

# ELBERT S7 HANDBOOK

# An Educational Guide to FPGA Design and Development

## CONFIDENTIAL and PROPRIETARY

This document contains confidential and proprietary information belonging exclusively to <b>Numato</b>	N
Systems Pvt Ltd. No confidential or proprietary	
information contained in this publication may be	RE
reproduced, stored in a retrieval system, or transmitted	Doc
in any form or by any means electronic, mechanical,	Doc
photocopied, recorded, or otherwise without prior	NS
written permission of Numato Systems Pvt Ltd.	
Copyright© 2025, Numato Lab.	

Numato System	ns Pvt Ltd
REV: V1.0	
Doc Name: ELBERT S7 HANDE	BOOK
Doc No:	Page
NSQMS25FP06B07-HB	Page 1 of 1

# Table of Contents

SECTION I	
1. INTRO	DUCTION
2. BOARI	O FEATURES
2.1. FPG.	A Device
2.2. Conf	guration and Memory6
-	nunication Interface7
2.4. User	I/O7
2.5. Perip	heral Interfaces7
0	CATION
4. WIRIN	G DIAGRAM
5. USB In	terfacing and Programming Options10
6. JTAG (	Connector
	_B and Reset Buttons 11
8. Power	Supply
9. USER I	14
9.1. LEDs	
9.2. DIP S	Switches14
9.3. PMO	D Headers 15
10. PER	IPHERAL INTERFACES
10.1. Seve	en Segment Display 17
10.2. HDI	MI TRANSMITTER18
10.3. Mic	ro SD Card19
10.4. Giga	abit Ethernet20
10.5. Aud	io Jack 21
11. Gene	erating Bitstream Using Vivado
12. Prog	ramming Elbert S7 Using JTAG
13. Prog	ramming Elbert S7 Using USB-JTAG25
14. Prog	ramming QSPI Flash using Vivado26

SECTION II	28
TRAIN ON THE BOARD	28
Getting Started with Vivado: Creating a New FPGA Project	
1. UART Communication – Printing "Hello World"	31
2. Controlling Onboard Peripherals (LEDs, Seven Segment Display, and PM) with Slide Switches	
3. HDMI OUTPUT EXAMPLE DESIGN	
4. DDR3 Mem test on ELBERT S7	58
5. Gigabit Ethernet Example Design	72
6. SD card test	
REFERENCES	105

# **Numato Lab**<sup>®</sup>

# SECTION I 1. INTRODUCTION



#### **Unleash Your Creativity with Compact Power**

The **Elbert S**<sub>7</sub> is a compact and versatile FPGA development board designed to support a wide range of educational and practical digital design applications. Built around the **Spartan-**7 **FPGA (XC7S50-1CSG324C)** from AMD, this board offers a reliable platform for both **beginners** and **experienced developers** looking to explore the world of FPGAs.

Whether you're a student learning digital logic or a developer building custom hardware solutions, the Elbert S7 makes FPGA development more accessible and engaging. Its thoughtful design supports hands-on experimentation, helping users understand fundamental concepts while also enabling the development of more advanced projects.

The board is an ideal choice for anyone interested in **digital design**, **embedded systems**, **signal processing**, **or rapid prototyping**. With strong support for industry-standard tools like **Vivado** and **Vitis**, and compatibility with both **Verilog** and **VHDL**, the Elbert S7 provides a smooth and scalable learning path for anyone stepping into the field of FPGA design.

# **Numato Lab**<sup>®</sup>

## 2. BOARD FEATURES

The Elbert S7 FPGA development board offers a rich set of features, making it a powerful tool for learning and developing digital systems. Its well-balanced combination of core components and peripherals allows users to implement a wide variety of applications—from basic logic design to advanced embedded systems.

#### 2.1. FPGA Device

- Model: AMD Spartan-7 (XC7S50)
- Package: CSGA324
- Speed Grade: -1

This low-power, high-performance FPGA offers enough logic resources and I/O options for both educational and practical digital design tasks.

#### 2.2. Configuration and Memory

- DDR3 SDRAM: MT41J128M16JT-125: KTR, A high-speed 2 Gb (128M x 16)
- DDR3 memory chip used for applications requiring larger memory capacity, such as video processing, buffering, or running soft processors.
- **Flash Memory:** 128 Mb Quad-SPI (MT25QU128ABA1ESE-oSIT TR) Used to store FPGA configuration bitstreams and other application data.
- **Clock Source:** 100 MHz CMOS oscillator Serves as the main clock input to drive system timing.
- Configuration Methods:
  - JTAG (standard FPGA programming method)
  - USB (via onboard FTDI chip; supported on Both Windows and Linux systems)

#### 2.3. Communication Interface

• **FTDI FT2232H**: Dual-channel USB-to-serial/FIFO interface Enables high-speed communication between the FPGA and a host PC. Also supports USB-based FPGA programming and debugging.

#### 2.4. User I/O

#### • 8 DIP Switches & 8 LEDs:

Useful for creating simple input/output projects and learning digital logic control.

#### • Four PMOD Headers:

Standard 2x6 connectors for attaching external modules and user-defined peripherals.

#### 2.5. Peripheral Interfaces

#### • SD Card Slot:

Provides external storage, useful for embedded applications requiring file access.

#### HDMI Transmitter:

Enables the board to send video signals to external displays, suitable for image processing or display generation projects.

#### • Gigabit Ethernet Port:

Supports high-speed networking, ideal for IoT, remote monitoring, or network-based applications.

#### • Audio Jack:

Allows audio input/output for sound-related projects.

#### • Seven-Segment Display:

Displays numeric or limited alphanumeric information, ideal for counter or output visualization tasks.

# **3. APPLICATION**

The Elbert S7 is a versatile development board suitable for a wide range of educational and practical applications:

- Educational Use: Ideal for teaching digital design and embedded systems in schools and universities.
- **Prototype Development**: Enables rapid hardware prototyping and testing of new product ideas.
- Accelerated Computing: Supports hardware-based acceleration of compute-intensive tasks.
- **Custom Processor Design**: Allows development and testing of soft-core or custom embedded processors.
- **Signal Processing**: Suitable for implementing and testing real-time digital signal processing applications.
- **Communication Systems**: Enables the design and evaluation of communication protocols and devices.
  - Video Processing: Supports projects involving HDMI output and image/video processing tasks.

# 4. WIRING DIAGRAM



# 5. USB Interfacing and Programming Options

The Elbert S7 board features a high-speed USB interface powered by the FTDI FT2232H chip, enabling seamless communication with Windows, Linux, or macOS computers. A standard USB Type-A to Type-C cable is used to connect the board to a host system. The USB connection also supplies power to the board by



default, so it's important to avoid connecting it to overloaded or unpowered USB hubs.

To provide flexibility in programming, the board includes a programming mode selection mechanism. A multiplexer (MUX) is used internally to switch between two configuration sources:

- JTAG (via external programmer)
- USB-JTAG (via onboard FT2232H chip)

The FT2232H's Channel A is configured specifically for USB-based FPGA programming. A PGM SEL switch is provided on the board, allowing users to select the desired programming method:

- Set the switch to USB to use onboard USB-JTAG.
- Set the switch to JTAG to program via an external JTAG programmer.

This flexible configuration makes the Elbert S7 suitable for both beginners using USB for simplicity and advanced users who prefer direct JTAG access.



#### 6. JTAG Connector



The Elbert S7 board includes a standard JTAG header that provides access to the FPGA's internal JTAG registers. This interface supports programming and debugging using tools such as the Xilinx Platform Cable USB. Users can connect a compatible JTAG cable to this header to perform direct configuration, in-system debugging, or low-level hardware testing. This option is especially useful for advanced users requiring greater control and visibility into the FPGA during development.



Elbert S7 features a Push-button **PROG B** normally meant to be used as a "PROG B" signal for configuration reset. Push-button PROG\_B is connected to FPGA enabling manual configuration reset, push-button **PROG B** is pin **R8.** For connected to GND. The user can reconfigure the FPGA manually, by PROG B pressing this push-button **PROG\_B**.



"PROG B" controls the configuration logic. When the PROG B pin is de-asserted, resets the FPGA and initializes the new configuration.

Elbert S7 features a Push-button **RESET** normally meant to be used as "Reset" signal for designs running on FPGA. Push-button RESET is connected to FPGA

pin **T14.** Push-button **RESET** is **active-high**. This push button can also be used for any other input and is not just limited to be used as a Reset signal.



# 8. Power Supply



The **Elbert S**<sub>7</sub> board features a dual power input architecture, allowing users to power the board either through a USB Type-C port or a **5V** DC jack. This flexible design is ideal for both development and deployment environments, where users may prefer different power sources based on convenience or availability.

Each input is routed through a Schottky diode (**D2** for USB Type-C and **D4** for the DC jack), which plays a crucial role in protecting the board. These diodes prevent reverse current flow, ensuring that power does not back feed into the sources, and also helps in automatic source selection by allowing the input with the higher voltage to take priority.

The outputs of these diodes feed into a power selector stage that determines which source will power the board. This intelligent selection ensures that only one source supplies power at a time, eliminating any risk of conflict or damage.

Once selected, the power signal goes through a filter and protection block. This section typically consists of inductors and capacitors to filter out voltage ripples and electromagnetic noise, ensuring clean and stable voltage. Additionally, it protects the board against sudden spikes or other electrical disturbances.

A user-accessible **PWR SW** (Sw11) is provided to manually control the power delivery to the board. This switch adds convenience by allowing users to turn the board ON or OFF without needing to disconnect the power cable physically.

Finally, an onboard Power LED (**PWR**) connected after the switch provides a visual indication when the board is powered.

The stable output voltage is labelled as VIN, which is then distributed to various subsystems and regulators on the board.

**Note:** Only a regulated 5V DC power supply should be used to power the board through either the USB Type-C port or the DC jack. Supplying a voltage higher than 5V may damage the internal circuitry.





# 9. USER I/O

#### 9.1. LEDs

The board features 8 user-controllable LEDs. These LEDs serve as visual indicators and are commonly used for debugging or representing binary output data. Each LED is connected to a dedicated FPGA pin and can be driven directly using logic outputs.

PIN NAME	FPGA PIN	IO STANDARD	
LED o	V14	LVCOMS33	
LED 1	V15	LVCOMS33	
LED 2	U12	LVCOMS33	
LED 3	V13	LVCOMS33	
LED 4	T12	LVCOMS33	
LED 5	T13	LVCOMS33	
LED 6	R11	LVCOMS33	
LED 7	T11	LVCOMS33	
	LED 0 LED 1 LED 2 LED 3 LED 4 LED 5 LED 6	LED 0       V14         LED 1       V15         LED 2       U12         LED 3       V13         LED 4       T12         LED 5       T13         LED 6       R11	LED 0V14LVCOMS33LED 1V15LVCOMS33LED 2U12LVCOMS33LED 3V13LVCOMS33LED 4T12LVCOMS33LED 5T13LVCOMS33LED 6R11LVCOMS33

Table 1: LED Pin Mapping

#### 9.2. DIP Switches

There are 8 onboard DIP switches that can be used to input binary data or control signals into the FPGA. These inputs are ideal for controlling the flow of a design, setting modes, or triggering actions within a design.

PIN NAME	FPGA PIN	IO STANDARD
SW o	C4	LVCOMS18
SW 1	B4	LVCOMS18
SW 2	C3	LVCOMS18
SW 3	B3	LVCOMS18
SW 4	A5	LVCOMS18
SW 5	A4	LVCOMS18
SW 6	A3	LVCOMS18
<b>SW</b> 7	A2	LVCOMS18

Table 2: DIP Switch Pin Mapping

#### 9.3. PMOD Headers

PMOD (Peripheral Module) connectors are a widely adopted standard for connecting peripheral devices to FPGA and microcontroller development boards. The Elbert S7 development board provides **four 2×6 PMOD headers** (P3, P4, P5, P6), allowing users to interface a variety of external modules including sensors, displays, communication interfaces, and custom digital circuits.

Each PMOD header offers **eight general-purpose I/O (GPIO) signals** plus **two dedicated power pins (3.3V and GND)**, organized in a dual-row format. The I/O lines from each PMOD are connected directly to the FPGA, enabling flexible software-defined control and configuration.



PIN NAME		IO STANDARD			
	CONNo(P1)	CONN1(P2)	CONN2(P3)	CONN <sub>3</sub> (P4)	IO STANDARD
Do	M18	H18	A10	C17	LVCMOS33
D1	P14	G16	C10	D18	LVCMOS <sub>33</sub>
D2	<b>P</b> 17	H16	C12	D16	LVCMOS <sub>33</sub>
D3	R18	K14	B11	E16	LVCMOS <sub>33</sub>
D4	N18	G18	A9	B18	LVCMOS <sub>33</sub>
D5	P15	G17	C9	C18	LVCMOS <sub>33</sub>
D6	P18	H17	C11	D17	LVCMOS33
<b>D</b> 7	T18	J15	A11	E17	LVCMOS33

Table 2: PMOD Pin Mapping

# **Numato** Lab<sup>®</sup>

G

C

Е

# 10. PERIPHERAL INTERFACES

#### 10.1. Seven Segment Display

A seven-segment display is a simple output device commonly used for displaying decimal numbers or characters. The **Elbert S7** includes a 4-digit seven segment display, which is useful for displaying counters, timers, or debugging information.

Each digit is made up of 7 individual segments (labelled A–G) and an optional decimal point (dp). The display works by quickly switching between digits using control signals-a technique known as multiplexing.

You can control which digit is shown by activating the digit's enable line and then lighting up the correct segments to form a number or letter.

**Note:** All signals (*a*, *b*, *c*, *d*, *e*, *f*, *g*, *dot*, *enable 1*, *enable 2*, *enable 3*, *enable 4*) used for controlling 7-Segment display are **active-low** signals. So, for example, for displaying "8" in display-2, users need to drive *Enable 2* to *o* as well as drive signals *a*, *b*, *c*, *d*, *e*, *f* to *o*. All other signals need to be driven to 1.



#### 10.2. HDMI TRANSMITTER

The HDMI (High-Definition Multimedia Interface) Transmitter on the Elbert S7 board allows the FPGA to send digital video and audio signals to an external display such as a monitor or TV.

HDMI uses a high-speed signalling technology known as **TMDS (Transition-Minimized Differential Signalling)**. This ensures reliable transmission of data over HDMI cables by reducing electromagnetic interference and ensuring signal integrity.

Inside the FPGA, video signals are generated in digital format—usually in RGB (Red, Green, Blue) colour space—along with synchronization signals such as horizontal sync, vertical sync, and data enable. These signals are then encoded and sent through the HDMI transmitter chip to the display device.

The HDMI interface typically includes the following signals:

- TMDS Data Channels (3 pairs): Carry video/audio/control data.
- TMDS Clock: Synchronizes the data.
- **DDC (I2C lines)**: Used to read the display's capabilities (EDID)
- **HPD**: Lets the FPGA know a display is connected.
- **CEC**: Optional control communication between HDMI devices

		1					J4
path	R15	HDMI_TX2_P	4		35	J4_D2_P	
	т15	HDMI_TX2_N	6		33	J4_D2_N	
	U16	HDMI_TX1_P	7		32	J4_D1_P	
	V17	HDMI_TX1_N	9		30	J4_D1_N	
SPARTAN 7	U17	HDMI_TX0_P	10		29	J4_D0_P	
XC7S50-1CSGA324C	U18	HDMI_TX0_N	12	HDMI TX BUFFER	27	J4_D0_N	HDMI
	<b>B1</b> 4	HDMI_TX_CLK_P		TPD125520DBTR		J4_CLK_P	CONNECTOR
	R16 R17	HDMI_TX_CLK_N	13 15		26 24		
			1				
	U15	HDMI_TX_CEC	16		23	J4_CEC	
	V16	HDMI_TX_SCL	17		22	J4_SCL	
	P13	HDMI_TX_SDA	18		21	J4_SDA	
	R13	HDMI_TX_HOT	19		20	J4_HPD	

#### 10.3. Micro SD Card

The Elbert S7 board features a **microSD card slot**, which allows users to interface with removable flash memory for data storage and retrieval. This is especially useful for embedded applications where storing files, logs, configuration settings, or multimedia content is required.

The SD card slot on the board is wired to the FPGA through an **SPI (Serial Peripheral Interface)**. SPI is a simple and widely used protocol for communicating with memory devices like SD cards. Although SD cards also support a more complex SD bus mode, SPI is preferred in FPGA designs due to its simplicity and ease of implementation.

The typical SPI signals used are:

- **MOSI (Master Out Slave In)** Data sent from the FPGA to the SD card.
- **MISO (Master In Slave Out)** Data received from the SD card to the FPGA.
- SCLK (Serial Clock) Clock signal generated by the FPGA to control communication timing.
- **CS (Chip Select)** Used to select and activate the SD card.



#### 10.4. Gigabit Ethernet

Elbert S7 Development Board features KSZ9031RNX, a highly integrated Ethernet transceiver from Microchip that comply with 10BASE-T, 100BASE-TX, and 1000Base-T IEEE 802.3 standards. It supports communication with the Ethernet MAC layer via standard RGMII interface. KSZ9031RNX implements auto-negotiation to automatically determine the best possible speed and mode of operation. It contains a high-performance 10/100/1000T transceiver and the RGMII interface supports 1000Mbps (1Gbps) operation.

P7       ETH_TXD0       19       RJ4         C7       ETH_TXD1       20       11         B7       ETH_TXD2       21       10         SPARTAN 7       D6       ETH_TX_CTL       25       ETHENET PHY       7         C7550-1CSGA324C       F4       ETH_RXD0       32       6       GPHY_TXRX_C1_N         A8       ETH_RXD0       32       6       GPHY_TXRX_C1_P         C5       ETH_RXD1       31       5       GPHY_TXRX_B1_N         C5       ETH_RXD2       28       GPHY_TXRX_A1_P         ETH_RXD3       27       2       GPHY_TXRX_A1_P         ETH_RXD3       27       2       GPHY_TXRX_A1_P         ETH_RXD3       27       2       GPHY_TXRX_A1_P         ETH_RXD3       27       2       GPHY_TXRX_A1_P         ETH_RXD4       35       17       LED_ACT         GPHY_TXRX_A1_P       ETH_MDC       36       15       LED_LINK         GFH_HMD1O       37       CT       J       J         GFH_HRS5       42       J       J       J
--

#### 10.5. Audio Jack

The Elbert S7 FPGA board is equipped with a 3.5mm stereo audio jack, enabling audio output to external speakers or headphones. This output is managed by a dedicated digital-to-analog converter (DAC) — the CS4345-CZZ, a high-performance stereo DAC from Cirrus Logic. The inclusion of this audio interface makes the board well-suited for multimedia, audio signal processing, and embedded audio playback applications.

#### **Key Features**

- **Stereo Output**: Supports two audio channels (Left and Right) for full stereo playback.
- **High-Quality DAC**: The CS4345 provides low distortion and high dynamic range, resulting in clear and high-fidelity sound.
- Digital Audio Interface: The DAC accepts audio data in I<sup>2</sup>S (Inter-IC Sound) format, a standard serial protocol for transmitting PCM audio between digital audio devices.
- **16/24-bit Audio Support**: Capable of processing CD-quality and high-resolution audio streams.
- **Headphone Compatible**: The output is designed to drive line-level audio, which can be connected to headphones or powered speakers.



# 11. Generating Bitstream Using Vivado

The bitstream can be generated for Elbert S7 in Vivado by following the steps below:

**Step 1:** It is recommended to generate .bin bitstream file along with .bit bitstream file. Click "Bitstream Settings".



