A Report on

Camera Interfacing to FPGA

for

Internship at Indrones Private Limited

by

1. Lavlesh A. Poyarekar (3020144)
2. Sarvesh S. Shinde (3020154)

Under the guidance of

Dr. Apurva Joshi

Dr. Megha Kolhekar

Dr. Prasad Kulkarni

Fr. C. Rodrigues Institute of Technology, Vashi

Indrones Private Limited, Vashi

ABSTRACT

In most of the real time video processing applications, cameras are used to capture live video with embedded systems/Field Programmable Gate Arrays (FPGAs) to process and convert it into the suitable format supported by display devices. In such cases, the interface between the camera and display device plays a vital role with respect to the quality of the captured and displayed video, respectively. In this internship we have attempted to create an efficient FPGA-basedlow cost Complementary Metal Oxide Semiconductor (CMOS)camera interfacing with Nexys A7 100-T FPGA for live video streaming and processing applications. The main significance of the proposed system is it has **minimum latency** in real video streaming.

One major block of dual port RAM was designed successfully as per the required parameters where the data is initially stored with the help of Intellectual Property core of FPGA. The output video was observed through VGA on computer monitor with maximum delay of **13 nanoseconds** which met the expectation of the project to larger extent. The mixing of red, green and blue tints can also be improved for better video quality.

Contents

List of figures

List of Tables

List of symbols and abbreviations

Symbols

Hz	Hetrz

Abbreviations

ADC	Analog To Digital Converter
DDR	Double Data Rate
DSP	Digital Signal Processor
FPGA	Field Programming Gate Array
HDL	Hardware Description Language
JTAG	Joint Test Action Group
LED	Light Emitting Diode
PDM	Pulse Density Modulation
PLL	Phase Lock Loop
PWM	Pulse Width Modulation
SCCB	Serial Camera Control Bus
UART	Universal Asynchronous Receiver Transmitter
VGA	Video Graphics Array
XDC	Xilinx Design Constraint file
WNS	Worst Negative Slack

I) COMPONENTS

1) Nexys A7 100T FPGA

Features

The features of the FPGA board are [4]:

- **Artix-7 FPGA**
    - 15,850 Programmable logic slices, each with four 6-input LUTs and 8 flip-flops (*8,150 slices)
    - 4,860 Kbits of fast block RAM (*2,700 Kbits)
    - Six clock management tiles, each with phase-locked loop (PLL)
    - 240 DSP slices (*120 DSPs)
    - Internal clock speeds exceeding 450 MHz
    - Dual-channel, 1 MSPS internal analog-digital converter (XADC)
- **Memory**
    - 128MiB DDR2
    - Serial Flash
    - microSD card slot
    - USB-JTAG programming circuitry
    - USB-UART bridge
    - USB HID Host for mice, keyboards and memory sticks

- **Audio and Video**
    - 12-bit VGA output
    - PWM audio output

2) OV 7670 Camera

The OV7670/OV7171 CAMERACHIPTM which is shown in figure 1.1 is a low voltage CMOS image sensor that provides the full functionality of a single-chip VGA camera and image processor in a small footprint package. The OV7670/OV7171 provides full-frame, sub-sampled or windowed 8-bit images in a wide range of formats, controlled through the Serial Camera Control Bus (SCCB) interface. This product has an image array capable of operating at up to 30 frames per second (fps) in VGA with complete user control over image quality, formatting and output data transfer. All required image processing functions, including exposure control, gamma, white balance, color saturation, hue control and more, are also programmable through the SCCB interface. In addition, OmniVision CAMERACHIPs use proprietary sensor technology to improve image quality by reducing or eliminating common lighting/electrical sources of image contamination, such as fixed pattern noise (FPN), smearing, blooming, etc., to produce a clean, fully stable color image.

