ABSTRACT:

IP qualification at the initial phase is the crucial part of SoC development. In the fast phased industry, Quality assurance comes with the penalty of time to market. As part of quality assurance, the newly designed or customized IP will be tested well and then integrated into SoC. This project deals with one of the methodologies that is used in the industry to Qualify the IP.

This project is based on verifying the functionality of a custom IP in an Artix 7 based FPGA platform. XC7A200T is the exact general purpose FPGA device which is used in this project.

First phase of the project is to bring up the eco system which is required to verify the functionality of any custom IP (i.e Custom SPI in this project). Eco system is nothing but the Processor subsystem, Memory Subsystem, Clocking and reset and other basic peripherals like Timer, UART and GPIO.

Second phase of the project will be integrating the Design Under Test with the platform, writing the C based testcase for the integrated Eco system. The development of testcases includes the understanding of different things like system specifications, system use cases, System interfaces and members. Once this understanding is done, we'll start thinking of C based Drivers and testcase development.

The core Ideology of the project is to take a custom SPI IP and Integrate it with the small SoC like ecosystem. Emulate this ecosystem in the FPGA and writing the C based testcases to verify the functionality of the IP. SPI is a communication protocol, which is used to interface with the External non-volatile memory like serial flash. To demonstrate this use case, we are using winbond w25q64 SPI flash, that is interfaced with the Custom IP in fast read mode.

Signature of the Student
Name: Selva Kumar S
Date: 22/3/2024
Place: Bengaluru

Signature of the Supervisor
Name: Siva Selvamani
Date:22/3/2024
Place: Bengaluru

Contents

1. Functional Block Diagram

The functional block diagram of FPGA based emulation environment that supports the IP validation is given below.

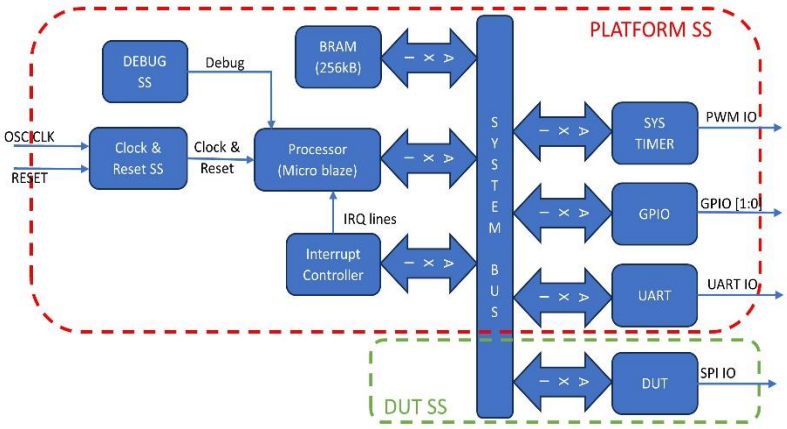


Figure 1Block diagram of the SoC

Entire SoC is divided into two major subsystem which are,

1. Platform Subsystem

   This subsystem consists of all major components of SoC like Processor (Micro blaze), Memory (BRAM), Interrupt Controller, Clock PLL and Reset System, Debug subsystem, System tick timer, GPIO, UART, System Bus (AXI Interconnect). All these components are connected with each other via AXI Interconnect through a 32-bit AXI bus.

2. DUT Subsystem

   This subsystem contains the targeted Custom IP to be qualified in Emulation Platform. In our case, it's a AXI to QSPI IP. This is customized to interface only with Dual SPI at fast read / write mode.