**Step 2:** Select "-bin\_file\*" option in the dialog window and Click OK.

Q.	Bitstream		
Project Settings	Specify various settings related to writing Bitstream		
General Simulation	(i) Note: Additional bitstream settings will be availab	le once you open an implemented design.	
Elaboration	VWrite Bitstream (write_bitstream)		
Synthesis	tcl.pre		
Implementation	tcl.post		
Bitstream	-raw_bitfile		
> IP	-mask_file		
Tool Settings	-no_binary_bitfile		
Project	-bin_file		
IP Defaults	-readback_file		
> Vivado Store	-logic_location_file		
Source File	-verbose		
Display	More Options		
Help			
Text Editor     3rd Party Simulators     Colors     Selection Rules     Shortcuts     Strategies	-bin_file Write a binary bit file without header ( bin).		
> Window Behavior			

Step 3: Finally click "Generate Bitstream".

~	PROGRAM AND DEBUG	_
	👫 Generate Bitstream	
	✓ Open Hardware Man.	ager
	Open Target	
	Program Device	
	Add Configuration	n Memory Device

# 12. Programming Elbert S7 Using JTAG

Set Switch **PGM\_SEL** to **JTAG** for JTAG programming.

Elbert S7 FPGA features an onboard JTAG connector which facilitates easy reprogramming of SRAM and onboard SPI flash through JTAG programmer like "AMD Platform cable USB". Following steps illustrate how to program FPGA on Elbert using JTAG.

**Step 1:** By using JTAG cable, connect AMD platform cable USB to Elbert S7 and power it up.

**Step 2:** Open Vivado project and open the target by clicking on the "Open Target" in "Open Hardware Manager" in the "Program and Debug" section of the Flow Navigator window. Select "Auto Connect".



**Step 3:** If the device is detected successfully, then select "Program Device" after right clicking on the target device "XC7S50\_0" as shown below.



**Step 4:** In the dialog window which opens, Vivado automatically chooses correct bitstream file if the design was synthesized, implemented and bitstream generated

successfully. If needed, browse to the bitstream which needs to be programmed to FPGA. Finally, click "Program".

Program Device		×
	gramming file and download it to your hardware device. You can optiona file that corresponds to the debug cores contained in the bitstream	
Bitstre <u>a</u> m file: Debu <u>q</u> probes file: I <u>E</u> nable end of s	C:/projects/Elbert_S7/Demo.bit	S
?	<u>P</u> rogram C	Cancel

As soon as "Program" is clicked, a green coloured DONE LED (DONE) on Elbert S7 should light up, indicating that programming process is going on. This LED will turn off when the configuration is complete.

# 13. Programming Elbert S7 Using USB-JTAG

Ensure that the <u>D2XX drivers</u> are installed prior to programming. The channel A of FTDI FT2232H chip on Elbert S7 board is connected to the JTAG interface of the FPGA. Through this connection, USB interface can be used as a JTAG programmer, eliminating the need for a dedicated JTAG cable or connector. Following steps illustrate how to program FPGA on Elbert S7 using USB.

1. Ensure that Switch **PGM\_SEL** is set to **USB** and Connect the USB Type-C cable to the FPGA board.

2. Click on "Auto connect" under hardware manager and it will automatically establish the connection.



# 14. Programming QSPI Flash using Vivado.

A .bin or .mcs file is required for programming Elbert S7 onboard QSPI flash.

**Step 1:** Open Vivado project and open the target by clicking on the "Open Target" in "Open Hardware Manager" in the "Program and Debug" section of the Flow Navigator window. Select "Auto Connect".



**Step 2:** If the device is detected successfully, then select "Add Configuration Memory Device" after right clicking on the target device "xc7s50\_0" as shown below.



Step 3: Select the memory device "mt25ql128-spi-x1\_x2\_x4", then click OK.

Add Configuration	on Memory Device					
Choose a cor	nfiguration memory	part.				
)evice: 🛑 xc7a5(	0t_0					
r						
<u>M</u> anufacturer	Micron	~		Туре	spi	~
Density ( <u>M</u> b)	128	~		Width	All	~
			<u>R</u> eset All Filters			
ect Configuration	Memory Part					
	Memory Part					
	Memory Part	Part	Manufacturer	Alias		
Search: Q-		Part mt25ql128	Manufacturer Micron		3v-spi-x1_x2_x4	
	pi-x1_x2_x4			n25q128-3.	3v-spi-x1_x2_x4 8v-spi-x1_x2_x4	
Search: Q- Name IVame mt25ql128-sp	pi-x1_x2_x4	mt25ql128	Micron	n25q128-3.		

Step 4: After completion of Step 3 the following dialog box will open. Click OK.



**Step 5:** Browse to the working .bin file or the .mcs file (whichever applicable) and click OK to program as shown below. If programming is successful, a confirmation message will be displayed.

	Program Configura	ation Me	emory Device					×	
	Select a configuration	n file ar	id set programn	ning options	i.			2	
			5ql128-spi-x1_x						
	Configuration file:	C:/proj	ects/Elbert_S//s	sample.bin					
	PR <u>M</u> file:							•••	R
	State of non-config	i mem l	/O pins: Pull-	none 🗸					
• •	Program Operatio	ns							
	Address Range	e:	Configuration	File Only		~			
	✓ Erase								
	Blank Chec	k							
	✓ P <u>r</u> ogram								
	✓ Verify								
	Verify <u>C</u> hec	ksum							
	SVF Options								
	Create <u>S</u> VF	Only (n	o program oper	ations)					
	SVF File:							•••	
	?				Ok	C C	Cancel	Apply	

# SECTION II train on the board

## Getting Started with Vivado: Creating a New FPGA Project

Before diving into the peripheral interface projects, it is essential to understand the basic procedure for setting up a new project in Vivado tailored for the Elbert S7 FPGA board. This section will guide you through the initial steps, including downloading and configuring the Board Support Package (BSP).

To begin, download the **Elbert S7 BSP** from our official <u>GitHub</u> repository and place it in the appropriate board files directory (follow the readme file in GitHub repo) on your computer. This allows Vivado to recognize the Elbert S7 board during project creation, simplifying IP integration and pin assignments.

The procedures described in this part will remain consistent across all peripheral interface projects throughout **Section II**. By following this workflow, you'll ensure that your development environment is correctly set up, enabling a smooth and efficient design experience.

#### **Prerequisites:**

#### Hardware:

- Elbert S7 FPGA Development Board.
- Xilinx Platform Cable USB II JTAG debugger. (optional)
- USB A to USB Type C cable.
- 5V DC power suppy.

#### Software:

- Vivado Design Suite with Vitis installed (2024.1)
- Serial terminal application (PuTTY, Tera Term, etc.)

#### Basic procedures to create new project in Vivado.

E

#### Step 1:

Download and install the Vivado Board Support Package files for Elbert S7 from <u>here</u>. Follow the README.md file on how to install Vivado board support files for Numato Lab boards.

#### Step 2:

Open the AMD Vivado Design suite, go to "File -> Project -> New" to create a new project. The "New project" window will pop up. Click "Next".

<u>F</u> ile	F <u>l</u> ow <u>T</u> ools <u>W</u>	/indow	Help Q- Quick Act
	Project		<u>N</u> ew
	Checkpoint	+	Open
	<u>C</u> onstraints	+	Open <u>R</u> ecent ►
	Simulation Waveform	, L	Open Ex <u>a</u> mple
	<u>I</u> P	+	
	I <u>m</u> port	∍ar	t
	Exit		

#### Step 3:

In the "Project Name" window, enter a name for the project and save it at a suitable location. Select the option "Create project subdirectory" to keep all the project files in a single folder.



Now you will see the "Project Type" page as shown below. Select the "RTL Project" and select the option "Do not specify sources at this time". Click "Next".

Nev	w Project	×
	ect Type If the type of project to create.	٦
	RTL Project You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.	
?	<back ca<="" einish="" td=""><td>incel</td></back>	incel

**Step 4:** In the "Default Part" window, select the "Boards" tab. Choose the Vendor as "numato.com", filter the Name "Elbert\_S7" and select the board as shown below. Click "Next" to continue. If Elbert S7 is not displayed in the boards list, make sure that the board support files are installed correctly.

endor: numato.co											
	om	`	*	Name: Elbe	rt_S7			~	Board Rev:	Latest	· · · · · · · · · · · · · · · · · · ·
Q,   ≚   ≑	•G, Y,										
earch: Q-			~								
Display Name P	Preview	Status	Vendor	File Version	Part	I/O Pin Count	Board Rev	Available IOBs	LUT Elements	FlipFlops	Block RAMs
Elbert_S7		Installed	numato.com	1.0	xc7s50csga324-1	324	1.0	210	32600	65200	75

In the next window, click "Finish" to complete creating the new project. When the new project wizard exits, a new project will be created by Vivado with the specified settings.

# 1. UART Communication – Printing "Hello World"

#### **INTRODUCTION**

In this first hands-on project, we will verify the UART (Universal Asynchronous Receiver/Transmitter) interface on the Elbert S7 FPGA board by printing a simple "Hello World" message to the serial terminal. UART is a widely used serial communication protocol that allows the FPGA to communicate with a host PC or other devices using simple text-based messaging.

This project serves as a basic sanity check to ensure the UART peripheral is functioning correctly and that the board can transmit data over a serial connection. It also helps familiarize users with integrating IP cores and observing output through a terminal emulator like Tera Term or PuTTY.

By completing this project, users will gain confidence in creating Vivado projects, generating bitstreams, using the BSP, and verifying output via UART communication.

**Creating Microblaze based Hardware Platform for Elbert S7** The following steps will walk you through the process of creating a new project with Vivado and building a hardware platform with MicroBlaze soft processor using an IP integrator.

#### STEP 1:

To create a new Vivado project specifically for the **Elbert S7** FPGA board, follow the procedure outlined in the section "<u>Getting Started with Vivado: Creating a New</u> <u>FPGA Project.</u>"

#### STEP 2:

After creating a new project successfully, In the "Flow Navigator" panel, select "Create Block Design" under the IP integrator section. Give an appropriate name (Eg: "Hello\_world ") to the design and click "OK ".

Create Block Design	I	×
Please specify name	of block design.	Д
<u>D</u> esign name:	Hello_world	
Directory:	Section 4	~
Specify source set:	🗅 Design Sources	~
?	ОК	Cancel

#### Step 3:

Go to Diagram window, right click and select "Add IP" from the popup menu. Search for "MicroBlaze" and add it to the design by double-clicking it.

Search: Q- mid (6 matches)	
👎 MicroBlaze	
👎 MicroBlaze Debug Module (MDM)	
👎 MicroBlaze Debug Module (MDM) V	
MicroBlaze MCS	
MicroBlaze MCS V	
👎 MicroBlaze V	
ENTER to select, ESC to cancel, Ctrl+Q for IP details	

Click "Run Block Automation" present in the "Designer Assistance available" bar on the top left corner of the window to complete the design. Select the settings as shown in the following image. Click "OK" for Vivado to automatically configure the blocks for you.

Q XII Automation (1 out of 1 selected) ✓ ✓ All Automation (1 out of 1 selected) ✓ ♥ ♥ microblaze_0	MicroBlaze Debug Module	tomation generates local memory of selected size, and caches can be configured. , Peripheral AXI Interconnect, Interrupt Controller, a clock source, Processor System Reset I as needed. A preset MicroBlaze configuration can also be selected.
	Information about the opti	ons can be found in the tooltips.
	Preset Local Memory	None ~ 64KB ~
	Local Memory ECC	None v
	Cache Configuration	None v
	Debug Module	Debug Only V
	Peripheral AXI Port	Enabled 🗸
	Clock Connection	New Clocking Wizard

#### Step 4:

Double click "Clocking Wizard" IP and customize "Board" settings as shown in the following image.

Clocking Wizard (6.0)			L
Documentation 📄 IP Location			
IP Symbol Resource	Component Name clk_wiz_1		
Show disabled ports	Board Clocking Options Output Clocks MMCM Settii	ngs Summary	
	Associate IP interface with board interface		
	IP Interface	Board Interface	
	CLK_IN1	sys clock	•
	CLK_IN2	Custom	•
	EXT_RESET_IN	reset	*
reset clk_out1			

#### Step 5:

Run "Connection Automation" and select all the pins.

Numato Lab<sup>®</sup> REV: V1.0

Run Connection Automation		×
Automatically make connections in your desig the right.	n by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on	λ
Q       X       ♦         ✓       All Automation (3 out of 3 selected)         ✓       ♥       totk_wiz_1         Ø       >       oth_in1         Ø       >       reset         ✓       ♥       rst_clk_wiz_1_100M         Ø       >       ext_reset_in	Select an interface pin on the left panel to view its options	
•	ОК Са	ncel

#### Step 6:

Go to the Board section, Drag and drop the USB UART from the Board section to the design.



Click on "Run Connection Automation" select all the pins and click ok.

Connect Slave interface (/axi_uartlite		
Options	_0/S_AXI) to a selected Master add	lress space.
Master interface	/microblaze_0 (Periph) 🗸 🗸	
Bridge IP	/microblaze_0_axi_periph 🗸 🗸	
Clock source for driving Bridge IP	/clk_wiz_1/clk_out1 (100 MHz)	×
Clock source for Slave interface	Auto	*
Clock source for Master interface	/clk_wiz_1/clk_out1 (100 MHz)	*
	Bridge IP Clock source for driving Bridge IP Clock source for Slave interface	Master interface     /microblaze_0 (Periph) ~       Bridge IP     /microblaze_0_axi_periph ~       Clock source for driving Bridge IP     /clk_wiz_1/clk_out1 (100 MHz)       Clock source for Slave interface     Auto

#### Step 7:

Connect interrupt output lines from "AXI Uartlite" to the "Concat" block as shown in the below figure. Select the "Validate Design" option from the "Tools" menu to make sure that connections are correct.



#### Step 8:

Select the "Validate Design" option from the "Tools" menu to make sure that connections are correct.



#### Step 9:

Right-click "Hello\_world" in the "Sources" window and select "Create HDL Wrapper" from the popup menu. Click "OK" on the window that appears to finish generating a wrapper.



#### Step 10:

Click "Generate Bitstream" under the "Program and Debug" section to synthesize, implement, and generate a bitstream.


# Step 11:

Once the implementation and generation of the bitstream are completed, we need to export the hardware along with the bitstream. Go to the "File" menu and select "Export->Export Hardware ". Select the "Include bitstream" checkbox and click "OK" in the "Export Hardware" wizard.

		Add Sources	Alt+A BLC	CK DESIGN - Hello_world		
	1	Save Block Design	Ctrl+S	Sources × Design		
		Save Block Design As	benn	Q 素 ♦ +		
		Close Block Design	sannach-ropannes	V 🖨 Design Sources (1)		
		Chec <u>k</u> point	> 1115	> • Hello_world_\		
		<u>C</u> onstraints	) b	<ul> <li>Constraints</li> <li>Simulation Sources</li> </ul>		
		Simulation Waveform	F	> a sim_1 (1)		
	1	<u>I</u> P	F	> 🚍 Utility Sources		
		Text E <u>d</u> itor	+			
		I <u>m</u> port	•			
		Export		Export <u>H</u> ardware		
	1	Print	Ctrl+P	Export Block Design Export Bitstream File		
		E <u>x</u> it		Export Simulation		
			1 1			
Set the platform proper		vnstream tools of the int re specification for down		he target platform's hard ire tools.	ware design.	Σ
<ul> <li>Pre-synthesis This platform in</li> <li>Include bitstrea This platform in</li> </ul>	dudes a hardwar n	re specification for down	stream softwa			ation for
Pre-synthesis This platform in Include bitstrea	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for
Set the platform proper Pre-synthesis This platform in Include bitstrea This platform in	dudes a hardwar n	re specification for down	stream softwa	are tools.		ation for

#### Step 12:

Select Launch Vitis IDE from the Tools menu.



#### Step 13:

After Vitis Unified IDE window opens, click on "Open Workspace" and select necessary folder to keep the Vitis files.



#### Step 14:

Create a new platform for the project, by selecting "Create Platform Component", click "Next", in the Flow tab select the XSA file saved using the step 11 and finally click "Next" and "Finish" respectively.





After successful creation of the platform, build the platform.



## Step 15:

Next create the Hello world Application component by selecting the "Hello world" template from the "examples",



In "Create Application Component" tab specify project name and location, click "Next".

Create Application Compone	ient - Hello World	×
Name and Location > I	Hardware > Domain > Sysroot > Summary	
Name and Location		
Choose a name for your co	omponent and specify a directory where component data files will be stored	
Component name	Hello	
Component location	C:\projects\Elbert_S7\Hello_world\VITIS  v Browse	
Component will be created a	at C:\projects\Elbert_S7\Hello_world\VITIS\Hello	
(Constant)		
Cancel		Next

Select newly created Platform and click "Next".

Create Application Component - Hello World					×
Name and Location > Hardware > Domain >					
Select Platform					
Platforms supporting the selected example from your	repositories	s. To create a new	platform, us	e "File -> New Component -> Platform"	
Q ७ − + ¥ \$					
NAME	BOARD	FLOW	VENDOR	РАТН	
C:\projects\Elbert_S7\Hello_world\VITIS\HELLO_WORLD\				t_S7\Hello_world\VITIS\HELLO_WORLD\export\HELLO_WORLD	
(1)					
HELLO_WORLD	elbert	Embedded	xilinx.com	VITIS\HELLO_WORLD\export\HELLO_WORLD\HELLO_WORLD.x Info	
Cancel				Back	

Select the domain as "Standalone\_microblaze\_0" and click "Next" and click on "Finish".



When the Helloworld project is added successfully, build the project manually.



**Step 16:** Once the build is completed successfully, power up Elbert S7 FPGA Development Board using USB type C cable.

**Step 17:** Program the FPGA on Elbert S7 with a simple boot loop program by selecting the Program Device option from the Vitis menu.



Once the "Program Device" window opens click on "Program".

Program Device		×
Specify the bitstream a	and the ELF files that reside in BRAM memory.	
Project	Hello ~	
Connection	Local v	New
Bitstream/PDI	:\projects\Elbert_S7\Hello_world\VITIS\Hello\_ide\bitstream\Hello_world_wrapper.bit	Browse Search
	Partial Bitstream	
BMM/MMI File	projects\Elbert_S7\Hello_world\VITIS\Hello\_ide\bitstream\Hello_world_wrapper.mmi	<u>Browse</u> <u>Search</u>
Software Configuration	n	
PROCESSOR	ELF/MEM FILE TO INITIALIZE IN BLOCK RAM	
microblaze_0	bootloop	~
Skip Revision Check		
Cancel		Generate Program

# **Step 18:**

Meanwhile, open any serial terminal program (such as PuTTY, Teraterm etc) and open the port corresponding to Elbert S7 with a 9600 baud rate (the default baud rate given in UART IP). Program the board by selecting the "Run."

✓ FLOW		≽	*
Component	🗰 Hello	~	錢
🕅 Build 🥪			
▷ Run			
🕸 Debug			

# Step 19:

If everything went well, the application running on the board should print "Hello World" over the UART and should be displayed on the Serial Terminal application.



# 2. Controlling Onboard Peripherals (LEDs, Seven Segment Display, and PMOD) with Slide Switches

# INTRODUCTION

This project demonstrates how to interface and control key onboard peripherals-**LEDs**, **Seven Segment Display**, and **PMOD connectors**-using the slide switches on the Elbert S7 FPGA board. It serves as a foundational experiment to understand how input signals (from switches) can control output peripherals in an FPGA-based system.