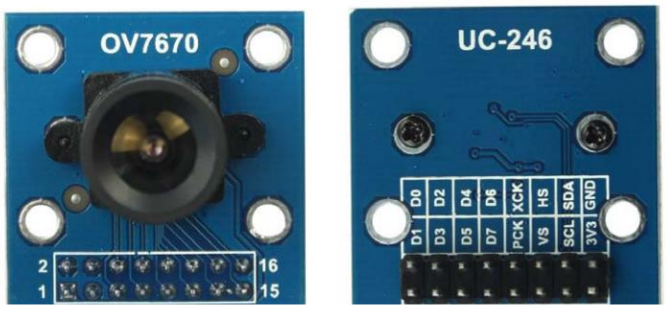


Figure 1.1: OV7670 Camera (Front and Back View)

Pin description for the camera is mentioned in table 1.1:

Pin No.	PIN NAME	TYPE	DESCRIPTION
1	VCC	POWER	3.3v Power supply
2	GND	Ground	Power ground
3	SCL	Input	Two-Wire Serial Interface Clock
4	SDATA	Bi-directional	Two-Wire Serial Interface Data I/O
5	VSYNC	Output	Active High: Frame Valid; indicates active frame
6	HREF	Output	Active High: Line/Data Valid; indicates active pixels
7	PCLK	Output	Pixel Clock output from sensor
8	XCLK	Input	Master Clock into Sensor
9	DOUT9	Output	Pixel Data Output 9 (MSB)
10	DOUT8	Output	Pixel Data Output 8
11	DOUT7	Output	Pixel Data Output 7
12	DOUT6	Output	Pixel Data Output 6
13	DOUT5	Output	Pixel Data Output 5
14	DOUT4	Output	Pixel Data Output 4
15	DOUT3	Output	Pixel Data Output 3
16	DOUT2	Output	Pixel Data Output 2 (LSB)

Table 1.1: Pin description of OV 7670 camera

Software Used:

Xilinx VIVADO

Vivado Design Suite is a software suite produced by Xilinx for synthesis and analysis of hardware description language (HDL) designs, superseding Xilinx ISE with additional features for system on a chip development and high-level synthesis. Vivado represents a ground-up rewrite and re-thinking of the entire design flow (compared to ISE).

Like the later versions of ISE, Vivado includes the in-built logic simulator. Vivado also introduces high-level synthesis, with a toolchain that converts C code into programmable logic. [3]

