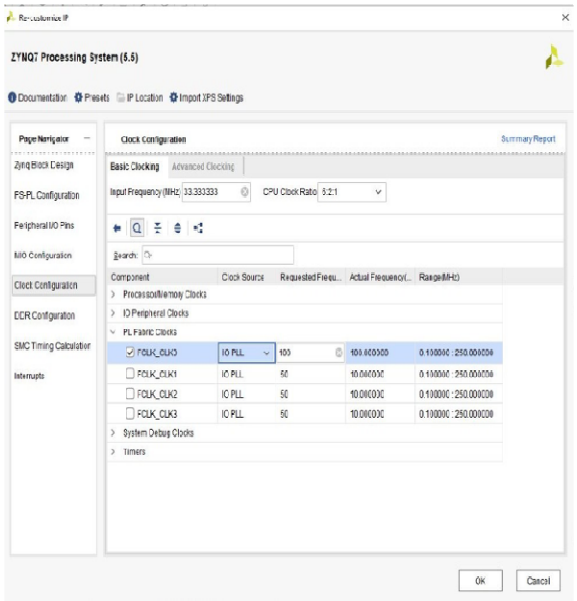

Fig. 5 PL fabric Clocks

- These clocks are generated by the PS and can be used by PL.
- These clocks are implemented by using IO PLL (phase-locked loop).
- The default frequency of the clock is 50 MHz.
- In Fig 1 PL block contains two counters named counter1 and counter2, which required a clock for running, so for that, these fabric clocks will be used.

Fig 6 shows the FIFO generator customization, in Vivado software there are pre-defined FIFO generators are available. Whenever we require, we can customize them.

- It contains WR, RD, FULL, EMPTY signals.
- Write Width: with this option we can customize the write width of the FIFO.
- Write Depth: with this option we can vary the write depth FIFO.
- Read Width: with this option we can vary the read width of the FIFO.
- In Fig 1 PL block contains one FIFO, used for data transfer between PL and PS.

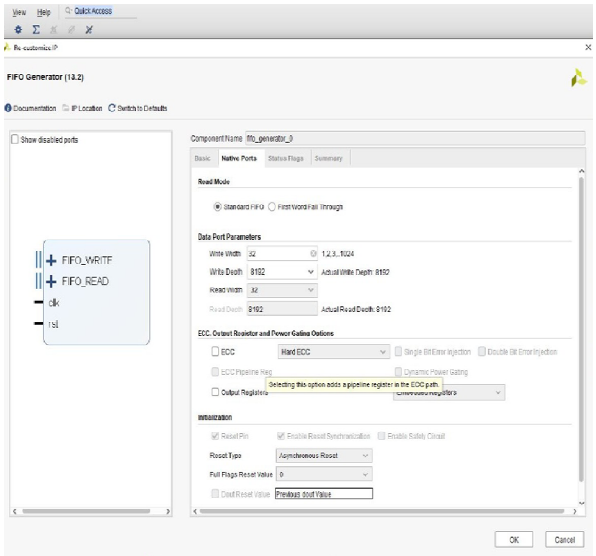

Fig. 6 FIFO Generator

III. DATA TRANSFER USING AXI MASTER IP

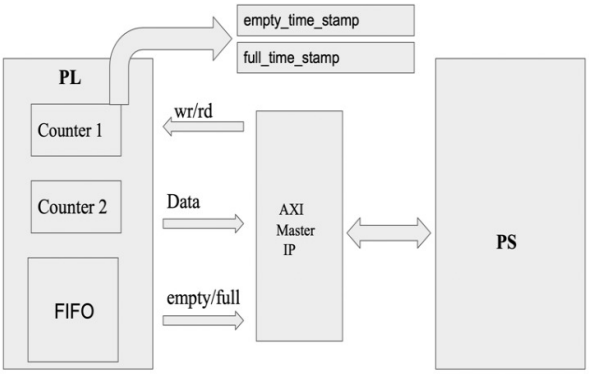

Fig. 7 Block Diagram of the Data transfer using AXI Master IP

Fig 7 shows the Block Diagram of data transfer using AXI Master IP between PL and PS. In AXI GPIO Data transfer method it consumes more CPU time, instead of using three GPIOs in Fig 1 if we replace those GPIOs with AXI master IP, then we can transfer data with high speed with less involvement of the CPU. The components used in the AXI GPIO Data transfer method, same components, and logic used in AXI master IP method. But the main difference is, here AXI Master IP takes care of all the handshaking signals and data requests.