The design includes predefined behaviours:

- When **no switch is active**, the **seven-segment displays** continuously count from 0 to 9 (same digit on all displays), while the **LEDs** perform a running light pattern.
- When **specific switches are turned ON**, the system displays corresponding values:
  - **Switch 1 (SWO)**: The seven-segment shows **3**, and its binary representation (**00000011**) is shown on the LEDs.
  - Switch 2 (SW1): The seven-segment shows 5, and the LEDs display 00000101.
  - Switch 3 (SW2): The seven-segment shows 7, and the LEDs display
     00000111.
- When **Switch 4 (SW3)** is turned ON, **all PMOD pins go high**, enabling external module activation or signalling.

This project is designed as a basic interface demo. You are encouraged to **modify the functionality**-such as changing the numbers shown on the seven-segment display, customizing LED patterns, or controlling PMOD outputs differently-to suit your specific application needs or to experiment with your own designs.

By completing this project, you'll gain hands-on experience with I/O interfacing, logic control based on inputs.

#### STEP 1:

To create a new Vivado project specifically for the **Elbert S7** FPGA board, follow the procedure outlined in the section "<u>Getting Started with Vivado: Creating a New</u> **FPGA Project.**"

#### STEP 2:

After creating a new project successfully, In the Sources tab, Right-click '**Design Sources**' and click '**Add Sources**'. It will open a new '**Add Sources**' window.



Once the "Add Sources" window opens select "Add or create design sources" and click on "Next "

Add Sources	×
AMD Vivado ML Edition	Add Sources   This guides you through the process of adding and creating sources for your project Add or greate constraints Add or create design sources Add or create gimulation sources
?	< <u>B</u> ack <u>Einish</u> Cancel

## Step 3:

Download and extract the RTL source files from <u>here</u> and add them to the project by selecting 'Add Files' in the "Add or Create Design Sources" tab and click on "Finish".

+, - + +				
<b>T</b> ,  <sup>−</sup>   <b>T</b>   ♥				
	Lico Add Film A	dd Directories or Create F	ila huttana halaw	
	Use Add Flies, A	ad Directories of Create F	lie duttons delow	
		<b></b>		
	<u>A</u> dd Files	A <u>d</u> d Directories	<u>C</u> reate File	
Scan and add RTL include fi	iles into project	_		
Copy <u>s</u> ources into project				
Add so <u>u</u> rces from subdirec				

In Sources tab of Vivado, Right-Click on 'Constraints' and click 'Add Sources'.

<ul> <li>т.</li> </ul>		Properties	Ctrl+E
> 🖻 Con > 📮 Sim		Hierarchy Update	▶
> = 5im	C	Refresh Hierarchy	
> 🗀 Utili		IP Hierarchy	▶
		Copy Constraints Set	
		Edit Constraints Sets	
		Edit Simulation Sets	
	+	Add Sources	Alt+A
		Report IP Status	
Hierarch	y I	P Sources Libraries	Compile Order

# Step 5:

Once the **"Add Sources**" tab opens select **"Add or create constraints**" and click on **"Next**".

Add Sources		×
AMD Vivado ML Edition	Add Sources         This guides you through the process of adding and creating sources for your project         Add or greate constraints         Add or create design sources         Add or create gimulation sources	
?	< Back Einish Car	ncel

# Step 6:

Download and extract the xdc file from <u>here</u> and add them to the project by selecting 'Add Files' in the "Add or Create Constraints" tab and click on "Finish".

Add Sources	×
Add or Create Constraints Specify or create constraint files for physical and timing constraint	to add to your project.
Specify constraint set: Constrs_1 (active)	
Use Add File	s or Create File buttons below
Add	Files Create File
(?)	<back next=""> Einish Cancel</back>

# Step 7:

In Project Manager tab, click on 'Generate Bitstream'.

✓ SYNTHES	IS
🕨 Run S	synthesis
> Oper	Synthesized Design
✓ IMPLEME	NTATION
🕨 Run li	mplementation
> Oper	Implemented Design
✓ PROGRA	M AND DEBUG
📲 Gener	rate Bitstream
> Oper	n Hardware Manager

#### Step 8:

Once the bitstream is successfully generated, close any "**Bitstream Generation Completed**" dialog which comes up asking for what to do next.

Bitstream Generation Completed	×	
<b>i</b> Bitstream Generation successfully completed.		
Next		
Open Implemented Design		
○ <u>V</u> iew Reports		
Open <u>H</u> ardware Manager		
○ <u>G</u> enerate Memory Configuration File		
Don't show this dialog again		
OK Cancel		

# Step 9:

Click on 'Open target' and 'Auto Connect'.

HARDWARE MANAGER - unconnected				
🕖 No hardware target is	s opei	n. Open target		
Hardware	ø	Auto Connect	псx	
		Recent Targets		
Q   ≚   ≑   Ø		Available Targets on Server	*	
		Open New Target		
			2	
		No content		

# Step 10:

Right Click on the device (**xc7s50\_0**) and select "**Program Device**" option.

# Step 11:

Click "**Program**" and observe the output.

Program Device		×	
	ramming file and download it to your hardware device. You can Ig probes file that corresponds to the debug cores contained in the g file.		
Bitstre <u>a</u> m file: Debu <u>q</u> probes file: ✔ <u>E</u> nable end of st	ieven_segment/Seven_segment.runs/impl_1/Seven_segment.bit artup check	•••	
?	<u>P</u> rogram Ca	ncel	

# 3. HDMI OUTPUT EXAMPLE DESIGN

# INTRODUCTION

HDMI (High-Definition Multimedia Interface) represents a significant advancement over the older VGA standard by offering a digital solution that integrates both highresolution video and audio into a single interface. Unlike VGA, which relies on analog signals, HDMI uses digital transmission to deliver superior picture and sound quality. HDMI achieves this by transmitting pixel data serially at ten times the pixel clock frequency through TMDS (Transition Minimized Differential Signalling). This method reduces the number of signal transitions, which helps minimize potential data errors and maintains signal integrity.

The tutorial focuses on demonstrating DVI-D output through the Elbert S7 FPGA Module. DVI-D, or Digital Visual Interface – Digital, is a variant of HDMI and shares the same electrical and physical layer specifications. As a result, HDMI cables can also carry DVI-D signals, meaning that HDMI monitors are fully compatible with DVI-D outputs.

#### STEP 1:

To create a new Vivado project specifically for the **Elbert S7** FPGA board, follow the procedure outlined in the section "<u>Getting Started with Vivado: Creating a New</u> <u>FPGA Project.</u>"

#### Step 2:

After creating a new project successfully, then in the Sources tab, Right-click '**Design Sources'** and click '**Add Sources**'. It will open a new '**Add Sources'** window.

Sources			? _ O Ľ >
Q   ₹   <b>≑</b>   <b>+</b>	? 0		٥
🗅 Design Sources			
> 🖻 Constraints	Properties	Ctrl+E	
✓ ➡ Simulation So	Hierarchy Update	•	
🖻 sim_1	Refresh Hierarchy		
> 🖻 Utility Sources	IP Hierarchy	<b>•</b>	
	Copy Constraints Set		
	Edit Constraints Sets		
	Edit Simulation Sets		
Hierarchy Libra	Add Sources	Alt+A	

Once the "Add Sources" window opens select "Add or create design sources" and click on "Next "

Add Sources		×
AMD Vivado ML Edition	Add Sources         This guides you through the process of adding and creating sources for your project         Add or greate constraints         Add or create design sources         Add or create gimulation sources	
?	< Back Next > Einish Can	icel

# Step 3:

Download and extract the RTL source files from <u>here</u> and add them to the project by selecting 'Add Files' in the "Add or Create Design Sources" tab and click on "Finish".

Add or Create Design Sources Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source file on disk
and add it to your project.
+  +  +
Use Add Files, Add Directories or Create File buttons below
Add Files Add Directories Create File
Scan and add RTL include files into project Copy sources into project
Lopy sources into project

The HDMI interface has 3 pairs of differential data signals and 1 pair of differential clock signals:

- data\_p[2:0] & data\_n[2:0] : These are HDMI/DVI differential signals carrying the video data to be displayed on screen.
- clk\_p & clk\_n : HDMI pixel clock differential pair of signals.

First, VGA signals are generated inside vga module. Then the VGA signals are encoded to 10-bits per channel and the data is then serialised to 10x of pixel clock rate. Finally, the three channels along with pixel clock are driven out using TMDS differential drivers.

In the top module (dvid\_test), the two submodules dvid and vga are instantiated. Clocking IP (clocking wizard) is used to generate clocks for VGA and DVI-D.

dvid\_test: In this module, a "Clocking Wizard" IP core is instantiated to generate required clocks for VGA and DVI-D. A 100MHz clock from the onboard oscillator is provided as input, and following clocks are derived from it:

- clk\_vga: 25MHz clock. This is the pixel clock frequency for 640×480@60Hz VGA resolution.
- clk\_dvi & clk\_dvin : 125 MHz clocks. clk\_dvin is 180 degrees out of phase to clk\_dvi. These clocks are used for serialization using ODDR2.

vga: VGA signals are generated in this module. This design generates VGA at 640×480@60 Hz resolution.

dvid: VGA signals and clocks are given as input to this module and the DVI TMDS signals are generated as the output. It uses TMDS\_encoder module to generate TMDS signals. TMDS uses 8b/10b encoding in which the 8-bit color data (red, green & blue) generated in VGA module is converted to 10 bits. Then this data is serialised using ODDR2 (Double Data Rate primitive). The 10-bit TMDS data is generated at 25 MHz. ODDR2 uses 5 times the frequency of pixel clock (i.e. 125MHz) to serialize the 10-bit encoded data. Note that ODDR2 serialises 2-bits in 1 clock cycle of 125MHz clock. This serialised data is converted into differential signals in top module (dvid\_test) using OBUFDS drivers.

# Step 4:

Add Clocking Wizard by clicking on IP catalog in Project Manager, type '**clocking**' in search box and double-click '**Clocking Wizard**' IP. It will open customisation window for '**Clocking Wizard**'.

Project Summary	y × IP Catalog ×					? 🗆	3 6
Cores   Interfa	ices						
Q	🔻 🔩 🖌 🖉 👜	0					۰
Search: Q. clocki	ing	(2 matches)					
Name		^1 AXI4	Status	License	VLNV		
🗸 🗁 Vivado Rep	pository						
🗸 🗁 FPGA Fe	eatures and Design						
🗸 🖻 Clock	king						
👎 Cl	locking Wizard	AXI4	Production	Included	xilinx.com:ip:clk_wiz:6.0		
Details							
Name: Clo	ocking Wizard						â
Version: 6.0	0 (Rev. 14)						
Interfaces: AX	XI4						~

# Step 5:

In '**Clocking Options**' tab, give Component Name as '**clocking**' and primary clock port name as '**clk\_in**'.

Documentation  P IP Location C	C Switch to Defaults	
IP Symbol Resource	Component Name docking	
IP Symbol         Resource           Image: Show disabled ports         Image: Show disabled ports	Board         Clocking Options         Output Clocks         Port Renaming         MMCM Settings         Summary           Image: Summary Synthesis         Image: Image: Summary Summa	~
	Phase Alignment Spread Spectrum Minimize Output Jitter	
	Dynamic Reconfig Dynamic Phase Shift Maximize Input Jitter filtering	
+ s_axijite + CuCJN1,D + CUCJN2,D + CUCJR2,D CUCFB_INLD CUCFB_OUT_D +	Safe Clock Startup Dynamic Reconfig Interface	
	Options       Phase Duty Cycle Config       Write DRP registers <ul> <li>AXI4Lite</li> <li>DRP</li> </ul> Input Clock Information	
<ul> <li>Lasi_aresth</li> <li>clictop100</li> <li>reset</li> <li>iterrupt</li> <li>reset,</li> <li>iterrupt</li> <li>ref_clk</li> <li>dl_ootd</li> <li>oter_clk1</li> <li>looked</li> <li>oter_clk2</li> </ul>	Optione     Orr     AXI4Lite ODRP     DRP     Phase Duty Cycle Config Write DRP registers	
<ul> <li>s_aki_aresetn</li> <li>cl_sitopi30i</li> <li>reset</li> <li>dl_sitopi30i</li> <li>resetn</li> <li>dl_sitopi30i</li> <li>ref_cik</li> <li>dl_sort[30]</li> <li>user_ciki</li> <li>locked</li> </ul>	Optione  AXI4Lite DRP Phase Duty Cycle Config Write DRP registers Input Clock Information	
<ul> <li>Lasi, arestn</li> <li>di, sopi, arestn</li> <li>di, gighch (3,0)</li> <li>reset</li> <li>interrupt</li> <li>ref, dk</li> <li>user, dk0</li> <li>user, dk1</li> <li>ločetel</li> <li>user, dk1</li> <li>user, dk2</li> </ul>	Optione       Phase Duty Cycle Config       Write DRP registers         Input Clock Information       Input Clock       Port Name       Input Frequency(MHz)       Jitter Options       Input.	•

In '**Output Clocks**' tab, enable 3 output clocks and provide their name, frequency as well as phase as shown in the image below. Click '**OK**'.

Documentation 📮 IP Location C	Switch to Defaults						
IP Symbol Resource	Component Nan	ne clocking					
Show disabled ports	Board Clock	ting Options	Output Clocks	Port	Renaming MI	MCM Settings Sum	ımary
	The phase is ca	lculated relative to	o the active input	clock.			
	Output Clock	Port Name	Output Freq (	MHz)		Phase (degrees)	
	Clk_out1	CLK DVI	Requested	Ø	Actual 125.00000	Requested	Actual
CLK_DVI 🗕		-					-
– reset CLK_DVIn –	✓ clk_out2	CLK_DVIn 😣		8		0.000	-
– clk_in CLK_VGA –	Clk_out3	CLK_VGA 🛛	25.000	8	25.00000	0.000	0.000
locked 🗕	clk_out4	clk_out4	100.000		N/A	0.000	N/A
	clk_out5	clk_out5	100.000		N/A	0.000	N/A
	clk_out6	clk_out6	100.000		N/A	0.000	N/A
	- · · ·						

# Step 6:

In Sources tab of Vivado, Right-Click on 'Constraints' and click 'Add Sources'.



#### Step 7:

Once the "Add Sources" tab opens select "Add or create constraints" and click on "Next ".

Add Sources		×
AMD Vivado ML Edition	Add Sources         This guides you through the process of adding and creating sources for your project         Add or greate constraints         Add or create design sources         Add or create simulation sources	
?	< <u>B</u> ack <u>Next</u> > <u>F</u> inish C	Cancel

# Step 8:

Download and extract the XDC file from <u>here</u> and add them to the project by selecting 'Add Files' in the "Add or Create Constraints" tab and click on "Finish".

Step 9:		
In Project Manager tab,	click on 'Generate Bitstream'.	
	✓ SYNTHESIS	
	Run Synthesis	
	> Open Synthesized Design	
	✓ IMPLEMENTATION	
	Run Implementation	
	> Open Implemented Design	
	Y PROGRAM AND DEBUG	
	Generate Bitstream	
	> Open Hardware Manager	

#### Step 10:

Once the bitstream is successfully generated, close any "**Bitstream Generation Completed**" dialog which comes up asking for what to do next.



#### Step 11:

Now click 'Open Hardware Manager' to program the FPGA.



#### Step 12:

Click on 'Open target' and 'Auto Connect'.

HARDWARE MANAGER - unconnected				
<b>1</b> No hardware target is open. Open target				
Hardware	ø	Auto Connect	осх	
		Recent Targets		
0,   ≚   ≑   Ø		Available Targets on Server	*	
		Open New Target		
			, 	
		No content		

# Step 13:

Right Click on the device (**xc7s50\_0**) and select "**Program Device**" option.

# Step 14:

Click "**Program**" and observe the output.

Program Device	×
	gramming file and download it to your hardware device. You can optionally file that corresponds to the debug cores contained in the bitstream
Bitstre <u>a</u> m file: Debu <u>g</u> probes file: ✔ Enable end of s	
?	Program Cancel

#### Step 15:

Once Elbert S7 is successfully programmed, it should begin generating HDMI signals and the monitor should display a colourful pattern at 640×480 @ 60Hz resolution.



That was it! You can play with the vga module to output different patterns and try to generate higher resolutions as well.

# 4. DDR3 Mem test on ELBERT S7

# **INTRODUCTION**

This article aims to guide readers on how to effectively utilize the DDR3 memory available on the Spartan-7 FPGA using the AMD Memory Interface Generator (MIG) 7 IP Core. The MIG 7 IP core is a powerful tool provided by AMD that simplifies the process of interfacing with external DDR3 memory by handling the complex timing and calibration requirements.

When working with the MIG 7 IP core, users are presented with two interface options:

User Interface: This is a straightforward wrapper built on top of the Native Interface. It simplifies communication with the DDR3 memory by providing clear signal naming and intuitive data flow control.

AXI4 Interface: This interface follows the AXI4 protocol, which is widely used in FPGA designs for memory-mapped transactions. It allows seamless integration with AXI-based designs, providing better scalability and compatibility with IP cores that utilize the AXI protocol.

In this tutorial, we'll focus on how to test the DDR3 memory using the Memory Tests Template available in Vitis. This test template is a convenient way to verify memory functionality, ensuring that data can be reliably written to and read from the DDR3 memory. By following this guide, you'll gain insights into configuring the MIG 7 IP core, integrating it into your Vivado design, and performing practical memory tests on the Spartan-7 FPGA.

Whether you're a beginner or an experienced FPGA developer, this article will provide clear steps and explanations to help you successfully interface with DDR3 memory on the Spartan-7 platform.

#### STEP 1:

To create a new Vivado project specifically for the **Elbert S7** FPGA board, follow the procedure outlined in the section "<u>Getting Started with Vivado: Creating a New</u> <u>FPGA Project.</u>"

# Step 2:

After creating a new project, Under the "Flow Navigator" panel, click "Create Block Design" under the IP Integrator section. Enter a name for the block design and click "OK". An empty block design will be created.

	Create Block Design		×
	Design name: Directory: Specify source set: ?	DDR Sources	
Step 3:	IM	ato	Lab

In the Diagram window, right-click and select "Add IP" from the popup menu. Search for "MicroBlaze" & "AXI Timer" and add them to the design by double-clicking them.

Search: Q- MICR (6 matches)
👎 MicroBlaze
👎 MicroBlaze Debug Module (MDM)
👎 MicroBlaze Debug Module (MDM) V
👎 MicroBlaze MCS
MicroBlaze MCS V
👎 MicroBlaze V
ENTER to select, ESC to cancel, Ctrl+Q for IP details

Search:	Q- AXI TI	(6 matches)
👎 AXI4-	Stream Verification IP	
👎 AXI N	lulti Channel Direct Me	emory Access
👎 AXI S	ideband Utility	
👎 AXI T	ïmebase Watchdog T	imer
👎 AXI T	ïmer	
👎 AXI V	erification IP	
ENTER to	select, ESC to cance	I, CtrI+Q for IP details

# Step 4:

Click "Run Block Automation" present in the "Designer Assistance available" bar on the top left corner of the window to complete the design. Select the settings as shown in the following image.

Q,   ¥,   ≑	Description
<ul> <li>All Automation (1 out of 1 select</li> <li># microblaze_0</li> </ul>	cled)       MicroBlaze connection automation generates local memory of selected size, and caches can be configured.         MicroBlaze Debug Module, Peripheral AXI Interconnect, Interrupt Controller, a clock source, Processor System Reset are added and connected as needed. A preset MicroBlaze configuration can also be selected.         Information about the options can be found in the tooltips.
	Options
	Preset Vone V
	Local Memory 64KB 🗸
	Local Memory ECC None 🗸
	Cache Configuration Vone V
	Debug Module Debug Only ~
	Peripheral AXI Port Enabled V
	Interrupt Controller      Clock Connection      New Clocking Wizard
	Clock Connection New Clocking Wizard V
?	OK Cancel

Click the "Board" tab. The default peripherals available for the Elbert board will be listed as shown below.