II) BLOCK DIAGRAM

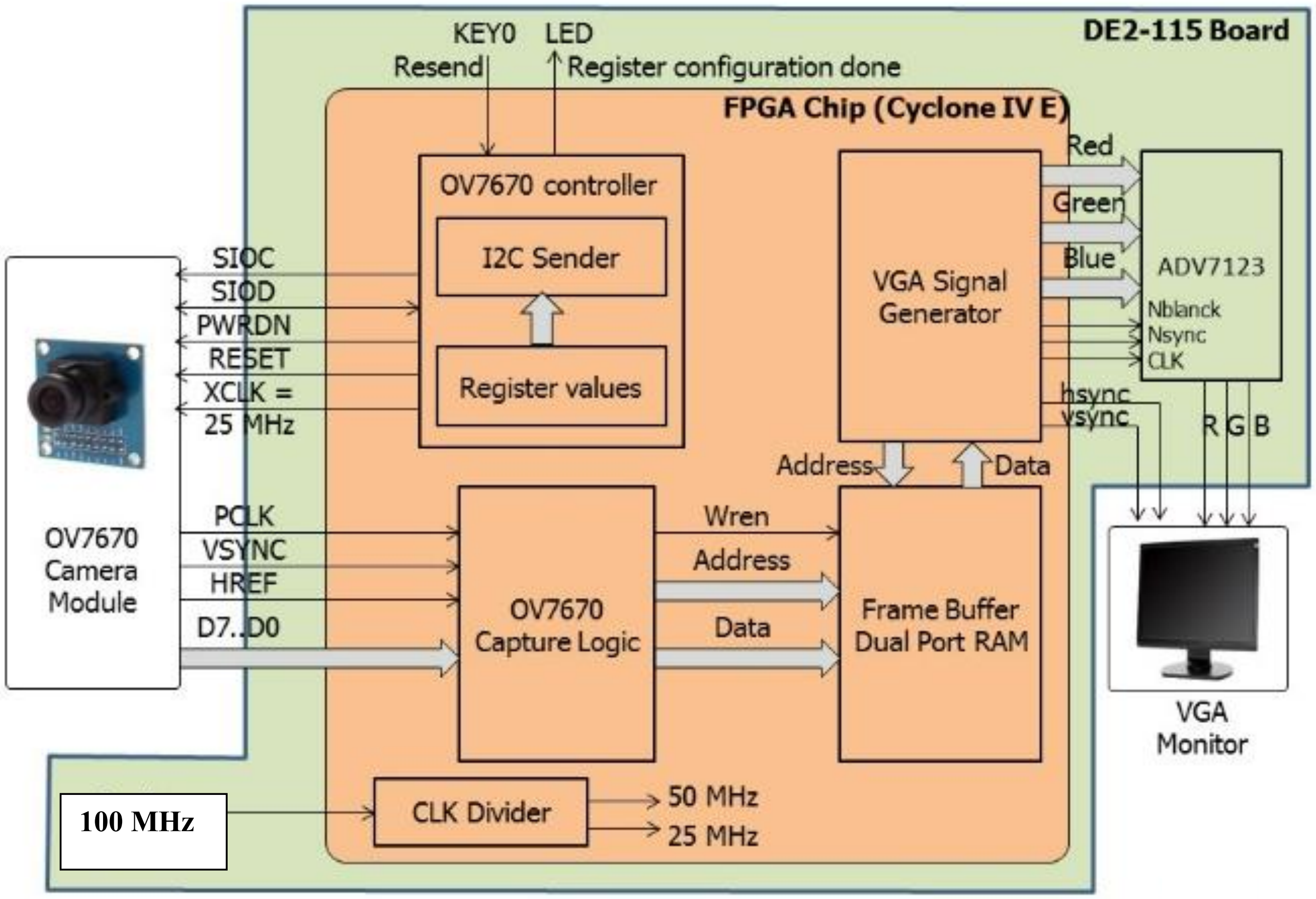


Figure 2.1: Block Diagram of the system

The OV 7670 camera is connected to FPGA. Block diagram in figure 2.1 [7] shows that the data lines are connected to image capture block whereas the control signals of the camera are connected to the camera controller block. The image capture take the captured image data to the dual port RAM which is the frame buffer. As an IP core is used here and as RAM is dual port processing becomes faster. Frame buffer sends data to VGA. Here the RGB colour mixing of the video occurs. This processed video is than with the help of timing signals of VGA are fed to VGA supported display device such as computer monitor where the final output is displayed.

III) METHODOLOGY

1) Create a logic and flow of design from camera to display device
2) Create VHDL code blocks for image capture, address generator, RGB, VGA and image buffer.
3) For image buffer use an IP core by creating a dual port RAM so as to increase the speed of processing by introducing parallelism.
4) Instantiate all the sub-modules of code in the Top module.
5) Map the entire design with respective to FPGA board by creating xdc file.
6) Simulate the design.
7) Synthesize the design.
8) Implement the design.
9) Create bitstream.
10) Upload the bitstream to FPGA.
11) Connect the VGA port of FPGA to the VGA of monitor screen to analyze the output.

Design Flow:

1. Camera configuration (OV7670_controller.vhd): To properly configure the OV7670 camera, check its specifications to see what communication protocol it supports for configuration and data capture. In this case, as shown in figure 3.1, it is an I2C type SSCB interface, so I2C code is needed to communicate with the camera, configure it and get image data. More information about the I2C-like interface can be found at i2c_sender.vhd. Check the device control register list to set it correctly. In addition to the output format (RGB, QVGA, QCIF, etc.), RGB format (RGB565, RGB555, etc.), timing signals (PCLK, HREF, VSYNC), matrix coefficients, including MTX1-MTX6, are important factors in image quality.[1] See OV7670_registers.vhd for more information on control register values.