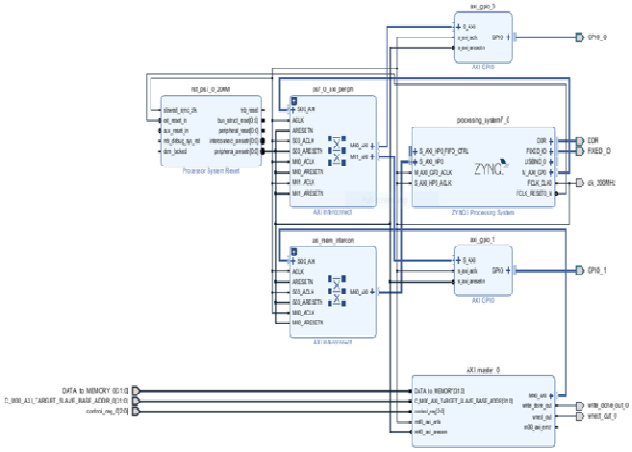

Fig. 8 Architectural View of data transfer using AXI Master IP

Fig 8 shows the architecture view of the data transfer using AXI Master IP. The design flow of the Vivado integrated design environment (IDE) is similar to AXI GPIO method except, AXI Master IP creation. Steps for AXI Master IP creation are shown below.

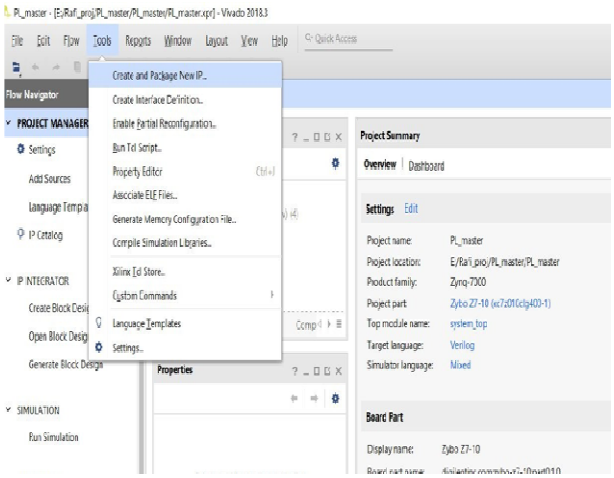

Fig. 9 Step1 for AXI Master IP creation

Fig 9 shows the first step for AXI Master IP creation. In Vivado software, there is option in Tools section, that is (create and package new IP). If you select that option, it will show another window.

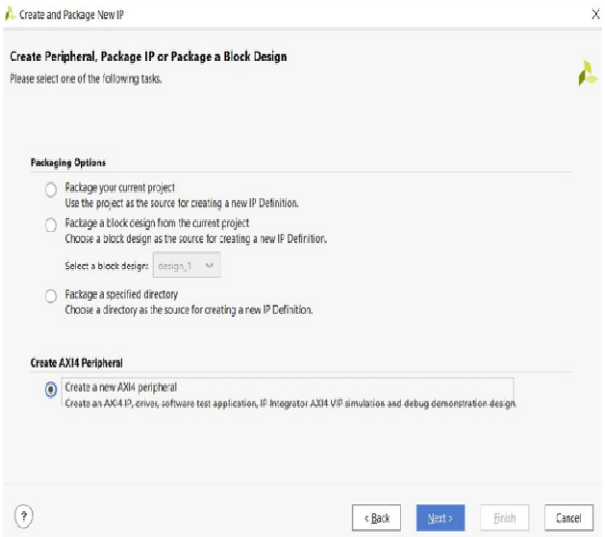

Fig. 10 Step 2 for AXI Master IP creation

Fig 10 shows the second step for AXI Master IP creation. After first step one window will pop up, there we have to select (create a new AXI4 peripheral) option.