Add DDR3 SDRAM and USB UART to the design by double-clicking the corresponding peripherals.

#### Step 6:

Double-click on the "Clocking Wizard" IP block and change the settings as shown below. In the "Output Clocks" section, set clk\_out1 frequency to 100 MHz and clk\_out2 to 200MHz.

P Symbol       Resource         Component Name       ck_wt_1         Show disabled ports       Component Name         Imiterface       Board Interface         Imiterface       Imiterface         Imiterfa	Clocking Wizard (6.0)			۲۲ ۲
Show disabled ports     Baard     Clocking Options     Output Clocks     MMCM Settings     Summary       Associate IP Interface     Board Interface     Board Interface       CLK_IN1     sys clock        CLK_IN1     custom        EXT_RESET_IN     reset        Clear Board Parameters	Documentation 🚡 IP Location			
Board     Clocking Options     Output Clocks     MMCM Settings     Summary       Associate IP Interface     IP Interface     Board Interface       IP Interface     Output Clocks     Sys clock        CLK_IN1     Sys clock        CLK_IN2     Custom        EXT_RESET_IN     reset	IP Symbol Resource	Component Name clk_wiz_1		
reset clk_out1	Show disabled ports		nmary	
CLK_IN1 sys dock   CLK_IN2 Custom    CLK_IN2 Custom			Poard Interface	
CLK_IN2 Custom				
reset cik_out1				•
reset clk_out1				•
	rocot olk out1			

cking Wizard (6.0)												
ocumentation 📄 IP Location												
P Symbol Resource	Compone	ent Name	clk_wiz_1									
) Show disabled ports	Board	Clocking	Options O	utput Clocks MI	лсм	Settings Sum	nmary					
	The pha	se is calcı	lated relative t	o the active input clo	ock.							
	Output		Port Name	Output Freq (MH	Z)		Phase (deg			Duty Cycle (%		Drives
				Requested		Actual	Requested		Actual	Requested	Actual	
	✓ clk_	out1	clk_out1	_	-	100.00000	0.000	_	0.000	50.000	50.0	BUFG
	✓ clk_	out2	clk_out2	200.000	8 1	100.00000	0.000	8	0.000	50.000	50.0	BUFG
	dk_	out3	clk_out3	100.000		N/A	0.000		N/A	50.000	N/A	BUFG
	Clk_	out4	clk_out4	100.000		N/A	0.000		N/A	50.000	N/A	BUFG
	Clk_	out5	clk_out5	100.000		N/A	0.000		N/A	50.000	N/A	BUFG
	Cik_	out6	clk_out6	100.000		N/A	0.000		N/A	50.000	N/A	BUFG
clk	out1 - Ck	out7	clk_out7	100.000		N/A	0.000		N/A	50.000	N/A	BUFG
- reset												
	_out2 💻 🗌 us	E CLOCK	SEQUENCING	3	Clo	cking Feedback						
- clk_in1	cked -					£			Cinnalia	-		
10	Ou	tput Clock	Sequenc	e Number		Source			Signalin	9		
		clk_out1	1			Autom	atic Control On	-Chip	۲	) Single-ended		
		clk_out2	1			O Autom	atic Control Off	-Chip		) Differential		
	(	clk_out3	1			O User-O	Controlled On-O	Chip				
		clk_out4	1			O User-O	Controlled Off-C	Chip				
		clk_out5	1									
		clk_out6	1									
		clk_out7	1									
	Enable	Optional Ir	puts / Outputs	s for MMCM/PLL		Reset	Туре					

**Step 7:** Remove the existing connection to sys\_clk\_i of the "MIG 7 Series" block and connect it to clk\_out2.

**Step 8:** Run "Connection Automation" so Vivado can connect the blocks to make a complete system.

Run Connection Automation		×
Automatically make connections in your design the right.	by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on	λ
Q       X       ♦         V       ✓       All Automation (7 out of 7 selected)         V       ✓       ♥ axi_time_0         V       ♥       axi_uartitle_0         V       ♥       axi_uartitle_0         V       ♥       Tak_uartitle_0         V       ♥       axi_uartitle_0         V       ♥       Tak_uitartitle_0         V       ♥       Tak_uitartitle_0	Select an interface pin on the left panel to view its options	
?	ОК Са	ncel

Click on "Run Block Automation" and select keep Classic Microblaze option as shown in the picture below.

2   ₹   ≑	Description	
All Automation (1 out of 1 selected) Image: The selected is a selected in the se	MicroBlaze conversion automation converts classic MicroBlaze processors to the RISC-V MicroBlaze V processor. The corresponding MicroBlaze Debug Module (MDM) is also converted.	
	Information about the options can be found in the tooltips.	
	Options	
	✓ Keep Classic MicroBlaze ✓ Enable Compressed Instructions	

**Step 9:** Connect interrupt output lines from "**AXI Timer**" and "**UARTLite**" to the "**Concat**" block as shown below figure. Select the "**Validate Design**" option from the **Tools** menu to make sure that connections are correct.



**Step 10:** Right-click "**ddr3**" in the "**Sources**" window and select "**Create HDL Wrapper**" from the popup menu. Click "**OK**" on the window that appears to finish generating a wrapper



# Step 11:

Click **"Generate Bitstream**" under the **"Program And Debug**" section to synthesize, implement and generate a bitstream.

Run Implementation
> Open Implemented Design
<ul> <li>PROGRAM AND DEBUG</li> </ul>
👫 Generate Bitstream
> Open Hardware Manager

**Step 12:** After generating the bitstream successfully, select **Export -> Export Hardware** from the **File menu**. Click **Next**.

	· · · ·	✓ □ Design Sources (2) > ● ♣ DDR_wrappe	e i	
Export	•	Export <u>H</u> ardware		
Print Exit	Ctrl+P	Export Block Design Export Bitstream File		

Select the "include bitstream" checkbox and click Next.

Expo	ort Hardware Platform	×
Outp Set the	ut e platform properties to inform downstream tools of the intended use of the target platform's hardware design.	Д
0	Pre-synthesis This platform includes a hardware specification for downstream software tools.	
۲	Include bitstream This platform includes the complete hardware implementation and bitstream, in addition to the hardware specification for software tools.	
	<back next=""> Einish Canc</back>	el

Export Hardwa	re Platform	×
Files		ភា
Enter the name of	f your hardware platform file, and the directory where the XSA file will be stored.	
XSA file name:	DDR_wrapper	8
Export to:	C:/projects/Elbert_S7/DDR3	⊗
	The XSA will be written to: C:\projects\Elbert_S7\DDR3\DDR_wrapper.xsa	
	<back next=""> Finish</back>	Cancel

Step 13: Select Launch Vitis IDE from the Tools menu.



**Step 14:** After Vitis Unified IDE window opens, click on "**Open Workspace**" and select necessary folder to keep the Vitis files.



**Step 15:** Create a new platform for the project, by selecting "**Create Platform Component**", click "**Next**", in the Flow tab select the XSA file saved using the step 12 and finally click "**Next**" and "**Finish**" respectively.



	Create Platform Component	×
	Name and Location > Flow > OS and Processor > Summary	
	Select Platform Creation Flow	
	Create a platform component by selecting the hardware design and add software domains.	
•	Hardware Design     Existing Platform	R
	Hardware Design (XSA) C:\projects\Elbert_S7\Hello_world\Hello_world_wrapper.xsa > Browse For Implementation	
	> Advanced Options	
	Cancel	Back Next

After successful creation of the platform, build the platform.



**Step 16:** Next create the DDR3\_test Application component by selecting the "Memory tests" template from the "examples",



In "Create Application Component" tab specify project name and location, click "Next"

Create Application Component - Memory Tests							
Name and Location > Hardware > Domain > Sysroot > Summary							
Name and Location							
Choose a name for your co	Choose a name for your component and specify a directory where component data files will be stored						
Component name	memory_tests						
Component location	C:\projects\Elbert_S7\DDR3\vitis_unified \v Browse						
Component will be created a	at C:\projects\Elbert_S7\DDR3\vitis_unified\memory_tests						
Cancel		lext					

Select newly created Platform and click "Next".



Select the domain as "Standalone\_microblaze\_o" and click "Next" and click on "Finish"

eate Application Component - Memory T Name and Location > Hardware >			×
	Johnan , January		
ielect Domain Thoose a domain from the available dom	ains in the platform		
Name	Details		
standalone_microblaze_0 + create new	Name Display Name OS Processor	standalone_microblaze_0 standalone_microblaze_0 standalone microblaze_0	

When the Memory tests project is added successfully, build the project manually.

~	FLOV	N		≽	*
	Com	ponent	memory_tests	~	\$
	<i>?</i> ∼ e	Build			4
	⊳ F	Run			
	资 [	Debug			

**Step 17:** Once the build is completed successfully, power up Elbert S7 using an USB type C cable

**Step 18:** Program the FPGA on Elbert S7 with a simple boot loop program by selecting the **Program Device** option from the **Vitis menu**.



Once the "Program Device" window opens click on "Program".

Program Device				×		
Specify the bitstream and the ELF files that reside in BRAM memory.						
Project	memory_tests ~					
Connection	Local ~	<u>New</u>				
Bitstream/PDI	rojects\Elbert_57\DDR3\vitis_unified\memory_tests\_ide\bitstream\DDR_wrapper.bit	<u>Browse</u>	<u>Search</u>			
	Partial Bitstream					
BMM/MMI File	jects\Elbert_S7\DDR3\vitis_unified\memory_tests\_ide\bitstream\DDR_wrapper.mmi	<u>Browse</u>	<u>Search</u>			
Software Configuration						
PROCESSOR	ELF/MEM FILE TO INITIALIZE IN BLOCK RAM					
microblaze_0	bootloop			~		
Skip Revision Check						
Cancel		Gener	ate P	rogram		

#### Step 19:

Meanwhile, open any serial terminal program (such as PuTTY, Teraterm etc) and open the port corresponding to Elbert S7 with a 9600 baud rate (the default baud rate given in UART IP). Program the board by selecting the "Run".



# Step 20:

If everything went well, the application running on the board should print the memory testing Process over the UART and should be displayed on the Serial Terminal application.

B COM45 - PuTTY	-		×
Starting Memory Test Application			
NOTE: This application runs with D-Cache disabled.As a result,	cachelin	e req	uest
s will not be generated			
Testing memory region: mig 0			
Memory Controller: mig 0			
Base Address: 0x80000000			
Size: 0x10000000 bytes			
Memory Test Application Complete			
Successfully ran Memory Test Application			
			Ψ.
	,		

# 5. Gigabit Ethernet Example Design

#### INTRODUCTION

Ethernet is a Link Layer Protocol in the TCP/IP protocol stack between the physical and data link layer. It is the most widely used protocol for Local Area Networks (LANs). Every device on Ethernet is assigned a unique MAC address for communication. <u>Gigabit Ethernet</u> refers to various technologies developed for transmitting Ethernet frames at the rate of gigabits per second. The <u>Reduced Gigabit</u> <u>Media-Independent Interface (RGMII)</u> is used to interface the Ethernet IP core on FPGA with the Gigabit Ethernet PHY chip on Elbert S7. The Media Access Layer converts the packets into a stream of data to be sent while the Physical Layer converts the stream of data into electrical signals. RGMII provides a media-independent interface so that MAC and PHY can be compatible, irrespective of the hardware used. In this tutorial, the Numato Lab Elbert S7 FPGA Development Board is used to demonstrate a TCP/IP echo server application. The echo server application runs on lightweight IP (lwIP) TCP/IP stack.

#### STEP 1:

To create a new Vivado project specifically for the **Elbert S7** FPGA board, follow the procedure outlined in the section "<u>Getting Started with Vivado: Creating a</u> <u>New FPGA Project.</u>"

**Step 2:** After creating a new project successfully, In the Flow Navigator panel, select Create Block Design under IP INTEGRATOR. Enter a name for the block design and click OK. An empty block design will be created


**Step 3:** Go to Diagram window, right click and select "Add IP" from the popup menu. Search for "MicroBlaze" and add it to the design by double-clicking it.

Search: Q- mid (6 matches)
👎 MicroBlaze
👎 MicroBlaze Debug Module (MDM)
👎 MicroBlaze Debug Module (MDM) V
👎 MicroBlaze MCS
👎 MicroBlaze MCS V
👎 MicroBlaze V
ENTER to select, ESC to cancel, Ctrl+Q for IP details

Click "Run Block Automation" present in the "Designer Assistance available" bar on the top left corner of the window to complete the design. Select the settings as shown in the following image. Click "OK" for Vivado to automatically configure the blocks for you.

	n by checking the boxes of the blocks to connect. Select a block on the left to display its configuration options on t	he right.
Q ≥ ✓ All Automation (1 out of 1 selected)	Description MicroBlaze connection automation generates local memory of selected size, and caches can be configured	
	MicroBlaze Debug Module, Peripheral AXI Interconnect, Interrupt Controller, a clock source, Processor Syst are added and connected as needed. A preset MicroBlaze configuration can also be selected. Information about the options can be found in the tooltips. Options	em Reset
	Preset None V Local Memory 64KB V Local Memory ECC None V	
	Cache Configuration None  Debug Module Debug Only  Peripheral AXI Port Enabled	
	Interrupt Controller  Clock Connection  New Clocking Wizard	