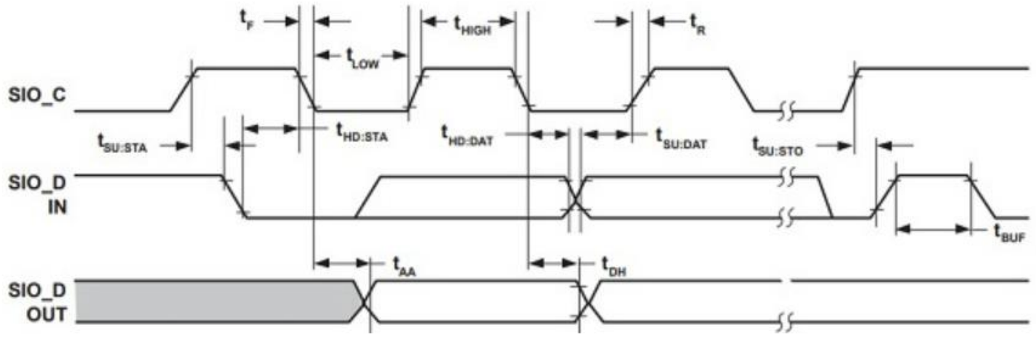


Figure 3.1: I2C like SCCB interface for communicating with the OV 7670 camera

2. Image data capture (OV7670_capture.vhd): After setting up the camera, the next step is to capture the image data. Check the camera specifications again to see the output timing chart for the selected output format. For