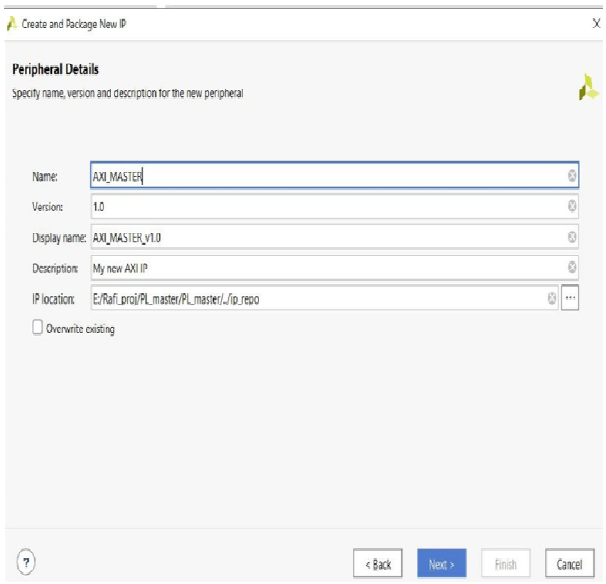

Fig. 11 Step 3 for AXI Master IP creation

Fig 11 shows the third step for AXI Master IP creation. After second step one window will pop up, there we have to give name of the IP, version and description, etc.

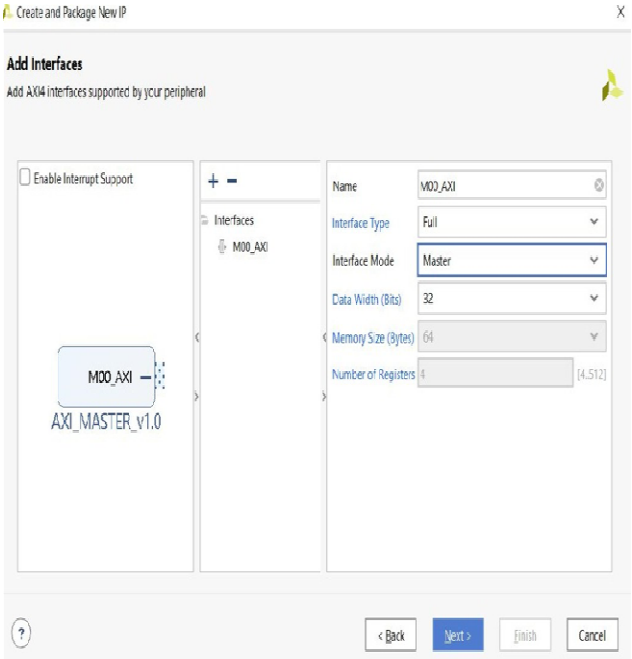

Fig. 12 Step 4 for AXI Master IP creation

Fig 12 shows the fourth step for AXI Master IP creation. In this step we have to select (interface mode) as Master, and (interface type) as Full.