**Step 4:** Double click "Clocking Wizard" IP and customize "Board" settings as shown in the following image.

IP Symbol Resource											
Show disabled ports		Component Nam	e CIK_WIZ_1								
		Board Clocki	ng Options	Output Clocks	имсм	Settings Su	mmary				
		Associate IP inte IP Interface	erface with boar	d interface			Board Inte	arface			
		CLK_IN1					sys clock	enace			-
		CLK_IN2					Custom				•
		EXT_RESET_IM					reset				•
		Clear Board	Parameters								
- reset	clk_out1 💻										
clk_in1	locked -										
-	J										
Re-customize IP										ОК	Cancel
locking Wizard (6.0)										OK	Cancel
Re-customize IP Slocking Wizard (6.0) Documentation 💿 IP Location										<u>ок</u>	
Clocking Wizard (6.0) Documentation 🚔 IP Location IP Symbol Resource		Component Nam	e cik_wiz_1							OK	
locking Wizard (6.0) Documentation 🕞 IP Location IP Symbol Resource				Dutout Clocks		I Settings Su	mmary			ок	
locking Wizard (6.0) Documentation 🕞 IP Location IP Symbol Resource		Board Clockin	ng Options	Dutput Clocks 1			mmary			<u>OK</u>	
Clocking Wizard (6.0) Documentation 🚔 IP Location IP Symbol Resource		Board Clockin The phase is ca	ng Options C	to the active input Output Freq (M	clock. IHz)		Phase (degr		Duty Cycle	(%)	5
Clocking Wizard (6.0) Documentation 🚔 IP Location IP Symbol Resource		Board Clockin The phase is ca Output Clock	ng Options C Iculated relative Port Name	to the active input Output Freq (M Requested	clock. IHz)	Actual	Phase (degr Requested	Actual	Requested	(%) Actual	Drives
Clocking Wizard (6.0) Documentation 🚔 IP Location IP Symbol Resource		Board Clockin The phase is ca Output Clock I clk_out1	rg Options C Iculated relative Port Name Clk_out1	to the active input Output Freq (M Requested 100.000	clock. IHz)	Actual 100.00000	Phase (degr Requested 0.000	Actual  0.000	Requested 50.000	(%) Actual 50.0	Drives BUFG
Clocking Wizard (6.0) Documentation 🚔 IP Location IP Symbol Resource		Board Clockin The phase is ca Output Clock I clk_out1 I clk_out2	Options     Control     Contro     Control     Control     Control     Control     Control     Co	to the active input Output Freq (M Requested 100.000 200.000	clock. IHz)	Actual 100.00000 200.00000	Phase (degr Requested	Actual 0.000 0.000	Requested 50.000 50.000	(%) Actual 50.0 50.0	Drives
Clocking Wizard (6.0) Documentation 🚔 IP Location IP Symbol Resource		Board Clockin The phase is ca Output Clock I cik_out1 I cik_out2 I cik_out3	Ing Options C Iculated relative Port Name Clk_out1 Clk_out2 Clk_out3	to the active input Output Freq (M Requested 100.000 200.000 125.000	clock. IHz)	Actual 100.00000 200.00000 100.00000	Phase (degr           Requested           0.000           0.000           0.000	Actual 0.000 0.000 0.000 0.000	Requested 50.000 50.000 50.000	(%) Actual 50.0 50.0 50.0	Drives BUFG BUFG BUFG
locking Wizard (6.0) Documentation 🕞 IP Location IP Symbol Resource		Board Clockin The phase is ca Output Clock If dk_out1 If dk_out2 If dk_out3 If dk_out4	Port Name dk_out1 dk_out2 dk_out3 dk_out4	to the active input Output Freq (M Requested 100.000 200.000	clock. IHz)	Actual 100.00000 200.00000	Phase (degr Requested 0.000 0.000	Actual 0.000 0.000	Requested 50.000 50.000	(%) Actual 50.0 50.0	Drives BUFG BUFG
Clocking Wizard (6.0) Documentation 🚔 IP Location IP Symbol Resource		Board Clockin The phase is ca Output Clock I cik_out1 I cik_out2 I cik_out3	Ing Options C Iculated relative Port Name Clk_out1 Clk_out2 Clk_out3	to the active input Output Freq (M Requested 100.000 200.000 125.000 100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 N/A	Phase (degr           Requested           0.000           0.000           0.000           0.000           0.000           0.000	Actual  Actual  O.000  O.000  O.000  N/A	Requested           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 N0A	Drives BUFG BUFG BUFG BUFG
locking Wizard (6.0)	clk_out1	Board Clockii The phase is ca Output Clock Ø dk_out1 Ø dk_out2 Ø dk_out3 dk_out4 Ø dk_out5	Port Name Cliculated relative Port Name Cliculated relative Clicul	to the active input Output Freq (M Requested 100.000 125.000 100.000 100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 N/A N/A	Phase (degr           Requested           0.000           0.000           0.000           0.000           0.000           0.000	Actual  Actual  O.000  O.000  N/A  N/A  N/A	Requested           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 10/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
Clocking Wizard (6.0) Documentation 🚔 IP Location IP Symbol Resource	cik_out1	Board     Clocki       The phase is ca       Output Clock       Image: Image of the phase is ca       Image of the phase is ca <td>rg Options C culated relative Port Name clk_out1 clk_out2 clk_out3 clk_out4 clk_out5 clk_out6</td> <td>to the active input. Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000</td> <td>clock. IHz)</td> <td>Actual 100.00000 200.00000 100.00000 N/A N/A N/A</td> <td>Phase (degr Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000</td> <td>Actual           ©         0.000           ©         0.000           ©         0.000           N/A         N/A           N/A         N/A</td> <td>Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000</td> <td>(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100</td> <td>Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG</td>	rg Options C culated relative Port Name clk_out1 clk_out2 clk_out3 clk_out4 clk_out5 clk_out6	to the active input. Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 N/A N/A N/A	Phase (degr Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000	Actual           ©         0.000           ©         0.000           ©         0.000           N/A         N/A           N/A         N/A	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
IDocumentation IP Location IP Symbol Resource Show disabled ports	clk_out2 🗕	Board     Clocki       The phase is ca       Output Clock       Image: Image of the phase is ca       Image of the phase is ca <td>Options     O       cutuated relative     Port Name       ck_out1     ck_out2       ck_out2     ck_out3       ck_out4     ck_out4       ck_out5     ck_out7</td> <td>to the active input           Output Freq (M           Requested           100.000           200.000           125.000           100.000           100.000           100.000           100.000           100.000           100.000</td> <td>clock. IHz)</td> <td>Actual 100.00000 200.00000 100.00000 N/A N/A N/A</td> <td>Phase (degr Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000</td> <td>Actual           ©         0.000           ©         0.000           ©         0.000           N/A         N/A           N/A         N/A</td> <td>Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000</td> <td>(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100</td> <td>Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG</td>	Options     O       cutuated relative     Port Name       ck_out1     ck_out2       ck_out2     ck_out3       ck_out4     ck_out4       ck_out5     ck_out7	to the active input           Output Freq (M           Requested           100.000           200.000           125.000           100.000           100.000           100.000           100.000           100.000           100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 N/A N/A N/A	Phase (degr Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000	Actual           ©         0.000           ©         0.000           ©         0.000           N/A         N/A           N/A         N/A	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Iocking Wizard (6.0) Documentation IP Location IP Symbol Resource Show disabled ports		Board Clockin The phase is ca Output Clock I dic_out1 I dic_out2 I dic_out3 I dic_out5 I dic_out5 I dic_out5 I dic_out7	Ig Options C culated relative Port Name dk_out1 dk_out2 dk_out3 dk_out3 dk_out4 dk_out4 dk_out5 dk_out6 dk_out7	to the active input Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000 100.000 100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 N/A N/A N/A N/A Ocking Feedback	Phase (degr Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000	Actual           0.000           0.000           0.000           N/A           N/A           N/A	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Iocking Wizard (6.0) Documentation P Location IP Symbol Resource Show disabled ports reset	clk_out2 🗕	Board     Clocklin       The phase is ca     Output Clock       If dis_out1     If dis_out2       If dis_out2     If cout3       If cout3     If cout3       If cout5     If cout5	Ig Options C Cultated relative Port Name Cl_out1 Cl_out1 Cl_out1 Cl_out1 Cl_out3 Cl_o	to the active input           Output Freq (M           Requested           100.000           200.000           125.000           100.000           100.000           100.000           100.000           100.000           100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 NNA NNA NNA NNA Source	Phase (dogr           Requested         0.000           0.000         0.000           0.000         0.000           0.000         0.000           0.000         0.000           0.000         0.000	Actual           0.000           0.000           0.000           N/A           N/A           N/A           Signal	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Iocking Wizard (6.0) Documentation P Location IP Symbol Resource Show disabled ports reset	clk_out2 🗕 clk_out3 🗕	Board Clockin The phase is ca Output Clock Ø dk_out1 Ø dk_out2 Ø dk_out3 dk_out4 Ø dk_out4 Ø dk_out4 Ø dk_out7 USE CLOC Output Cloc	Ig Options C culated relative Port Name ck_out1 ck_out1 ck_out2 ck_out3 ck_out4 ck_out3 ck_out4 ck_out5 ck_out4 ck_out5 ck_out7 ck_out6 ck_out7 ck_out	to the active input Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000 100.000 100.000	clock. IHz)	Actual 100.0000 200.0000 100.0000 N/A N/A N/A N/A N/A N/A Source © Autor	Phase (dogr           Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000	Actual           0.000           0.000           0.000           0.000           N/A           N/A           N/A           N/A           N/A           Chip	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Iocking Wizard (6.0) Documentation P Location IP Symbol Resource Show disabled ports reset	clk_out2 🗕 clk_out3 🗕	Board Clocki The phase is ca Output Clock Ø dk_out2 Ø dk_out3 @ dk_out3 @ dk_out4 @ dk_out5 @ dk_out7 USE CLOC Output Clock @ dk_out7	C C C C C C C C C C C C C C C C C C C	to the active input Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000 100.000 100.000	clock. IHz)	Actual 100.00000 200.00000 N/A N/A N/A N/A Source © Autor Q Autor	Phase (dogr           Requested           0.000	Асца)           Долово         0.000           Долово         0.000	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Iocking Wizard (6.0) Documentation P Location IP Symbol Resource Show disabled ports reset	clk_out2 🗕 clk_out3 🗕	Board Clockin The phase is ca Output Clock Ø dk_out1 Ø dk_out2 Ø dk_out3 dk_out4 Ø dk_out4 Ø dk_out4 Ø dk_out7 USE CLOC Output Cloc	Instruction         C           Cutated relative         Port Name           Cht_out1         Cht_out2           Cht_out3         Cht_out4           Cht_out4         Cht_out5           Cht_out5         Cht_out6           Cht_out6         Cht_out7	to the active input Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000 100.000 100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 NAA NAA NAA Source @ Autor Q Juser	Phase (degree           Requested           0.000	Actual           0.000         0.000           0.000         0.000           0.000         0.000           NA         NA           NA         NA           NA         NA           Chip         (Chip           Chip         (Chip           Nip         (Chip	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Iocking Wizard (6.0) Documentation P Location IP Symbol Resource Show disabled ports reset	clk_out2 🗕 clk_out3 🗕	Board Clockin The phase is ca Output Clock Control the control Control the control the control Control the control the control the control Control the control the control the control the control the control Control the control	Image         C           Cubic         Cubic           Cubic         Port Name           Choose         Choose           Choose	to the active input Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000 100.000 100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 NAA NAA NAA Source @ Autor Q Juser	Phase (dogr           Requested           0.000	Actual           0.000         0.000           0.000         0.000           0.000         0.000           NA         NA           NA         NA           NA         NA           Chip         (Chip           Chip         (Chip           Nip         (Chip	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Iocking Wizard (6.0) Documentation P Location IP Symbol Resource Show disabled ports reset	clk_out2 🗕 clk_out3 🗕	Board Clocki The phase is ca Utput Clock Column Clock Column Clock Column Clock Column Clock Column Clock Cl	ag Options ( culated relative dk_out1 dk_out2 dk_out3 dk_out3 dk_out3 dk_out4 dk_out5 dk_out5 dk_out5 dk_out5 dk_out6 dk_out6 dk_out6 1 1 1 1 1 1	to the active input Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000 100.000 100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 NAA NAA NAA Source @ Autor Q Juser	Phase (degree           Requested           0.000	Actual           0.000         0.000           0.000         0.000           0.000         0.000           NA         NA           NA         NA           NA         NA           Chip         (Chip           Chip         (Chip           Nip         (Chip	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Iocking Wizard (6.0) Documentation P Location IP Symbol Resource Show disabled ports reset	clk_out2 🗕 clk_out3 🗕	Board Clocki The phase is ca Output Clock Ø dk_out1 Ø dk_out2 Ø dk_out3 dk_out4 Ø dk_out4 Ø dk_out4 Ø dk_out5 Ø dk_out7 Output Cloc Ø dk_out7 Ø dk_out6 Ø dk_out7 Ø dk_out6 Ø dk_out7 Ø dk_out7	ag Options ( culated relative dk_out1 dk_out2 dk_out3 dk_out3 dk_out3 dk_out4 dk_out5 dk_out5 dk_out5 dk_out5 dk_out6 dk_out6 dk_out6 1 1 1 1 1 1	to the active input Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000 100.000 100.000	clock. IHz)	Actual 100.00000 200.00000 100.00000 NAA NAA NAA Source @ Autor Q Juser	Phase (degree           Requested           0.000	Actual           0.000         0.000           0.000         0.000           0.000         0.000           NA         NA           NA         NA           NA         NA           Chip         (Chip           Chip         (Chip           Nip         (Chip	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Iocking Wizard (6.0) Documentation P Location IP Symbol Resource Show disabled ports	clk_out2 🗕 clk_out3 🗕	Board     Clockit       The phase is co     Output Clock       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the control       Image: state of the control     Image: state of the	Image Opplions         C           Called at relative         Relative           Called at	to the active input Output Freq (M Requested 100.000 200.000 125.000 100.000 100.000 100.000 100.000 100.000	clock. IHz)	Actual 100.0000 200.0000 100.0000 NMA NMA NMA NMA NMA Source @ Autor _ User _ User	Phase (degree           Requested           0.000	Actual           0.000         0.000           0.000         0.000           0.000         0.000           NA         NA           NA         NA           NA         NA           Chip         (Chip           Chip         (Chip           Nip         (Chip	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	(%) Actual 50.0 50.0 50.0 50.0 100 100 100 100 100	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG

Click **OK** to customize the IP.

**Step 5:** Click the **Board** tab. The default peripherals available for the Elbert S7 board will be displayed.



Drag and drop **DDR3 SDRAM**, **USB UART**, and **Gigabit Ethernet PHY** into IP Canvas.

Ensure that **sys\_clk\_i** of Memory Interface Generator is connected to **clk\_out2**.

Step 6: Click Run Connection Automation and select all.



#### Step 7:

#### **Run Block Automation**

• for AXI Ethernet and select "FIFO" for the AXI Streaming interface.

🐂 Numato Lab°
<b>REV: V1.0</b>

Run Block Automation		×
Automatically make connections in your design by c	hecking the boxes of the blocks to connect. Select a block on the left to display its configuration options on the right.	2
<ul> <li>✓ All Automation (2 out of 2 selected)</li> <li>✓ ♥ # axi_ethernet_0</li> <li>✓ ♥ microblaze_0</li> </ul>	AXI Ethernet connection automation generates DMA or FIFO for TX and RX streaming interfaces of instance "axi_ethernet_0". GTX clock will be connected to 125MHz source /clk_wiz_1/clk_out3. REF clock will be connected to 200.00MHz source /clk_wiz_1/clk_out2. These clock connections are for Physical interface type "RGMII". <b>Ntions</b> Physical Interface Selection Connect AXI Streaming Interfaces to FIFO	
<b>?</b>	OK Cancel	

• for Microblaze\_o select "Keep Classic MicroBlaze" option and click "OK"

Q   ጟ   ♦	Description
<ul> <li>All Automation (2 out of 2 selected)</li> <li>axi_ethernet_0</li> <li># microblaze_0</li> </ul>	MicroBlaze conversion automation converts classic MicroBlaze processors to the RISC-V <i>MicroBlaze V</i> processor. The corresponding MicroBlaze Debug Module (MDM) is also converted. Information about the options can be found in the tooltips.
	Options
	Keep Classic MicroBlaze Compressed Instructions

**Step 8:** Click **Run Connection Automation**. Select the **All Automation** option and click **OK**.

Run Connection Automation		×
Automatically make connections in your design the right.	by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on	2
Q X All Automation (1 out of 1 selected) ∨ ♥ ♣ axi_ethemet_0_fifo ♥ ⊕ \$_AXI	Select an interface pin on the left panel to view its options	
(?)	ок	Cancel

Step 9: Add AXI Timer into IP Canvas and click Run Connection Automation.

Q X + Constant Const	Description Connect Slave interface (/axi_timer_0 Options	/S_AXI) to a selected Master address space.
	Master interface Bridge IP Clock source for driving Bridge IP Clock source for Slave interface Clock source for Master interface	/microblaze_0 (Periph) v /microblaze_0_axi_periph v /clk_wiz_1/clk_out1 (100 MHz) v /clk_wiz_1/clk_out1 (100 MHz) v

Step 10: Customize the Concat IP block as shown below.

cumentation 📄 IP Location		
Show disabled ports	Component Name microblaze_0_xiconcat	
	Number of Ports 5 (1 - 128)	
	мито In0 Width 1 [1 - 4096]	
	M/TO In1 Width 1 [1 - 4096]	
	Auro         In2 Width         1         [1 - 4096]           Auro         In3 Width         1         [1 - 4096]	
	wrro         In3 Width         1         [1 - 4096]           wrro         In4 Width         1         [1 - 4096]	
In2[0:0] dout[4:0] In3[0:0] In4[0:0]		

Route the following connections to the inputs of the Concat block:

- **interrupt** on AXI Uartlite block
- **interrupt** on AXI Timer block
- interrupt on AXI-Stream FIFO
- interrupt and mac\_irq on AXI 1G/2.5G Ethernet Subsystem



Make sure that the final design looks as shown above.

**Step 11:** Select the Validate Design option from the Tools menu to ensure that connections are correct.

Tool	s Rep <u>o</u> rts	<u>W</u> indow	Layout	View	Hel
Y	<u>V</u> alidate Desi	gn		F6	6
ž	Create and Pa Create Interfa <u>R</u> un Tcl Script Property Edito Associate ELJ Generate Mer Compile Simu	ce Definition t or E Files nory Config <u>u</u>	 gration File.		trl+J
Q	Vivado Store Custom Com Launch Vitis I Language Te	mands DE			Þ
ø	Settings				

**Step 12:** In the **Sources window**, right-click on the design and select **Create HDL Wrapper**. Click **OK** in the dialog box that appears.

В	LOCK DE SIGN - Ethernet *		
Properties	Sources × Design 5 Q ≍ ≑ + ?	Signals Board ? _ C	
	✓		
Source	> Constraints	Source Node Properties Ctrl+E	
	> 🗅 Utility Sources	Create HDL Wrapper View Instantiation Template	
		Generate Output Products Reset Output Products	

**Step 13:** Click **Generate Bitstream** under the **PROGRAM AND DEBUG** section of Vivado to synthesize, implement and generate the bitstream.

~	IMPLEMENTATION	
	Run Implementation	
	> Open Implemented Design	
~	PROGRAM AND DEBUG	
	Senerate Bitstream	
	> Open Hardware Manager	

**Step 14:** After generating the bitstream successfully, select **Export -> Export Hardware** from the **File menu**. Click **Next**.



Select the "include bitstream" checkbox and click Next.

Export Hardware Platform	×	
Output Set the platform properties to inform downstream tools of the intended use of the target platform's hardware design.		
<ul> <li>Pre-synthesis         This platform includes a hardware specification for downstream software tools.     </li> <li>Include bitstream         This platform includes the complete hardware implementation and bitstream, in addition to the hardware specification for     </li> </ul>		
software tools.		
< <u>Back</u> <u>Next&gt;</u> EinishCan	cel	

Provide the **XSA file name** and save it at a suitable **location**. Click **Next** and click **Finish** in the next dialog box.

Export Hardwa	re Platform	×
Files Enter the name o	f your hardware platform file, and the directory where the XSA file will be stored.	Σ
Export to:	Ethernet_wrapper C:/projects/Elbert_S7/Ethernet The XSA will be written to: C:\projects\Elbert_S7\Ethernet\Ethernet_wrapper.xsa	8
	<back einish<="" td=""><td>Cancel</td></back>	Cancel

**Step 15:** Launch Vitis classic.

**Note:** In Vivado 2024.1, accessing Vitis via the tools menu inadvertently launches Vitis Unified instead of Vitis Classic, which is our preferred tool for project creation. To utilize Vitis Classic, it is necessary to launch it separately.

**Step 16:** In Vitis, IDE window select **Create Application Project** and click **Next** in the dialog box that appears.

A New Application Project			×
Create a New Application Project			•••
This wizard will guide you through the 4 steps of creating new application projects. 1. Choose a <b>platform</b> or create a <b>platform project</b> from Vivado exported XSA 2. Put application project in a <b>system project</b> , associate it with a processor 3. Prepare the application runtime – <b>domain</b> 4. Choose a template for application to quick start development			
Processor Domain App XSA			
A platform provides hardware information and software environment settings.     Skip welcome page next time. (Can be reached with Back button)	)		
(7) < Back Next > Finish		Cance	el l

In the **Platform**, window select **Create a new platform from the hardware** Tab and import the **XSA file** which is already created (Provide XSA file location). Click **Next**.

	t a platform from repository	tform from hardware (XSA)			
Har	vare Specification C:\projects\Elbert_S7\Ethernet\Ethernet_wrappe	r.xsa			
	vck190				
	vmk180 zc702				
XSA	le: zc706 zcu102			Browse	
	zcu106 zed				
	C:\projects\Elbert_S7\Ethernet\Ethernet_wrapper	r.xsa			
Platf	m name: Ethernet_wrapper				

In the **Application Project Details** window, give an appropriate **name** for the Vitis Project and click **Next**. Click **Next** in the **Domain** window.

Select the **lwIP Echo Server** template from the list of available templates and click **Finish**.

Templates		
Select a template to create your project.		L
Available Templates:		
Find:	⊜ ⊞	lwIP Echo Server
Embedded software development temp Dhrystone Empty Application (C++) Empty Application(C) Hello World MIP Echo Server IwIP TCP Perf Client IwIP TCP Perf Client IwIP UDP Perf Client IwIP UDP Perf Server Memory Tests OpenAMP echo-test OpenAMP entrix multiplication Dem OpenAMP RPC Demo Peripheral Tests RSA Authentication App Zyng DRAM tests Zymg ESBI		The lwIP Echo Server application provides a simple demonstration of how to use the light-weight IP stack (lwIP). This application sets up the board to use IP address 192.168.1.10 or IPv6 FE80:0:0:0:20A:35FF:FE00:102, with MAC address 00:0a:35:00:01:02. The server listens for input at port 7 and simply echoes back whatever data is sent to that port.

#### Step 17: Select Navigate to BSP Settings from Application Project Settings.

Runtime:	nernet nernet wrapper.	Options View current BSP settings, or configure settings like STDIO peripheral selection, compiler flags, SW intrusive profiling add/remove libraries, assign drivers to peripherals, change versions of OS/libraries/drivers etc. Navigate to BSP Settings
CPU: r OS: s	andalone_microblaze_0 nicroblaze_0 andalone ations: View processors, memory ranges and peripherals.	

Select Board Support Package and click on Modify BSP Settings option.

View current BSP settings, or configure s				
peripherals, change versions of OS/libra Modify BSP Settings Reset BSP Sou	ries/drivers etc.	ral selection, compiler flags, SW intru	isive profiling, add/remove libr	aries, assign drivers to
existing modifications done. All the sub			ngs, click the below link. This o	peration clears any
Operating System				
Version: 9.1 Description: Standalone is a simpl as well as the basic fe Documentation: - Drivers Libraries	e, low-level software laye atures of a hosted enviro	. It provides access to basic processo ment, such as standard input and ou	features such as caches, intern tput, profiling, abort and exit.	upts and exceptions
Name	Driver	Documentation	Examples	
axi ethernet 0	axiethernet	Documentation Link	Import Examples	
		Documentation Link	Import Examples	
	tmrctr	Documentation Link	Import Examples	
	uartlite	Documentation Link	Import Examples	
microblaze_0_axi_intc	intc	Documentation Link	Import Examples	
microblaze_0_local_memory_dlm	bram	Documentation Link	Import Examples	
microblaze_0_local_memory_ilmb	bram	Documentation Link	Import Examples	
mig_7series_0	mig_7series	Documentation Link	-	
	A BSP settings file is generated with the existing modifications done. All the sub: Load BSP settings from file Operating System Name: standalone Version: 9.1 Description: as well as the basic fe Documentation: - Drivers: Libraries Name axi, ethernet_0 axi, ethernet_0 file axi, ethernet_0 axi, ethernet_0 axi, ethernet_0 axi, ethernet_0 axi, ethernet_0 axi, ethernet_0 microblaze_0_local_memory.imb	A BSP settings file is generated with the user options selected in the existing modifications done. All the subsquent changes are applie Load BSP settings from file Operating System           Name: standalone           Name: standalone           Version: 9.1           Description:           Standalone is a simple, low-level software layer           Description:           Drivers           Libraries           Name           axi_ethernet_0           below           below            bel	A BSP settings file is generated with the user options selected in the settings dialog. To use existing setting indiffications done. All the subsquent changes are applied on top of the loaded settings. Load BSP settings from file Operating System Name: standalone Version: 9.1 Description: Standalone is a simple, low-level software layer. It provides access to basic processor as well as the basic features of a hosted environment, such as standard input and ou Documentation: - Drivers Libraries Name axi_ethernet_0 builtifio Documentation Link axi_utrific_0 uurtite Documentation Link axi_utrific_0 microblaze_0_local_memory_llmb bram Documentation Link Docum	A BSP settings file is generated with the user options selected in the settings dialog. To use existing settings, click the below link. This of existing modifications done. All the subsquent changes are applied on top of the loaded settings. Load SSP settings from file Operating System           Operating System           Name: standalone           Version: 9.1           Description:           Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, intern swell as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.           Documentation: -           Drivers         Libraries           Name         priver           Documentation Link         Import Examples           avi_ethernet_0         aviethernet           avi_ethernet_0         timetre           avi_ethernet_0         timetre           avi_ethernet_0         timetre           avi_ethernet_0         aviethernet           avi_ethernet_0         timetre           avi_ethernet_0         uaritite           avi_ethernet_0         uaritite           avi_ethernet_0         uaritite           avi_ethernet_0         uaritite           avi_ethernet_0         uaritite           avi_ethernet_0.0         uaritite           avi_ethernet_0.1         import Examples

In the Board Support Package Settings window, select lwip (Iwip220) library, change the **dhcp\_options** to "**false**" and ensure that "debug options" are "false".