example, in this project, the RGB565 format is selected so that the RGB565 output timing scheme is used to correctly capture the RGB data from the camera's output signals. OV7670_capture.vhd is created based on the following timing scheme as shown in the figure 3.1

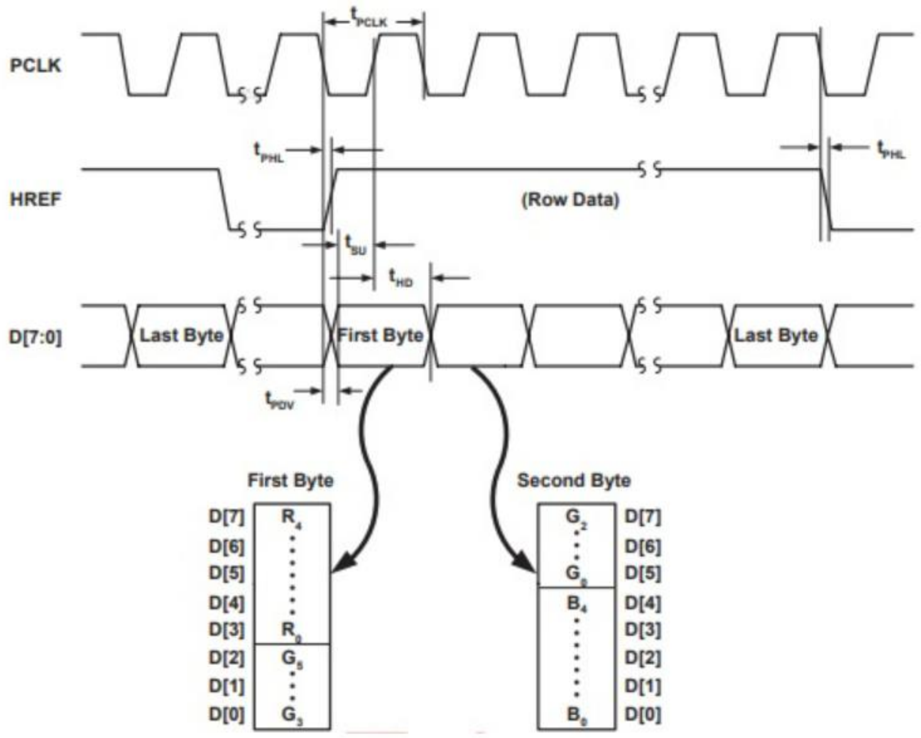
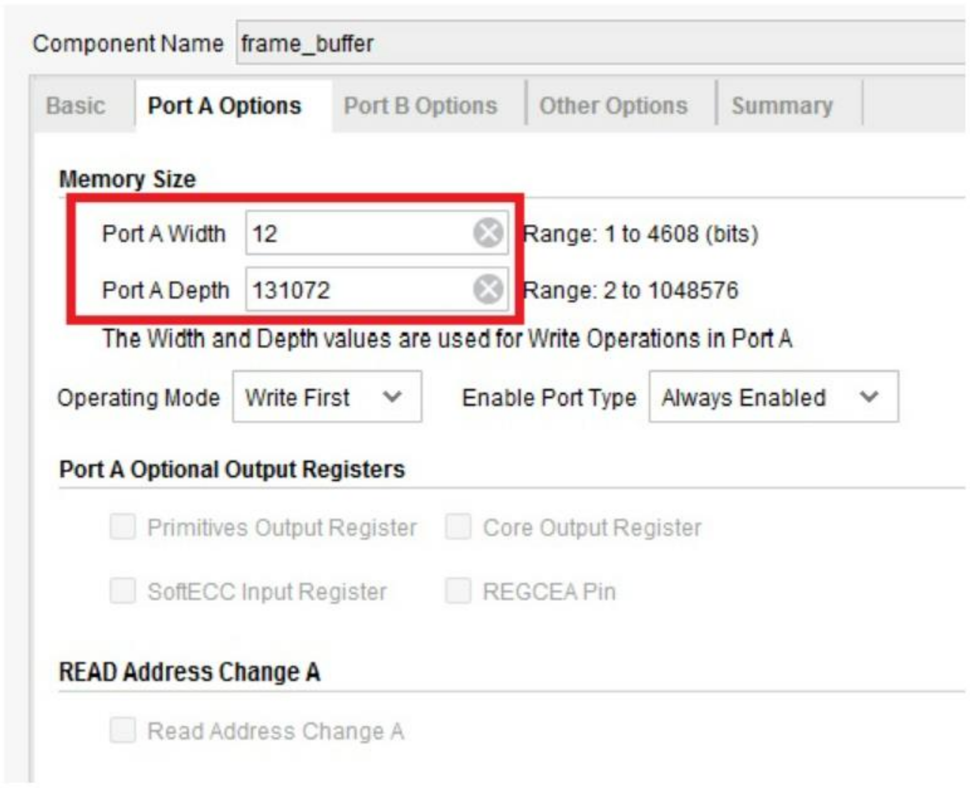