Xilinx IP used in Platform and DUT Subsystem:

Xilinx IP Used	Description	Address map	Size
clk_wiz:6.0\	PLL to generate different clock domain for the SoC	NA	
microblaze:11.0\	Processor system to execute the test code	NA	
mdm:3.2\	Processor debug interface for code JTAG	NA	
proc_sys_reset:5.0\	Processor and SoC reset system	NA	
axi_gpio:2.0\	GPIO module to toggle the Application alive LED	0x40000000	1kB
axi_bram_ctrl:4.1\	AXI BRAM controller to connect RAM to Sys Bus	NA	
blk_mem_gen:8.4\	BRAM block to have array of system memory	0xC0000000	512kB
xlconcat:2.1\	Multiplex to concatenate the Interrupt from different IP	NA	
axi_intc:4.1\	AXI Interrupt Controller to aggregate the IRQs	0x41200000	1kB
axi_uartlite:2.0\	AXI UART IP to display the debug prints in Code	0x40600000	1kB
system_ila:1.1\	Integrated Logic analyzer to probe the RTL signals	NA	
axi_timer:2.0\	AXI Timer to generate the general delay functions	0x41C00000	1kB
lmb_v10:3.0\	Processor Cache memory generator	0x00000000	16kB
axi_quad_spi:3.2\	Targeted Custom SPI block which is to be tested	0x44A00000	4kB