# Select **phy\_link\_speed** in **temac\_adapter\_options** as

CONFIG\_LINKSPEED1000.

	age Settings				1
Control various settings	of your Board Support Package.				L
✓ Overview					
✓ standalone	Configuration for library: lwip220				
lwip220	Name	Value	Default	Turne	Description
✓ drivers				Туре	
microblaze_0	api_mode	RAW API (RAW_API)	RAW_API false	enum boolean	Mode of operation for IwIP (R
	lwip_tcp_keepalive no_sys_no_timers	false	true	boolean	Enable keepalive processing v
	sgmii_fixed_link	true false	false	boolean	Drops support for sys_timeou Enable fixed link for GEM SGN
	socket_mode_thread_prio	2	2	integer	Priority of threads in socket m
	use_axieth_on_zynq	1	1	integer	Option if set to 1 ensures axie
	use_emaclite_on_zynq	1	1	integer	Option if set to 1 ensures ema
	> arp_options	true	true	boolean	ARP Options
	> debug_options	true	true	boolean	Turn on IwlP Debug?
	✓ dhcp_options	true	true	boolean	Is DHCP required?
	lwip_dhcp	false	false	boolean	Is DHCP required?
	lwip_dhcp_does_acd_check	false	false	boolean	ACD check on offered addres
	> icmp_options	true	true	boolean	ICMP Options
	igmp_options	false	false	boolean	IGMP Options
	> lwip_ip_options	true	true	boolean	IP Options
	> ipv6_enable	false	false	boolean	IPv6 enable value
	> lwip_memory_options				Options controlling lwIP men
	> mbox_options	true	true	boolean	Mbox Options
	> pbuf_options	true	true	boolean	Pbuf Options
	> stats_options	true	true	boolean	Turn on IwIP statistics?
	> tcp_options	true	true	boolean	Is TCP required ?
	> temac_adapter_options	true	true	boolean	Settings for xps-II-temac/Axi- Is UDP required ?
	> udp_options	true	true	boolean	is ODP required ?
? 					OK Cancel
Board Support Packag					
Board Support Packag			-		
Board Support Packag Board Support Pack Control various settings c	age Settings				
Board Support Packag	age Settings				
Board Support Packag Board Support Pack Control various settings c V Overview	age Settings If your Board Support Package. Configuration for library: Iwip220	Value	Default	Tune	
<ul> <li>Board Support Packag</li> <li>Board Support Pack.</li> <li>Control various settings of</li> <li>Overview</li> <li>standalone</li> </ul>	age Settings If your Board Support Package. Configuration for library: Iwip220 Name	Value	Default	Туре	Description
Board Support Packag Board Support Pack Control various settings c Overview v standalone Iwip220	age Settings of your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio	2	2	integer	Description Priority of threads in socket r
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq	2 1	2 1	integer integer	Description Priority of threads in socket r Option if set to 1 ensures axi
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings of your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq	2 1 1	2 1 1	integer integer integer	Description Priority of threads in socket I Option if set to 1 ensures axi Option if set to 1 ensures em
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq > arp_options	2 1 1 true	2 1 1 true	integer integer integer boolean	Description Priority of threads in socket Option if set to 1 ensures axi Option if set to 1 ensures em ARP Options
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_emaclice_on_zynq use_emaclice_on_zynq arp_options > debug_options	2 1 1 true true	2 1 1 true true	integer integer integer boolean boolean	Description Priority of threads in socket Option if set to 1 ensures axi Options Turn on IwIP Debug?
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq use_emaclite_on_zynq ) arp_options ) debug_options ) dhcp_options	2 1 1 true true true	2 1 1 true true true	integer integer boolean boolean boolean	Description Priority of threads in socket Option if set to 1 ensures axi Option if set to 1 ensures en ARP Options Turn on IwIP Debug? Is DHCP required?
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq arp_options > debug_options > dhcp_options > icmp_options	2 1 true true true true	2 1 1 true true true true	integer integer boolean boolean boolean boolean	Description Priority of threads in socket 1 Option if set to 1 ensures axi Option if set to 1 ensures en ARP Options Turn on IN/P Debug? Is DHCP required? ICMP Options
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_maclite_on_zynq arp_options debug_options debug_options dicp_options igmp_options	2 1 1 true true true	2 1 1 true true true	integer integer boolean boolean boolean	Description Priority of threads in socket Option if set to 1 ensures axi Option if set to 1 ensures en ARP Options Turn on IwIP Debug? Is DHCP required?
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq arp_options > debug_options > dhcp_options > icmp_options	2 1 1 true true true true false	2 1 1 true true true true false	integer integer boolean boolean boolean boolean boolean	Description Priority of threads in socket Option if set to 1 ensures axi Option if set to 1 ensures en ARP Options Turn on IwIP Debug? Is DHCP required? ICMP Options IGMP Options IP Options
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq > arp_options > debug_options > debug_options > debug_options > icmp_options > lwip_ip_options > lwip_ip_options	2 1 1 true true true true true true true true true	2 1 1 true true true true true true true true	integer integer boolean boolean boolean boolean boolean boolean	Description Priority of threads in socket I Option if set to 1 ensures axi Options Turn on IwIP Debug? Is DHCP required? ICMP Options IGMP Options IP Options IP Options IP Options
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings of your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclice_on_zynq arp_options > debug_options > debug_options > icmp_options > icmp_options > icmp_options > ipo_options > ipo_options > ipo_options > ipo_options	2 1 1 true true true true true true true true true	2 1 1 true true true true true true true true	integer integer boolean boolean boolean boolean boolean boolean	Description Priority of threads in socket I Option if set to 1 ensures axi Options Turn on IwIP Debug? Is DHCP required? ICMP Options IGMP Options IP Options IP Options IP Options
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings of your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq use_emaclite_on_zynq use_emaclite_on_zynq office_on_zynq setup_options debug_options debug_options debug_options imp_options lwip_ip_options lwip_ip_options lwip_ip_options lwip_imemory_options	2 1 1 true true true false true false	2 1 1 true true true false true false	integer integer boolean boolean boolean boolean boolean boolean	Description Priority of threads in socket I Option if set to 1 ensures axi Option if set to 1 ensures en ARP Options Turn on IwIP Debug? Is DHCP required? ICMP Options IGMP Options IPV6 enable value Options controlling IwIP me
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_maclite_on_zynq arp_options debug_options debug_options debug_options debug_options jimp_options jimp_options jivp_enable jivf_enable jivf_enable jivf_enable	2 1 1 true	2 1 1 true true true true false true	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean	Description Priority of threads in socket I Option if set to 1 ensures axi Option if set to 1 ensures en ARP Options Turn on IwIP Debug? Is DHCP required? IGMP Options IGMP Options IGMP Options IP Options IP Options IP Options IP Options IP Options IP Options
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_amaclite_on_zynq > arp_options > debug_options > debug_options > dhcp_options > icmp_options > lwip_ip_options > lwip_ip_options > lwip_enable > lwip_options > ipvo_enable > lwip_options > mbox_options > mbox_options > pbuf_options	2 1 1 true true true true false true	2 1 1 true true true true false true	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean	Description Priority of threads in socket Option if set to 1 ensures axi Option if set to 1 ensures axi Options Turn on IwIP Debug? Is DHCP required? ICMP Options IGMP Options IP Options IP Options IP Options IP Options IP Options Pbuf Options Pbuf Options
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq use_emaclite_on_zynq debug_options debug_options debug_options debug_options igmp_options ligmp_options ligmp_options ligv6_enable livip_inemory_options mbox_options buit_options buit_options buit_options buit_options buit_options buit_options buit_options	2 1 1 1 true true true true true true false true true true true true true true tru	2 1 1 true true true true false true false true	integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean	Description Priority of threads in socket I Option if set to 1 ensures axi Option if set to 1 ensures en ARP Options Turn on IwIP Debug? Is DHCP required? ICMP Options IGMP Options IP Options IS TUR on IwIP statistics? Is TCP required ?
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings of your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq use_emaclite_on_zynq arp_options > debug_options > debug_options > icmp_options > icmp_options > imp_options > lwip_ip_options > lwip_ip_options > lwip_memory_options > mbox_options > pbuf_options > stats_options > tcp_options	2 1 1 1 true true true true false true true true true true true true tru	2 1 1 true true true true false true false true	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean	Description Priority of threads in socket I Option if set to 1 ensures axi Option if set to 1 ensures en ARP Options Turn on IwIP Debug? Is DHCP required? ICMP Options IGMP Options IGMP Options IP Options IP Options IP Options IP Options IP Options Pbuf Options Turn on IwIP statistics? Is TCP required ? Settings for xps-II-temac/Axi
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_maclite_on_zynq arp_options > debug_options > debug_options > debug_options > debug_options > icmp_options > imp_options > lwip_ip_options > iwp_options > iwp_options > iwp_options > iwp_options > iwp_options > iwp_options > jwof_enable > lwip_ip_options > mbox_options > pbuf_options > stat_options > tcp_options > tcp_options	2 1 1 true true true true false true true true true true true true tru	2 1 1 true true true true false true	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean	Description Priority of threads in socket Option if set to 1 ensures axi Option if set to 1 ensures axi Option if set to 1 ensures en ARP Options Turn on twIP Debug? Is DHCP required? ICMP Options IGMP Options IGMP Options IP Options ITurn on twIP statistics? Is TCP required ? Settings for xps-II-temac/Axi Zyng Ethernet Interface num
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq ap_options > debug_options > debug_options > debug_options > icmp_options > icmp_options > lwip_ip_options > lwip_ip_options > lwip_options > lwip_options > lwip_options > byoe_enable > lwip_options > pbuf_options > stats_options > tcp_options > tcp_options > tcp_options > tcp_options > tcp_options > tcp_options > tcp_options	2 1 1 1 true true true true false true true true false true true true true true true true tru	2 1 1 true true true true false true false true true true true true true true true true true true true true true true true false true true true true false true true true false true true true false true true true false true true true false true true true false true true true false true true true true false true fal 64	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean	Description Priority of threads in socket Option if set to 1 ensures arri ARP Options Turn on IwIP Debug? Is DHCP required? ICMP Options IGMP Options IP Options IS TCP required? Is TCP required? Is TCP required? Is TCP required? Is TCP required? Settings for xps-II-temac/Arri Zyng Ethernet Interface num Setting for RX Interrupt coald
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq use_emaclite_on_zynq arp_options > debug_options > debug_options > debug_options > debug_options > debug_options > debug_options > icmp_options > lwip_ip_options > lwip_ip_options > ipv6_enable > lwip_options > mbxo_options > stats_options > tcp_options > tcp_options > tcp_options > tcmac_adapter_options emac_number n_nx_coalesce	2 1 1 true true true false true true true true true true true true true true true true true true true true false true true true true true true true false true true true true true true true false true true true true true true true false true	2 1 1 true true true false true false true 1 1 1 1 1 1 1 1 1 1 1 1 1	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean integer integer	Description Priority of threads in socket Option if set to 1 ensures ari Option if set to 1 ensures en APP Options Turn on IwIP Debug? Is DHCP required? ICMP Options IBMP Options IP Options IP Options IP Options IP Options IP Options IP Options Turn on IwIP statistics? Is TCP required ? Settings for xps-II-temac/Axi Zynq Ethernet Interface num Setting for RX Interrupt coal Number of RX Buffer Descrip
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieh_on_zynq use_emaclite_on_zynq ap_options > debug_options > debug_options > debug_options > debug_options > icmp_options > icmp_options > lwip.jp_options > lwip.jp_options > lwip.options > lwip.options > byoc_enable > lwip.options > pbuf_options > stats_options > tcp_options > tcp_options > tcp_options > tcp_options > tcp_options > tcp_options > tcp_options > tcma_adapter_options emac_number n_tx_coalesce n_tx_descriptors	2 1 1 1 true true true false true false true true false true true true true true true true tru	2 1 1 true true true false true false true	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean integer integer integer integer	Description         Priority of threads in socket         Option if set to 1 ensures axi         Option if set to 1 ensures axi         Option if set to 1 ensures axi         Option if set to 1 ensures en         ARP Options         Turn on lwIP Debug?         Is DHCP required?         ICMP Options         IGMP Options         IP Options         IP Options         IP Options         IP Options         IP Options         IP Options         Turn on lwIP statistics?         Is TCP required ?         Settings for xps-II-temac/Axi         Zynq Ethernet Interface num         Setting for X Interrupt coale         Number of RX Buffer Descrip         Setting for X Interrupt coale         Number of TX Buffer Descrip
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq ap_options > debug_options > debug_options > debug_options > debug_options > debug_options > imp_options > lwip_ip_options > lwip_ip_options > lwip_options > mbx_options > stats_options > stats_options > temac_adapter_options = emac_number n_tx_coalesce n_tx_descriptors phy_link_speed	2 1 1 1 true true true false true false true true true true true true true tru	2 1 1 true true true false true false true false true true true true true false true	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean integer integer integer integer integer	Description Priority of threads in socket Option if set to 1 ensures axi Option if set to 1 ensures axi Option if set to 1 ensures axi Options Turn on IwIP Debug? Is DHCP required? ICMP Options IPO options IPO options IPO options IPO options IPO options Turn on IwIP statistics? Is TCP required ? Settings for xps-II-temac/Axi Zyng Ethernet Interface num Setting for TX Interrupt coald Number of TX Buffer Descrip Setting for TX Interrupt coald Number of TX Buffer Descrip Ink speed as negotiated by to
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_anieth_on_zynq use_emacle_on_zynq use_emacle_on_zynq use_emacle_on_zynq debug_options > debug_options > debug_options > debug_options > debug_options > debug_options > debug_options > debug_options > debug_options > hwip_ip_options > lwip_ip_options > lwip_ip_options > lwip_ip_options > blyof_enable > lwip_memory_options > pbuf_options > temac_adapter_options = emac_number n_rx_coalesce n_tx_descriptors pty_link_speed - up_ip_ix_creecksum_onroad	2 1 1 1 true true true true false true true true false true true true true true true true tru	2 1 1 true true true false true false true t	integer integer boolean boolea	Description Priority of threads in socket r Option if set to 1 ensures and APP Options Turn on IwIP Debug? Is DHCP required? ICMP Options IPO Options IPO Options IPO options IPO options IPO options IPO options IPO options Turn on IwIP statistics? Is TCP required ? Settings for xps-II-temac/Aci Zynq Ethemet Interface num Setting for XX Interrupt coale Number of RX Buffer Descrip Setting for TX Interrupt coale Number of TX Buffer Descrip Setting for TX Interrupt coale Number of TX Buffer Descrip Setting for XX Interrupt coale Number of TX Buffer Descrip
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	Age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq arp_options debug_options debug_options debug_options debug_options debug_options bitrop_options piv6_enable piv6_enable piv6_enable bivip_memory_options piv6_options piv6_options bitrop_options bitrop_options bitrop_options bitrop_options come_n_nc_descriptors n_tc_coalesce n_tc_descriptors n_tc_coalesce n_tc_descriptors n_tc_coalesce n_tc_descriptors use_int_coalesce use_int_coalesce use_int_coalesce in	2 1 1 1 true true true true true false true true true true true true true tru	2 1 1 true true true false true false true	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean integer integer integer integer integer integer boolean	Description Priority of threads in socket 1 Option if set to 1 ensures axi Option if set to 1 ensures axi Option if set to 1 ensures em ARP Options Turn on IwIP Debug? Is DHCP required? IGMP Options IGMP Options IPV6 enable value Option controlling IwIP me Mbox Options Turn on IwIP statistics? Is TCP required ? Settings for xps-II-temac/Axi Zyng Ethernet Interface num Setting for RX Interrupt coale Number of TX Buffer Descrip Setting for TX Interrupt coale Number of TX Buffer Descrip Setting for TX Interrupt coale Number of TX Buffer Descrip Setting for TX Interrupt coale Number of TX Buffer Descrip Setting for TA Interrupt coale Number of TX Buffer Descrip Setting for TA Interrupt coale Number of TA Buffer Descrip
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	Arge Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieh_on_zynq use_emaclite_on_zynq arg_options > debug_options > debug_options > debug_options > debug_options > icmp_options > inp_options > lwip_ip_options > lwip_ip_options > lwip_options > hwip_options > byof_options > pbuf_options > stats_options > tcp_options > tcp_options > tcp_options > tcp_options > tcp_options > tcp_options > tcp_options = mac_number n_tx_coalesce n_tx_descriptors n_tc_options = tcp_ip_tc_checksum_offload	2 1 1 1 true true true true false true true true true true true true tru	2 1 1 true true true false true false true false true	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean integer integer integer integer integer boolean	Description         Priority of threads in socket I         Option if set to 1 ensures axi         Option if set to 1 ensures em         ARP Options         Turn on lwIP Debug?         Is DHCP required?         IGMP Options         IGMP Options         IGMP Options         IP Options         IP Options         IP Options         IP Options         ISTCP required?         Is TCP required?         Settings for xps-II-temac/Axi         Zynq Ethernet Interface num         Setting for X Interrupt coale         Number of TX Buffer Descrip         Ink speed as negotiated by t         Offload TCP and IP Transmit         Offload TCP Receive checksts
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq ap_options > debug_options > debug_options > debug_options > debug_options > debug_options > drop_options > imp_options > lwip_ip_options > lwip_ip_options > imp_options > bwip_enable > lwip_options > stats_options > stats_options > tcp_options > tcp_options > tcp_options > tcp_options = emac_number n_tx_coalesce n_tx_descriptors n_tc_options phy_link_speed tcp_ip_tx_checksum_offload tcp_tx_checksum_offload	2 1 1 1 1 true true true true true false true true true true true true true tru	2 1 1 true true true true false true tr	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean integer	Description Priority of threads in socket I Option if set to 1 ensures avi Option if set to 1 ensures em ARP Options Turn on twIP Debug? Is DHCP required? ICMP Options IGMP Options IGMP Options IGMP Options IP Options IP Options IP Options IF Options Turn on twIP statistics? Is TCP required ? Settings for xps-II-temac/Axi Zyng Ethernet Interface num Setting for RX Interrupt coale Number of RX Buffer Descrip Setting for TX Interrupt coale Number of TX Buffer Descrip Ink speed as negotiated by t Offload TCP and IP Transmit Offload TCP Receive e Offload TCP Transmit checks Offload TCP Transmit checks
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_anieth_on_zynq use_emaclice_on_zynq use_emaclice_on_zynq use_emaclice_on_zynq arg_options > debug_options > debug_options > debug_options > debug_options > debug_options > debug_options > debug_options > hwip_ip_options > lwip_ip_options > lwip_ip_options > lwip_ip_options > hwip_emanory_options > pbuf_options > stats_options > temac_adapter_options = emac_number n_rx_coalesce n_tx_descriptors n_tx_coalesce n_tx_descriptors phy_link_speed tcp_ip_tx_checksum_offload tcp_tx_checksum_offload tcp_tx_checksum_offload	2 1 1 1 true true true true true false true true false true true true true true true true tru	2 1 1 true true true false true false true true true true true true true true true true true false true true false true false true false true true false true false true false true false true true true true false true true true false false true true false	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean integer	Description Priority of threads in socket I Option if set to 1 ensures axi Option if set to 1 ensures axi Option if set to 1 ensures em ARP Options Turn on IwIP Debug? Is DHCP required? ICMP Options IPV6 enable value Options controlling IwIP me Mbox Options Pbuf Options Pbuf Options Turn on IwIP statistics? Is TCP required ? Settings for xps-II-temac/Axi Zynq Ethernet Interface num Setting for XX Interrupt coale Number of TX Buffer Descrip Setting for TX Interrupt coale Number of TX Buffer Descrip Ink speed as negotiated by t Offload TCP and IP Receive checkss Offload TCP ransmit checks use jumbo frames
Board Support Packag Board Support Packa Control various settings c Voerview v standalone Wwip220 v drivers	age Settings if your Board Support Package. Configuration for library: lwip220 Name socket_mode_thread_prio use_axieth_on_zynq use_emaclite_on_zynq ap_options > debug_options > debug_options > debug_options > debug_options > debug_options > drop_options > imp_options > lwip_ip_options > lwip_ip_options > imp_options > bwip_enable > lwip_options > stats_options > stats_options > tcp_options > tcp_options > tcp_options > tcp_options = emac_number n_tx_coalesce n_tx_descriptors n_tc_options phy_link_speed tcp_ip_tx_checksum_offload tcp_tx_checksum_offload	2 1 1 1 1 true true true true true false true true true true true true true tru	2 1 1 true true true true false true tr	integer integer integer boolean boolean boolean boolean boolean boolean boolean boolean boolean boolean integer	Description Priority of threads in socket I Option if set to 1 ensures avi Option if set to 1 ensures em ARP Options Turn on twIP Debug? Is DHCP required? ICMP Options IGMP Options IGMP Options IGMP Options IP Options IP Options IP Options IF Options Turn on twIP statistics? Is TCP required ? Settings for xps-II-temac/Axi Zyng Ethernet Interface num Setting for RX Interrupt coale Number of RX Buffer Descrip Setting for TX Interrupt coale Number of TX Buffer Descrip Ink speed as negotiated by t Offload TCP and IP Transmit Offload TCP Receive e Offload TCP Transmit checks Offload TCP Transmit checks