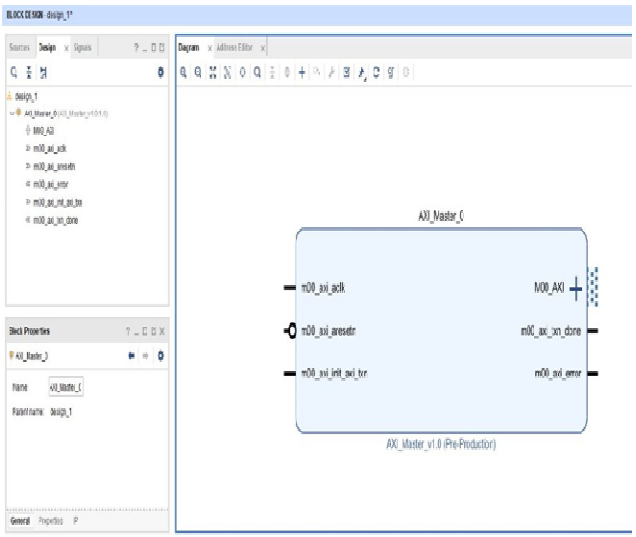

Fig. 13 Step 5 for AXI Master IP creation

Fig 13 shows the fifth step for AXI Master IP creation. In work area AXI Master IP will be generated with necessary signals, like shown in Fig 13.

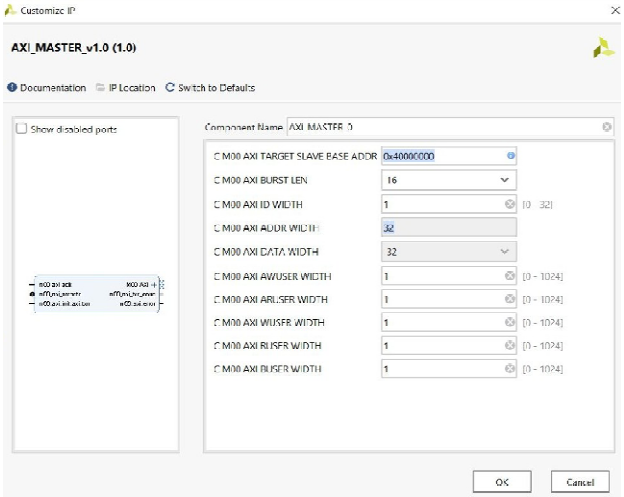

Fig. 14 Step 6 for AXI Master IP creation

Fig 14 shows the sixth step for AXI Master IP creation. Here we can customize AXI master IP, we can give (Target Slave Base Address), and we can modify the parameters of the IP.

IV. WAVEFORMSAND PERFORMANCECALCULATION

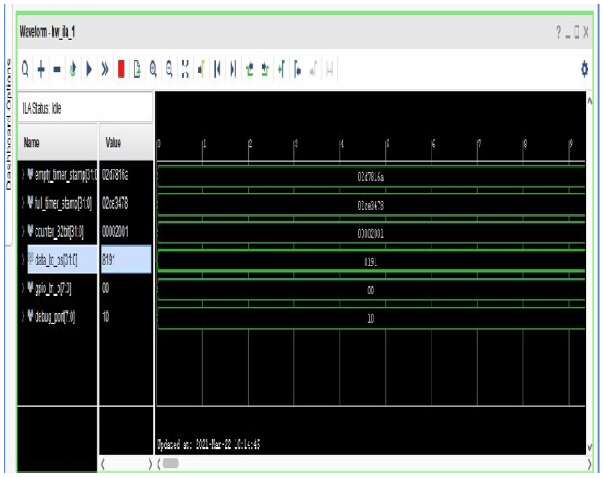

Fig. 15 Waveforms of the AXI GPIO based transfer

Fig 15 shows the waveforms of the data transfer between PL and PS using AXI GPIO. Here Integrated Logic Analyzer (ILA) is used to monitors the internal signals of the design. From Software Development Kit (SDK) all three GPIOs initialization and the data transaction request are made from SDK. With the help of the waveforms, the time required to transfer the data between PL and PS is calculated below.