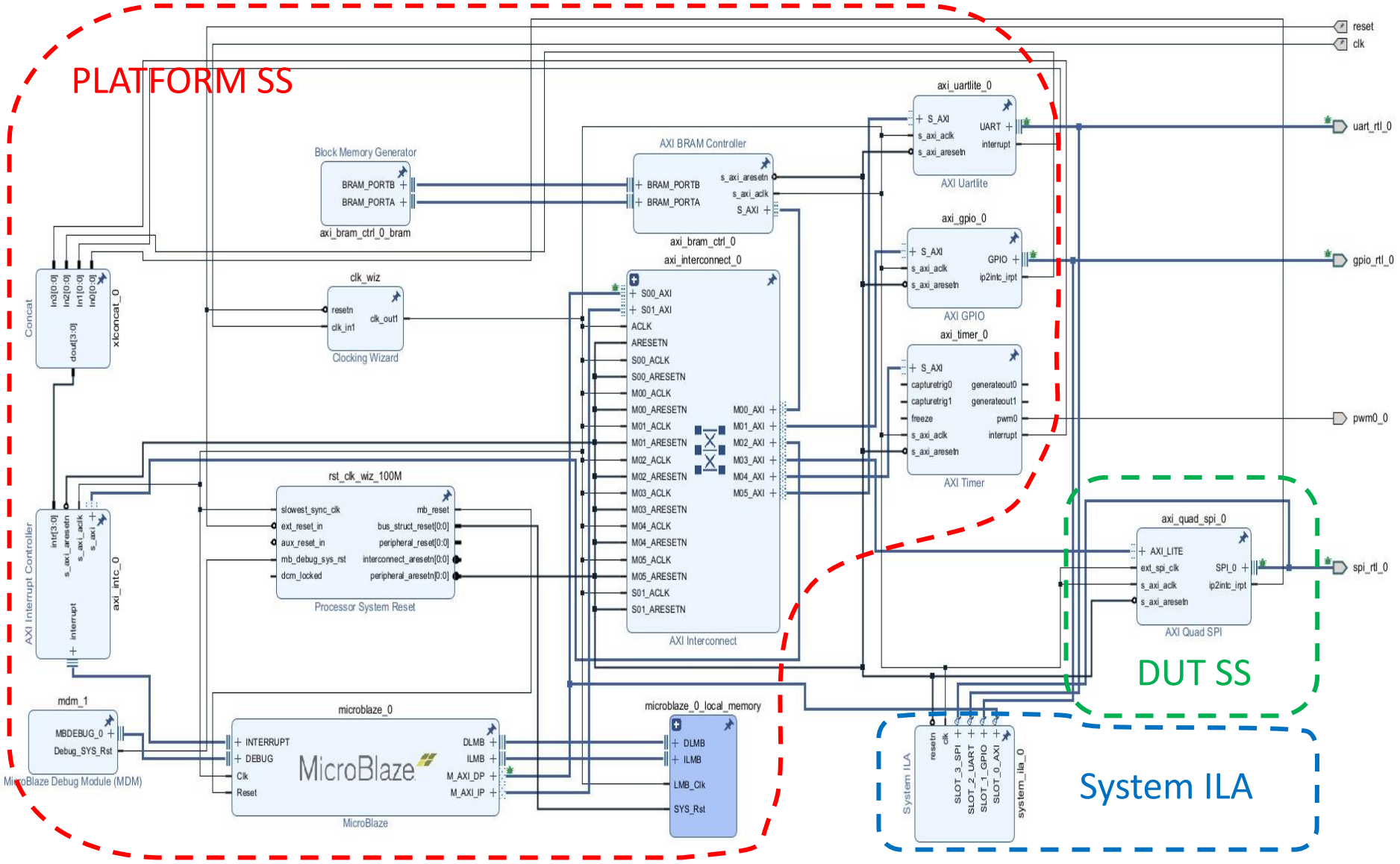


Figure 2Xilinx IP Integrator Block Design

2. Hardware and Software Requirements:

The below table contains the Hardware and Software Requirements for developing an FPGA based emulation environment for Custom SPI IP.

Sl. No	Hardware Required	Software Required
1.	Artix 7 based TE0712-02 SOM	Xilinx Vivado 2023.1
2.	FPGA Carrier Card	Xilinx Vitis 2023.1
3.	W25Q64 SPI Daughter card	SPI Protocol Analyzer
4.	USB to UART FTDI adaptor	
5.	Logic Analyzer	

Table 1Hardware and Software Requirements

Hardware Requirements:

Artix 7 FPGA SOM:

TE0712-02 is a general-purpose FPGA SoM which can be used with custom carrier card. For implementation of this SoC design exactly XC7A200TFFG1156-3 part is used. This FPGA part is specifically suitable for small Microcontroller implementations.

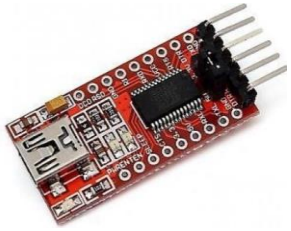
FPGA Carrier Card

Xilinx custom carrier card is used to expose the IOs of this FPGA. This carrier card has all the peripheral interfaces and its respective slave devices.

W25Q64 SPI Daughter card:

W25Q64 is a breakout board which has 64Mbits flash with SPI and DSPI interface. It supports up to 104MHz. It works on 2.7V to 3.5V range. Based on this we need to allocate the IO bank from the FPGA.

USB to UART FTDI adaptor:



FTDI USB port is used for COM port connectivity via which debug prints in the code will be sent out. This FTDI will work in 3.3V Range and it supports up to 230Kbps UART baud rate.

Software Requirements:

Xilinx Vivado 2023.1:

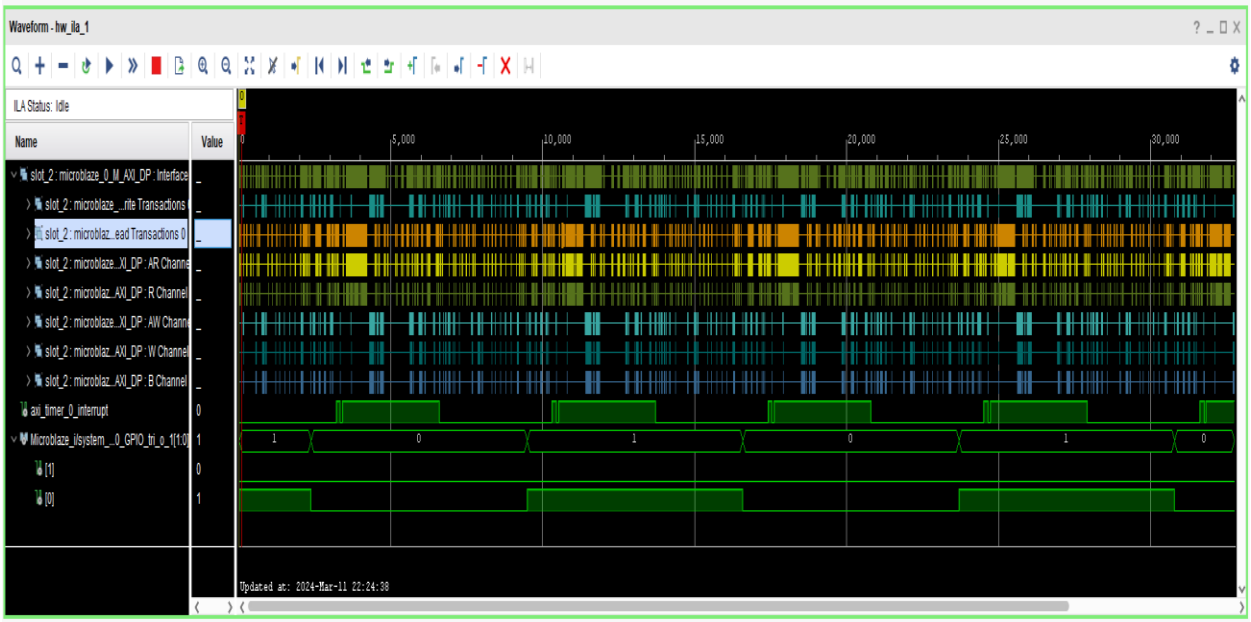
Vivado tool is used to generate the RTL from the Block design of the SoC and handle all the SoC integration part. This tool will also handle the flow of synthesis, place and route, and bit file generation which is need for the target Artix 7 Family FPGA. Block design for this dissertation is shown in the section 2 of the document. This Block design is generated with the help of Xilinx IP integrator flow. This tool has Hardware manager tool which will get the debug probe date via Integrated Logic Analyzer from the real hardware.

Xilinx Vitis 2023.1:

Vitis tool is used to generate the Embedded Application for the SoC design implemented on FPGA and handle the Application compilation and debug part. This tool will also handle the flow of compilation, debug and binary file generation which is need for the target Artix 7 Family FPGA. The linker script required to handle the memory map of the SoC will be generated automatically from hardware specification file (.xsa file). This tool will also provide serial terminal to monitor the debug prints in the application.

3. OUTPUT (Initial Samples)

The below screenshots contain the ILA capture of Initial Platform where UART and System Timer was brought up. This Timer is used as general-purpose timer which generates microsecond Delays required in the test Application. UART is used for debug prints written on C code in the Tera Term Application.



4. Design Considerations:

- UART COM port is working in 230KBps.
- Target DUT is Operating at 10MHz-50MHz.
- Timer is configured to generate interrupt on every 30 micro second.
- User LED will be toggled on every 500 ms to present the application alive status.
- System BUS clock frequency is 100 MHz
- BRAM size is 512 kB.
- Nonvolatile memory controller (flash controller) is not part of platform design.
- Entire Application code is stored and executed from BRAM.
- FPGA is having a separate config flash to store the BIT file.

5. Future Plans

Phases	Start Date – End Date	Work to be Done	Status
Dissertation Outline	11 January 2024 – 22 January 2024	Design Review and Preparation for Dissertation Outline	Completed
Understanding of concepts	23 January 2024 – 20 February 2024	Understanding the IP features, FPGA and Emulation concepts. Analyzing suitable Hardware for required application.	Completed
Design and development	21 February 2024 – 31 March 2024	Implementing Platform architecture that supports the IP Qualification.	Completed
		Integrating the targeted DUT and writing the final test applications based on the features to be tested	InProgress
Testing	01 April 2024 – 10 April 2024	Testing, Evaluation & Conclusion	Pending
Dissertation Review	11 April 2024 – 22 April 2024	Submitting the Dissertation work to Supervisor for review and feedback	Pending
Submission	23 April 2024	Final submission of Dissertation	Pending

Table 2Future plans and tasks

6. Abbreviations:

FPGA	Field Programable Gate Array
SPI	Serial Peripheral Interface
IP	Intellectual Property
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
GPIO	General Purpose Input and Output Pin
SoC	System On Chip
DUT	Design Under test
BRAM	Block RAM (Random Access Memory)

Table 3Abbreviations