After changing the library settings, click OK. vitis will update the BSP automatically. If that didn't happen for any reason, run a build manually.

NOTE: Vitis does not include built-in support for KSZ Ethernet PHY drivers. To enable compatibility, the xaxiemacif\_physpeed file must be manually updated with the KSZ driver modifications. Replace the existing xaxiemacif\_physpeed file in your project with the provided <u>file</u>, which includes the necessary changes to support KSZ PHY drivers. This ensures proper Ethernet functionality in your application.

After modifying the xaxiemacif\_physpeed file Build the project.

**Step 18:** Once the build is completed successfully, power up Elbert S7 using an USB type C cable.

**Step 19:** Program the FPGA on Elbert S7 by selecting the **Program Device** option from the **vitis menu**.

Vitis     Project     Window     Help       Platform Repositories     Software Repositories       Software Repositories       Examples       Libraries	
🍌 Vivado Integration >	
<ul><li>XSCT Console</li><li>Vitis Shell</li></ul>	
Generate Linker Script       Senerate Device Tree	
Start/Stop Emulator	
💁 Program Device	
🔯 Create Boot Image >	
📝 🛛 Program Flash	
Dump/Restore Memory	
Export Workspace To Unified IDE	

Project:	Ethernet_system	~		
Connection:	Local		New	
		~		
Device:	Auto Detect		Select	
Bitstream/PDI:	\${project_loc:Ether	rnet}/_ide/bitstream/Ethernet_wrapper.l	Search	Browse
Partial Bitstre				
BMM/MMI File:	\${project_loc:Ether	rnet}/_ide/bitstream/Ethernet_wrapper.ı	Search	Browse
Software Config	juration			
Processor		ELF/MEM File to Initialize in Block RAM	alize in Block RAM	
microblaze_0		bootloop		
O Shin Douisi - A	2h h			
Skip Revision (	LNECK			
		Generate	Program	Cancel

Open the COM port corresponding to Elbert S7 in any serial terminal (PuTTY, Tera Term, etc.) with a 9600 baud rate. Now, right-click on the .elf file in Project Explorer and select "Launch on Hardware" as shown below.

	ject <u>W</u> indow <u>H</u> elp ≽ ▼ <b>() ▼</b>			
Explorer 🛛			👗 Ethernet_system 🛛 💥 Ethernet 🛛 🗖 Ethern	et_wrapper 🛛
↓     Ithernet_system [ Ithernet       ↓     Ithernet ( standalone_mi       ↓     Ithernet_system.spj       ↓     Ithernet_wrapper       ↓     Ithernet wrapper       ↓     Ithernet wrapper       ↓     Ithernet wrapper       ↓     Ithernet wrapper	wrapper ]	Ctrl+V Delete F5	type filter text	Board Si View current peripherals, c Modify BSP A BSP setting existing mod Load BSP set Operating Si
> 👝 logs > 👝 microblaze_0 🍋 resources 🗾 platform.spr 📄 platform.tcl	Clean Project Generate Linker Script C/C++ Build Settings Team	>		Ver Descrip Documenta Drivers Lii
	Run As	>	1 Launch Hardware (Single Application Debug)	,
	Debug As		2 Launch SW Emulator (Single Application Debug	)
	Properties	Alt+Enter	3 Launch Hardware (Single Application Debug (Gl	DB))
Assistant 🔀	- 	☆ 8 - □	Run Configurations	

Observe the details displayed on the serial terminal.



**Step 21:** Connect the Ethernet cable to the board and the other end to the PC Ethernet port. Go to **Control Panel**. Go to **Network and Internet -> Network and Sharing Centre -> Change adapter settings**. Select "Change adapter settings". Right-click on Ethernet, click properties, and select "**IPv4**". Change the IPv4 address to **192.168.1.15** (any IP address can be used) and the default gateway to **192.168.1.1**.

🔄 Network Cor	nnections	- 🗆 X
$\leftarrow \rightarrow \lor$	↑ 💘 « Network and Internet → Network Con	nections $\checkmark$ G Search Network Connections $\red P$
Organise 👻	Disable this network device Diagnose this conne	ction Rename this connection » 🗄 👻 😗
-		Network Adapter VMware Network Adapter
	Ethernet Properties	Internet Protocol Version 4 (TCP/IPv4) Properties $\qquad \qquad \qquad$
	Networking Sharing	General
×	Connect using:	You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
	Config This connection uses the following items:	Obtain an IP address automatically Use the following IP address:
	Pient for Microsoft Networks     Pient for Microsoft Networks     Piene and Printer Sharing for Microsoft Networks     Piene and Printer Sharing for Microsoft Networks     Piene Scheduler     Piene Driver	Dec dire following in address:         192.168.1.15           Subnet mask:         255.255.0           Default gateway:         192.168.1.1
	Internet Protocol Version 4 (TCP/IPv4)     Microsoft Network Adapter Multiplexor Protocol	Obtain DNS server address automatically Use the following DNS server addresses:
	Install Uninstall Proper	Preferred DNS server: 8 . 8 . 8 . 8
	Description	Alternative DNS server: 8 . 8 . 4 . 4
	Transmission Control Protocol/Internet Protocol. The def wide area network protocol that provides communication across diverse interconnected networks.	Validate settings upon exit Advanced
4 items 1	ОК	OK Cancel

**Step 22:** Open a telnet session with IP Address **192.168.1.10** (IP address as per main.c) at **port 7**, give input through the keyboard and observe the output. If you enter a character from the keyboard, you can observe the transmitted and echoed characters on telnet as shown.



## 6. SD card test

## Introduction

In this tutorial, we'll explore the process of creating an SD card test project using Vivado and Vitis Unified IDE for the Elbert S7 FPGA Development Board. Our design will feature a MicroBlaze soft processor, which will be integrated with an SD card interface via the AXI bus. This project will allow us to perform basic read and write operations on the SD card, helping to verify the functionality of the SD card interface on the Elbert S7. Although MicroBlaze designs can utilize either PLB or AXI bus systems, we'll focus on the AXI bus for this tutorial. For detailed information on MicroBlaze and additional resources, including the datasheet, please visit <u>AMD's dedicated MicroBlaze page</u>.

#### STEP 1:

To create a new Vivado project specifically for the **Elbert S**<sub>7</sub> FPGA board, follow the procedure outlined in the section "<u>Getting Started with Vivado: Creating a New</u> <u>FPGA Project.</u>"

**Step 2:** After creating a project, In the "Flow Navigator" panel, select "Create Block Design" under the IP integrator section. Give an appropriate name (Eg: "SD\_card ") to the design and click "OK ".



#### Step 3:

Go to Diagram window, right click and select "Add IP" from the popup menu. Search for "**MicroBlaze**" and add it to the design by double-clicking it.



Click "**Run Block Automation**" present in the "**Designer Assistance available**" bar on the top left corner of the window to complete the design. Select the settings as shown in the following image. Click "**OK**" for Vivado to automatically configure the blocks for you.

C	.	Description	
	✓ All Automation (1 out of 1 sele ✓ ♥ microblaze_0	Alternation         Alternation generates local memory of selected size, and caches can be configure           MicroBlaze         Debug Module, Peripheral AXI Interconnect, Interrupt Controller, a clock source, Processor Systemate added and connected as needed. A preset MicroBlaze configuration can also be selected.           Information about the options can be found in the tooltips.         Information about the options can be found in the tooltips.	
		Options Preset None V	
		Local Memory 64KB V	
		Local Memory ECC None ~	
		Cache Configuration None V	
		Debug Module Debug Only 🗸	
		Peripheral AXI Port Enabled V	
		Clock Connection New Clocking Wizard	

#### Step 4:

Double click "**Clocking Wizard**" IP and customize "**Board**" and "**Output Clocks**" settings as shown in the following image.

#### ELBERT S7 HANDBOOK

🐂 Numato Lab' REV: V1.0

ing Wizard (6.0)									
imentation 🗁 IP Location									
mentasion 🔚 IP Location									
nbol Resource	Component Name	e clk_wiz_1							
ow disabled ports	Board Clockin	g Options 0	utput Clocks MM	CM Settings S	ummary				
	Associate IP inte	rface with board	l interface						
	IP Interface CLK_IN1				Board Inte sys clock	erface			
	CLK_IN2				Custom				
	EXT_RESET_IN				Custom				
	Clear Board	Parameters							
reset clk_out1 - clk_in1 locked									
stomize IP ng Wizard (6.0)								ок	_ Ca
stomize IP Ing Wizard (6.0) mentation IP Location							(	OK	Ca
Ing Wizard (6.0) Immentation IP Location Imbol Resource	Component Nam	e cik_wiz_1						ОК	
Ing Wizard (6.0) Immentation IP Location Imbol Resource			Putput Clocks MM	CM Settings St	ummary			OK	
Ing Wizard (6.0) Immentation IP Location Imbol Resource	Board Clockir The phase is cal	ng Options C	to the active input cloo	ĸ.		223	Duty Cycle PA	OK	
Ing Wizard (6.0) Immentation IP Location Imbol Resource	Board Clockir The phase is cal Output Clock	ng Options C Iculated relative Port Name	to the active input cloc Output Freq (MHz) Requested	k. Actual	Phase (degre Requested	Actual	Duty Cycle (%) Requested	Actual	Drives
Ing Wizard (6.0) Immentation IP Location Imbol Resource	Board Clockin The phase is cal Output Clock	Port Name	to the active input close Output Freq (MHz) Requested 100.000	Actual 100.00000	Phase (degre Requested 0.000	Actual 0.000	Requested 50.000	Actual 50.0	Drives
Ing Wizard (6.0) Immentation IP Location Imbol Resource	Board Clockin The phase is cal Output Clock Cloc	Options     O     Options     O	to the active input close Output Freq (MHz) Requested 100.000	k. Actual 100.00000 32.00000	Phase (degre Requested 0.000 0.000	Actual	Requested 50.000 50.000	Actual 50.0 50.0	Drives BUFG BUFG
Ing Wizard (6.0) Immentation IP Location Imbol Resource	Board Clockin The phase is cal Output Clock Clk_out1 Clk_out2 Clk_out2	Ig Options C Iculated relative Port Name clk_out1 clk_SD clk_out3	to the active input cloo Output Freq (MHz) Requested 100.000 © 32.000 © 100.000	k. Actual 100.00000 32.00000 N/A	Phase (degre           Requested           0.000           0.000           0.000	Actual  0.000  0.000  N/A	Requested           50.000           50.000           50.000	Actual 50.0 50.0 N/A	Drives BUFG BUFG BUFG
Ing Wizard (6.0) Immentation IP Location Imbol Resource	Board     Clockin       The phase is cal       Output Clock       Image: Image of the phase is cal       Output Clock       Image of the phase is cal       Imag	Options         O           Port Name         Clk_out1           Clk_out3         Clk_out3	to the active input close Cutput Freq (MHz) Requested 100.000 32.000 100.000 100.000	Actual 100.00000 32.00000 N/A N/A	Phase (degree           Requested           0.000           0.000           0.000           0.000           0.000           0.000	Actual 0.000 0.000 N/A N/A	Requested           50.000           50.000           50.000           50.000	Actual 50.0 50.0 N/A N/A	Drives BUFG BUFG BUFG BUFG
Ing Wizard (6.0) Immentation IP Location Imbol Resource	Board Clockin The phase is cal Output Clock If alk_out1 If alk_out2 If alk_out3 If alk_out4 If alk_out5	Port Name         Classifier           clk_out1         clk_out1           clk_out3         clk_out4           clk_out4         clk_out4	to the active input cload Output Freq (MHz) Requested 100.000 32.000 100.000 100.000 100.000	k. Actual 100.00000 32.00000 N/A	Phase (degre           Requested           0.000           0.000           0.000	Actual  0.000  0.000  N/A	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000	Actual 50.0 50.0 N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
Ing Wizard (6.0) Immentation IP Location Imbol Resource	Board     Clockin       The phase is cal       Output Clock       Image: Image of the phase is cal       Output Clock       Image of the phase is cal       Imag	Options         O           Port Name         Clk_out1           Clk_out3         Clk_out3	to the active input close Cutput Freq (MHz) Requested 100.000 32.000 100.000 100.000	k. Actual 100.00000 32.00000 N/A N/A	Phase (degree           Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000	Actual 0.000 0.000 N/A N/A N/A	Requested           50.000           50.000           50.000           50.000	Actual 50.0 50.0 N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
Ing Wizard (6.0)  mentation IP Location  mbol Resource ow disabled ports	Board Clockin The phase is cal Output Clock If dk_out1 If dk_out2 If dk_out3 If dk_out4 If dk_out5 If dk_out7	Options         O           Port Name         Port Name           ck_out1         ck_out1           ck_out2         ck_out2           ck_out3         ck_out3           ck_out4         ck_out4	to the active input cloc         Output Freq (MHz)           Requested         100.000           32.000         0           100.000         100.000           100.000         100.000           100.000         100.000	Actual         100.00000           32.00000         N/A           N/A         N/A           N/A         N/A           N/A         N/A           N/A         N/A	Phase (degree           Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000	Actual           0.000           0.000           N/A           N/A           N/A           N/A	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
stomize IP ng Wizard (6.0) mentation IP Location mbol Resource ov disabled ports	Board Clockin The phase is cal Output Clock If dk_out1 If dk_out2 If dk_out3 If dk_out4 If dk_out5 If dk_out7	Port Name         Classifier           clk_out1         clk_out2           clk_out3         clk_out3           clk_out4         clk_out4           clk_out5         clk_out5	to the active input cloc         Output Freq (MHz)           Requested         100.000           32.000         0           100.000         100.000           100.000         100.000           100.000         100.000	Actual         100.0000         32.00000           32.00000         N/A         N/A           N/A         N/A         N/A           N/A         N/A         N/A	Phase (degree           Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000	Actual           0.000           0.000           N/A           N/A           N/A           N/A           N/A           N/A	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
stomize IP ng Wizard (6.0) mentation IP Location mbol Resource ow disabled ports	Board     Clockin       The phase is cal       Output Clock       Ø clk_out1       Ø clk_out2       Ø clk_out3       Ø clk_out4       Ø clk_out5       Ø clk_out6       Ø clk_out7	Image options         Image options <thimage options<="" th="">         Image opt</thimage>	to the active input cloc         Output Freq (MHz)           Requested         100.000           32.000         0           100.000         100.000           100.000         100.000           100.000         100.000	Actual           100.00000           32.0000           NA           N/A           N/A           N/A           N/A           Source	Phase (degre Requested 0.000 0.000 0.000 0.000 0.000 0.000 0.000	Actual           0.000           0.000           N/A           N/A           N/A           N/A           N/A           N/A           N/A	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG BUFG
stomize IP ing Wizard (6.0) mentation IP Location mbol Resource ow disabled ports	Board     Clockin       The phase is cal       Output Clock       Ø clk_out1       Ø clk_out2       Ø clk_out3	Image options         Image op	to the active input doc <b>Output Freq (MHz)</b> <b>Requested</b> 100.000 © 100.000 100.000 100.000 100.000 000	Actual           100.00000           32.0000           NA           NA           NA           NA           NA           NA           NA           Source	Phase (degre Requested 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000	Actual           0.000           0.000           N/A	Requested           8cuested           5c.000           5c.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
Ing Wizard (6.0)	Board     Clockin       The phase is cal       Output Clock       Ø clk_out1       Ø clk_out2       Ø clk_out3       Ø clk_out4       Ø clk_out5       Ø clk_out6       Ø clk_out7	Ig Options         Q           readed relative         Port Name           ck_out1         ck_out3           ck_out3         ck_out4           ck_out4         ck_out5           ck_out5         ck_out4           ck_out5         ck_out4           ck_out5         ck_out4           ck_out5         ck_out4           ck_out6         ck_out4	to the active input doc <b>Output Freq (MHz)</b> <b>Requested</b> 100.000 © 100.000 100.000 100.000 100.000 000	Actual           100.00000           32.00000           N/A           N/A           N/A           N/A           N/A           Source           ● Auto           ● Auto	Phase (degre Requested 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000	Actual           ©         0.000           №         0.000           №         №           №         №           №         №           №         №	Requested           50.000           50.000           50.000           50.000           50.000           50.000           50.000           50.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG
ustomize IP	Board Clockin The phase is cal Output Clock C dk_out1 C dk_out2 dk_out3 dk_out4 dk_out4 dk_out7 USE CLOC Output Clo dk_out1 Clockin Cl	Image Options         Image Op	to the active input doc <b>Output Freq (MHz)</b> <b>Requested</b> 100.000 © 100.000 100.000 100.000 100.000 000	Actual           Actual           100.00000           32.0000           N/A           N/A           N/A           N/A           N/A           Source <ul> <li>Auto</li> <li>Auto</li> <li>Use</li> </ul>	Phase (degre Requested 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000	Actual           ©         0.000           №         0.000           №         №           №         №           №         №	Requested           8cuested           5c.000           5c.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
ustomize IP  ing Wizard (6.0)  umentation IP Location  mbol Resource  row disabled ports  clk_out1  clk_out2  clk_out2	Board     Clockin       The phase is cal       Output Clock       Image: Clocking Clocking Clocking       Image: Clocking Clocking Clocking Clocking       Image: Clocking Clocking Clocking Clocking       Image: Clocking	Ig Options         Q           culated relative         entertaine           dk_out1         dk_out3           dk_out4         dk_out4           dk_out6         dk_out6           dk_out7         dk           ck         Sequencing           tk         sequencing           th         th           th	to the active input doc <b>Output Freq (MHz)</b> <b>Requested</b> 100.000 © 100.000 100.000 100.000 100.000 000	Actual           Actual           100.00000           32.0000           N/A           N/A           N/A           N/A           N/A           Source <ul> <li>Auto</li> <li>Auto</li> <li>Use</li> </ul>	Phase (degree           Requested           0.000	Actual           ©         0.000           №         0.000           №         №           №         №           №         №	Requested           8cuested           5c.000           5c.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
ustomize IP	Board     Clockin       The phase is cal       Output Clock       Image: Clocking Clocking       Image: Clocking <t< td=""><td>goptions         Q           uclated relative         entities           elk_out1         elk_out2           elk_out3         elk_out3           elk_out4         elk_out3           elk_out4         elk_out4           elk_out4         elk_out4</td><td>to the active input doc <b>Output Freq (MHz)</b> <b>Requested</b> 100.000 © 100.000 100.000 100.000 100.000 000</td><td>Actual           Actual           100.00000           32.0000           N/A           N/A           N/A           N/A           N/A           Source            <ul> <li>Auto</li> <li>Auto</li> <li>Use</li> </ul></td><td>Phase (degree           Requested           0.000</td><td>Actual           ©         0.000           №         0.000           №         №           №         №           №         №</td><td>Requested           8cuested           5c.000           5c.000</td><td>Actual 50.0 50.0 N/A N/A N/A N/A</td><td>Drives BUFG BUFG BUFG BUFG BUFG BUFG</td></t<>	goptions         Q           uclated relative         entities           elk_out1         elk_out2           elk_out3         elk_out3           elk_out4         elk_out3           elk_out4         elk_out4	to the active input doc <b>Output Freq (MHz)</b> <b>Requested</b> 100.000 © 100.000 100.000 100.000 100.000 000	Actual           Actual           100.00000           32.0000           N/A           N/A           N/A           N/A           N/A           Source <ul> <li>Auto</li> <li>Auto</li> <li>Use</li> </ul>	Phase (degree           Requested           0.000	Actual           ©         0.000           №         0.000           №         №           №         №           №         №	Requested           8cuested           5c.000           5c.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
ustomize IP	Board     Clockin       The phase is cal       Output Clock       Image: Clocking Clocking Clocking       Image: Clocking Clocking Clocking Clocking       Image: Clocking Clocking Clocking Clocking       Image: Clocking	goptions         Q           uclated relative         entities           elk_out1         elk_out2           elk_out3         elk_out3           elk_out4         elk_out3           elk_out4         elk_out4	to the active input doc <b>Output Freq (MHz)</b> <b>Requested</b> 100.000 © 100.000 100.000 100.000 100.000 000	Actual           Actual           100.00000           32.0000           N/A           N/A           N/A           N/A           N/A           Source <ul> <li>Auto</li> <li>Auto</li> <li>Use</li> </ul>	Phase (degree           Requested           0.000	Actual           ©         0.000           №         0.000           №         №           №         №           №         №	Requested           8cuested           5c.000           5c.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
ustomize IP	Board     Clockin       The phase is cal       Output Clock       Image: Clocking Clocking       Image: Clocking <t< td=""><td>Image Options         Image Options         <thimage options<="" th="">         Image Opt</thimage></td><td>to the active input doc Qutput Freq (MHz) Requested 100.000 © 100.000 100.000 100.000 Ce Number Ce Number</td><td>k. Actual 100.00000 32.00000 N/A N/A N/A N/A Ctocking Feedbac Source © Auto Quse Quse</td><td>Phase (degre Requested 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000000</td><td>Actual           ©         0.000           №         0.000           №         №           №         №           №         №</td><td>Requested           8cuested           5c.000           5c.000</td><td>Actual 50.0 50.0 N/A N/A N/A N/A</td><td>Drives BUFG BUFG BUFG BUFG BUFG BUFG</td></t<>	Image Options         Image Options <thimage options<="" th="">         Image Opt</thimage>	to the active input doc Qutput Freq (MHz) Requested 100.000 © 100.000 100.000 100.000 Ce Number Ce Number	k. Actual 100.00000 32.00000 N/A N/A N/A N/A Ctocking Feedbac Source © Auto Quse Quse	Phase (degre Requested 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000000	Actual           ©         0.000           №         0.000           №         №           №         №           №         №	Requested           8cuested           5c.000           5c.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
reset clk_out2	Board     Cockin       The phase is cal       Output Clock       Image: Clock of the phase is cal       Image: Clock of the phase is cal <t< td=""><td>Image Options         Q           value of the sector of</td><td>to the active input doc         Output Freq (MHz)         Requested         100.000         200.000         100.000</td><td>k. Actual 100.00000 32.00000 N/A N/A N/A N/A Ctocking Feedbacc Source @ Auto Quse Use</td><td>Phase (degre Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000</td><td>Actual           0.000           0.001           0.001           NA           NA</td><td>Requested           8cuested           5c.000           5c.000</td><td>Actual 50.0 50.0 N/A N/A N/A N/A</td><td>Drives BUFG BUFG BUFG BUFG BUFG BUFG</td></t<>	Image Options         Q           value of the sector of	to the active input doc         Output Freq (MHz)         Requested         100.000         200.000         100.000	k. Actual 100.00000 32.00000 N/A N/A N/A N/A Ctocking Feedbacc Source @ Auto Quse Use	Phase (degre Requested           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000           0.000	Actual           0.000           0.001           0.001           NA           NA	Requested           8cuested           5c.000           5c.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG
ustomize IP	Board     Cockin       The phase is cal       Output Clock       Image: Clock of the clock of the clock       Image: Clock of the clock of	Image Options         Q           value of the sector of	to the active input doc           Output Freq (MHz)           Requested           100.000           232.000           100.000           100.000           100.000           100.000           100.000           100.000           100.000           100.000           0000           100.000           0000 <t< td=""><td>k. Actual 100.00000 32.00000 N/A N/A N/A N/A Ctocking Feedbacc Source @ Auto Quse Use</td><td>Phase (degre Requested 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000000</td><td>Actual           0.000           0.001           0.001           NA           NA</td><td>Requested           8cuested           5c.000           5c.000</td><td>Actual 50.0 50.0 N/A N/A N/A N/A</td><td>Drives BUFG BUFG BUFG BUFG BUFG BUFG</td></t<>	k. Actual 100.00000 32.00000 N/A N/A N/A N/A Ctocking Feedbacc Source @ Auto Quse Use	Phase (degre Requested 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000000	Actual           0.000           0.001           0.001           NA           NA	Requested           8cuested           5c.000           5c.000	Actual 50.0 50.0 N/A N/A N/A N/A	Drives BUFG BUFG BUFG BUFG BUFG BUFG