- T = 1/F formula used here
- Counter running at 100MHZ
- Empty_timer_stamp = 0 x 2D7816A (Hex)
- Full_timer_stamp = 0 x 2CE3478 (Hex)
- Result = Empty_timer_stamp - Full_timer_stamp
  = 60,95,22 (Dec)
- Time Required to transfer 8192 samples data between PL and PS = 60,95,22 x 10 ns
  = 6.09ms (using AXI GPIO)

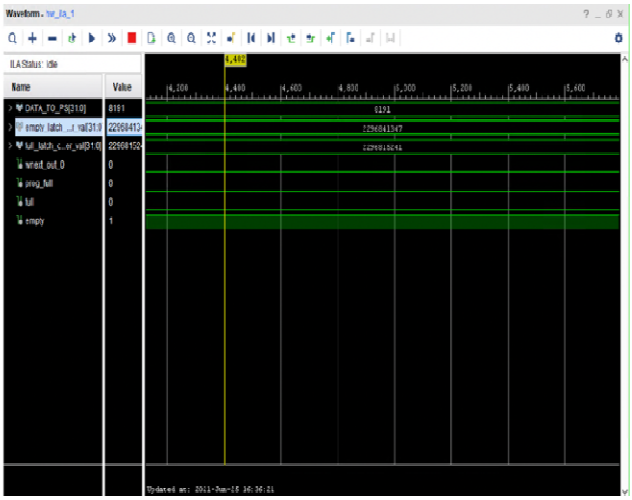

Fig. 16 Waveforms of the AXI Master IP based transfer

Fig 16 shows the waveforms of the data transfer between PL and PS using AXI Master IP. Assigning target base address and initialization of AXI Master IP happened from SDK. Because of Master interface CPU involved at starting only, after that AXI Master IP take care of the data transaction. The time required to transfer the data between PL and PS is calculated below.

- T = 1/F formula used here
- Counter running at 100MHZ
- Empty_timer_stamp = 2296841347
- Full_timer_stamp = 2296841347
- Result = Empty_timer_stamp - Full_timer_stamp
  = 2,61,06 (Dec)

- Time Required to transfer 8192 samples data between PL and PS = 26106 x 10ns
  = 0.26ms (using AXI Master IP)

V. CONCLUSION

The AXI Master IP was implemented on Zynq Board successfully and, it is used as communication logic between PL and PS, for high-speed data transfer purposes. And also implemented the basic data transfer method (using AXI GPIOs).for both methods, the time required, to transfer the 8,192 data samples from PL to PS is calculated. From those results, it proved that AXI Master IP is 24 times faster than AXI GPIO

.

ACKNOWLEDGMENT

I express my sincere thanks to Industry guide Mrs. Madhuri Nallapaneni, Director, Invecas Pvt. Ltd, Guntur, and VEDA IIT guide Mr. Srinivas Rao Pokuri, Assistant Professor, VEDA IIT, Amaravati for helping me to carry out this Project Work.

I convey my deep sense of gratitude to Prof. K.Malakondaiah, Director, VEDA IIT, Amaravati, for providing me with an opportunity for carrying out this Project Work.

REFERENCES

[1] "A review and analysis of communication logic between PL and PS in ZYNQ AP SoC", Sunita Ramagond, Siva Yellampalli, C kanagasabapathi, IEEE Conference, 14 May 2018.
[2] UG1037 - Vivado AXI v4.0 User Guide, Xillinx, July 15 2017.
[3] PG144 - AXI GPIO v2.0 Product Guide, Xillinx, October 5 2016.
[4] PG164 - Processor System Reset Module v5.0 Product Guide, Xillinx, November 18 2015.
[5] PG057 - FIFO Generator v13.2 Product Guide, Xillinx, October 4, 2017.
[6] DS190 - Zynq 7000 SoC Data Sheet v1.11.1, Xillinx, July 2 2018.
[7] Zybo Reference Manual, Digilent.
[8] PG082 - Processing System 7 v5.5, Product Guide, Xillinx, May 10 2017.
[9] PG059 - AXI Interconnect v2.1, Product Guide, Xillinx, December 20 2017.
[10] UG904 - Vivado Design Suite v2020.2, User Guide, Xillinx, February 26 2021.