## Step 5:

Run "Connection Automation" and select all the pins.



Go to the Board section, Drag and drop the USB UART from the Board section to the design.



Click on "Run Connection Automation" select all the pins and click ok.

All Automation (1 out of 1 selected)	Description Connect Slave interface (/axi_uartlite	_0/S_AXI) to a selected Master addr	ess space.
✓ ♥ ≢ axi_uartlite_0 ♥ ⊕ S_AXI	Options		
	Master interface	/microblaze_0 (Periph) 🗸	
	Bridge IP	/microblaze_0_axi_periph 🗸	
	Clock source for driving Bridge IP	/clk_wiz_1/clk_out1 (100 MHz)	~
	Clock source for Slave interface	Auto	¥
	Clock source for Master interface	/clk_wiz_1/clk_out1 (100 MHz)	~

Connect interrupt output lines from "**AXI Uartlite**" to the "**Concat**" block as shown in the below figure.



#### Step 7:

Add the SD card IP repository to Vivado IP catalog from <u>here</u>. Open IP Catalog under PROJECT MANAGER, right click on Vivado Repository -> Add Repository.

			Flow Navigate	or 😤	≑ ? _			
			✓ PROJECT	MANAGER				
			🌣 Setting	gs				
			Add So	ources				
			Langu	age Templates				
			👎 IP Cata	alog				
			1					
	Diagram ×	Address Editor	× Address Map	× IP Catalog	×			
	Cores   Inte	erfaces						
	Q	\$ <b>₽</b> •	r   2   0					
	Search: Q-							
	Name		^1	AXI4	Status	License	VLNV	
	> 📄 Vivado F	Repository	Properties	Ctrl+E				
			IP Settings					
			Add Repository					
			Refresh All Reposito	ories				
			Export to Spreadshee	et				
<b>"</b>				11.101				
Provid	ie the Dir	ectory path	n of the IP an	a click Sele	ct.			
And A	dd the SI	Ocard IP to	the Block De	esign				
		Name			^1 AXI4			



## Step 8:

After adding the **SD card** Ip to the Block design, click on **Run Block Automation.** 

Now, connect the clock\_32 pin of the SD card IP to the clk\_SD pin of the Clocking wizard as shown in the below figure.



**Step 9:** Right click on Sdcard ip and click on Make External.



**Step 10:** In Sources tab of Vivado, Right-Click on '**Constraints**' and click '**Add Sources**'.



**Step 11:** Once the "**Add Sources**" tab opens select "**Add or create constraints**" and click on "**Next** ".

	Add Sources
Vivado	This guides you through the process of adding and creating sources for your project
ML Edition	Add or greate constraints
	○ <u>A</u> dd or create design sources
	○ Add or create simulation sources
-	
?	< <u>B</u> ack <u>N</u> ext> Einish Cancel
	' <b>ile</b> ' and give ' <b>SD_test</b> ' as File name. Click ' <b>OK</b> ' and ' <b>Finis</b> l
on ' <b>Create F</b> Add Sources	' <b>ile</b> ' and give ' <b>SD_test</b> ' as File name. Click ' <b>OK</b> ' and ' <b>Finis</b> l
Add Sources	
Add Sources	aints
Add Sources	aints files for physical and timing constraint to add to your project.
Add Sources dd or Create Constr pecify or create constraint	aints files for physical and timing constraint to add to your project.
Add Sources dd or Create Constr pecify or create constraint Specify constraint set:	aints files for physical and timing constraint to add to your project.
Add Sources dd or Create Constr pecify or create constraint Specify constraint set:	aints files for physical and timing constraint to add to your project. Create Constraints File Create a new constraints file and add it to
Add Sources dd or Create Constr pecify or create constraint Specify constraint set:	aints files for physical and timing constraint to add to your project. Create Constraints File Create a new constraints file and add it to
Add Sources dd or Create Constr pecify or create constraint Specify constraint set:	aints files for physical and timing constraint to add to your project. Create a new constraints File Create a new constraints file and add it to your project
Add Sources dd or Create Constr pecify or create constraint Specify constraint set:	aints files for physical and timing constraint to add to your project.
Add Sources dd or Create Constr pecify or create constraint Specify constraint set:	aints files for physical and timing constraint to add to your project. Create a new constraints File Create a new constraints file and add it to your project File name: SD_test
Add Sources dd or Create Constr pecify or create constraint Specify constraint set:	aints files for physical and timing constraint to add to your project.
Add Sources dd or Create Constr pecify or create constraint Specify constraint set:	aints files for physical and timing constraint to add to your project.
Add Sources dd or Create Constr pecify or create constraint Specify constraint set:	aints files for physical and timing constraint to add to your project. Create a new constraints file and add it to File type: DDC File name: SD_test File location: Cancel Add Files Create File

#### Step 12:

#### Copy the following constraints in your constrains file and save it.

set\_property -dict {PACKAGE\_PIN F14 IOSTANDARD LVCMOS33} [get\_ports sdcard\_if\_1\_0\_sd\_clk]

set\_property -dict {PACKAGE\_PIN F15 IOSTANDARD LVCMOS33} [get\_ports sdcard\_if\_1\_0\_sd\_cs]

set\_property -dict {PACKAGE\_PIN B17 IOSTANDARD LVCMOS33} [get\_ports sdcard\_if\_1\_0\_sd\_miso]

set\_property -dict {PACKAGE\_PIN A17 IOSTANDARD LVCMOS33} [get\_ports sdcard\_if\_1\_0\_sd\_mosi]

#### Step 13:

Right-click "**SD\_card**" in the "**Sources**" window, and select "**Create HDL Wrapper**" from the popup menu. Click "**OK**" on the window that appears to finish generating a wrapper.

Sources × Design Signals		? _ 🗆 🖒
Q   素   <b>≑</b>   <b>+</b>   ?   ● 0		۰
v 🖨 Design Sources (1)		
✓ ● ♣ SD_card_wrapper (SD_ca	rd_wra	apper.v) (1)
✓ A	D ca	rd.bd) (1)
> SD_card (SD_card.v)		Source Node Properties
Constraints (1)		Open File
✓		Open With
D SD.xdc		Oranta LIDI. Wrannar
> 🗁 Simulation Sources (1)		Create HDL Wrapper
> 🚍 Utility Sources		View Instantiation Template
		Generate Output Products
		Reset Output Products

#### Step 14:

Click "Generate Bitstream" under the "Program and Debug" section to synthesize, implement and generate a bitstream.

✓ SYNTHESIS
Run Synthesis
> Open Synthesized Design
Run Implementation
> Open Implemented Design
✓ PROGRAM AND DEBUG
Generate Bitstream
> Open Hardware Manager

**Step 15:** After generating the bitstream successfully, select **Export -> Export Hardware** from the **File menu**. Click **Next**.

	Add Sources Alt+A Save Block Design Ctrt+S Save Block Design As Close Block Design Model Checgooint Constraints Simulgiton Waveform IP Text Egitor	>     >
	I <u>m</u> port → Expor <u>t</u>	Export Hardware
<b>R</b> Nu	Print Ctrl+P Egit	

Select the "include bitstream" checkbox and click Next.

Exp	ort Hardware Platform	×
Outp Set the	ut e platform properties to inform downstream tools of the intended use of the target platform's hardware design.	ת
0	Pre-synthesis This platform includes a hardware specification for downstream software tools.	
۲	Include bitstream This platform includes the complete hardware implementation and bitstream, in addition to the hardware specification for software tools.	
	< <u>₿ack</u> <u>N</u> ext> Einish Can	icel

Provide the **XSA file name** and save it at a suitable **location**. Click **Next** and click **Finish** in the next dialog box.

#### Step 16:

Select Launch Vitis IDE from the Tools menu.

	<u>T</u> ools	Rep <u>o</u> rts <u>W</u> indow La <u>v</u> out	<u>V</u> iew <u>H</u> e		
		Validate Design	F6		
		Create and Package New IP			
		Create Interface Definition			
		Run Tcl Script			
		Property Editor	Ctrl+J		
		Associate EL <u>F</u> Files			
		Generate Memory Configuration File			
		Compile Simulation Libraries			
		Vivado Store			
		C <u>u</u> stom Commands	×		
		Launch Vitis IDE			
	Q	Language <u>T</u> emplates			
	•	Settings			
Step 17:					

After Vitis Unified IDE window opens, click on "**Open Workspace**" and select necessary folder to keep the Vitis files.

Welcome to the Vitis Unified IDE
Get Started
₽ <u>Open Workspace</u>
â <u>Examples</u>
<i>Aigrate Classic IDE Workspace</i>

#### **Step 18:**

Create a new platform for the project, by selecting "**Create Platform Component**", click "**Next**", in the Flow tab select the XSA file saved using the step 20 and finally click "**Next**" and "**Finish**" respectively.



After successful creation of the platform, build the platform.

∼ FLOW	♦	*	
Component	🔹 SD_test	~	ŝ
沦 Build			

#### Step 19:

Next create the Helloworld Application component by selecting the "Helloworld" template from the "examples",



In "Create Application Component" tab specify project name and location, click "Next"

Create Application Component - Hello World X Name and Location > Hardware > Domain > Sysroot > Summary							
Name and Location							
Choose a name for your component and specify a directory where component data files will be stored							
Component name	SD_card_test						
Component location	C:\projects\Elbert_S7\SD_card\vitis	<u>Browse</u>					
Component will be created of	at C\projects\Elbert_S7\SD_card\vitis\SD_card_test		Next				

Select newly created Platform and click "Next".

Create Application Component - Hello World									
Name and Location > Hardware > Domain > Summary									
Select Platform									
Platforms supporting the selected example from your repositories. To create a new platform, use "File -> New Component -> Platform"									
Q ∪ + × ×					ן				
NAME	BOARD	FLOW	VENDOR	РАТН	T I				
✓ C:\projects\Elbert_S7\SD_card\vitis\SD_test\export\SD_te				ojects\Elbert_S7\SD_card\vitis\SD_test\export\SD_test					
(1)									
SD_test	elbert	Embedded	xilinx.com	S7\SD_card\vitis\SD_test\export\SD_test\SD_test.xpfm					
Cancel				(Back Nex	t )				

Select the domain as "**Standalone\_microblaze\_o**" and click "**Next**" and click on "**Finish**"

**Step 20:** Download the SD\_test.c file from <u>here</u>, Copy the content of SD\_test.c file to the helloworld.c file to test SD card .

After adding the source file build the Project.

#### Step 21:

Once the build is completed successfully, power up Elbert S7 FPGA Development Board using an USB type C cable. and insert the SD card into the micro SD card slot of Elbert S7 FPGA Development Board.

#### Step 22:

Program the FPGA on Elbert S7 with a simple boot loop program by selecting the **Program Device** option from the **Vitis menu**.



Once the "Program Device" window opens click on "Program ".

Program Device				×					
Specify the bitstream and the ELF files that reside in BRAM memory.									
Project	5D_card_test ~								
Connection	.ocal v	<u>New</u>							
Bitstream/PDI	\projects\Elbert_S7\SD_card\vitis\SD_card_test\_ide\bitstream\SD_card_wrapper.bit	<u>Browse</u>	<u>Search</u>						
C	Partial Bitstream								
BMM/MMI File	projects\Elbert_S7\SD_card\vitis\SD_card_test\_ide\bitstream\SD_card_wrapper.mmi	<u>Browse</u>	<u>Search</u>						
Software Configuration									
PROCESSOR	ELF/MEM FILE TO INITIALIZE IN BLOCK RAM								
microblaze_0	bootloop			~					
Skip Revision Check									
Cancel		Genera	ate	Program					

#### Step 23:

Meanwhile, open any serial terminal program (such as PuTTY, Teraterm etc) and open the port corresponding to Elbert S7 with a 9600 baud rate (the default baud rate given in UART IP). Program the board by selecting the "Run".



#### Step 24:

If everything went well, Serial terminal would show the execution is Successful.



## REFERENCES

## > **PRODUCT LINKs**:

- o <u>Product Page</u>.
- o <u>User manual</u>.
- o <u>Schematics</u>.
- o <u>Xdc Constraints file</u>.

## > Tools Link:

- o <u>Vivado Design suite</u>.
- <u>PUTTY</u>.