Figure 3.2: Timing diagram for image capture block

3. Image data storage (frame_buffer.vhd): After the camera is set up and the image data is correctly captured, the image data is stored in the cache. We have kept the framebuffer size by a factor of 4 to fit the Basys 3 FPGA, while keeping the full 640x480 image size on the VGA screen. For 320x240 and 160x120, it is designed to display a 320x240 image size by pressing the Left button, while pressing the Right button on the FPGA will select an image size of 160x120 on the VGA screen. You can do this in Vivado by updating the block memory file and #40;frame_buffer.xco - Xilinx ISE created by Core Generator and#41 as shown in figure 3.3; to the Vivado-compatible LogiCore IP Block Memory v8.4 (frame_buffer.xci), reducing the memory size by a factor of 4, or use the Vivado IP List Block Memory Generator to create a new frame buffer with a depth of 131072 (17-bit address).



Component Name: frame_buffer

Port A Width: 12 Range: 1 to 4608 (bits)

Port A Depth: 131072 Range: 2 to 1048576

The Width and Depth values are used for Write Operations in Port A

Operating Mode: Write First

Enable Port Type: Always Enabled

Figure 3.3: IP core customization

IV) **RESULTS**

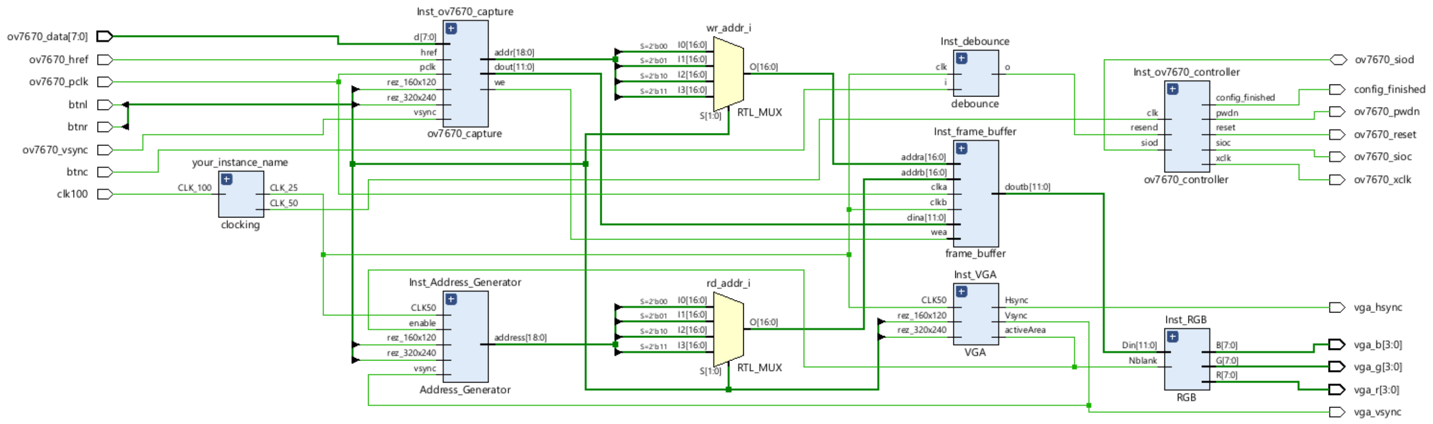


Figure 4.1: RTL design of the work

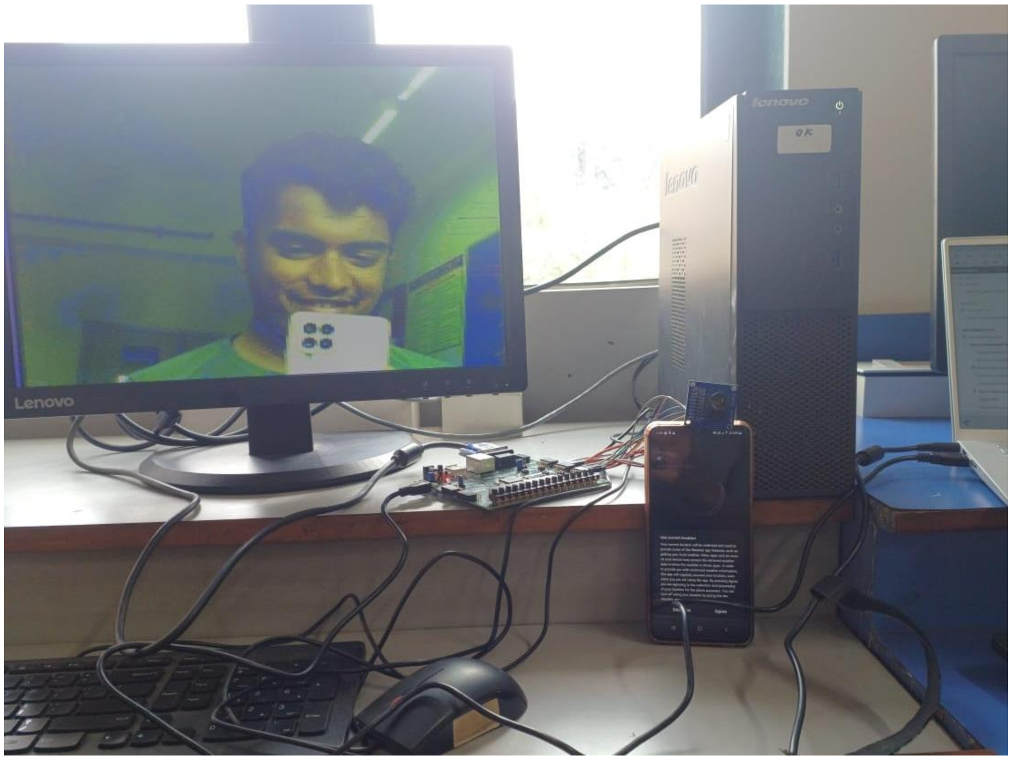


Figure 4.2: Output Image

The VHDL code was designed by referring various papers and websites. It was then successfully compiled and according to figure 4.1 the RTL design was designed . In the next step synthesis took place followed by implementation. In implementation report it is observed that the Worst Negative Slack i.e. the maximum delay for video streaming is **13 nanoseconds** which is meeting the requirements of the project as observed in figure 4.4. The latency is good but there are some problems in colour mixing as seen in figure 4.2.

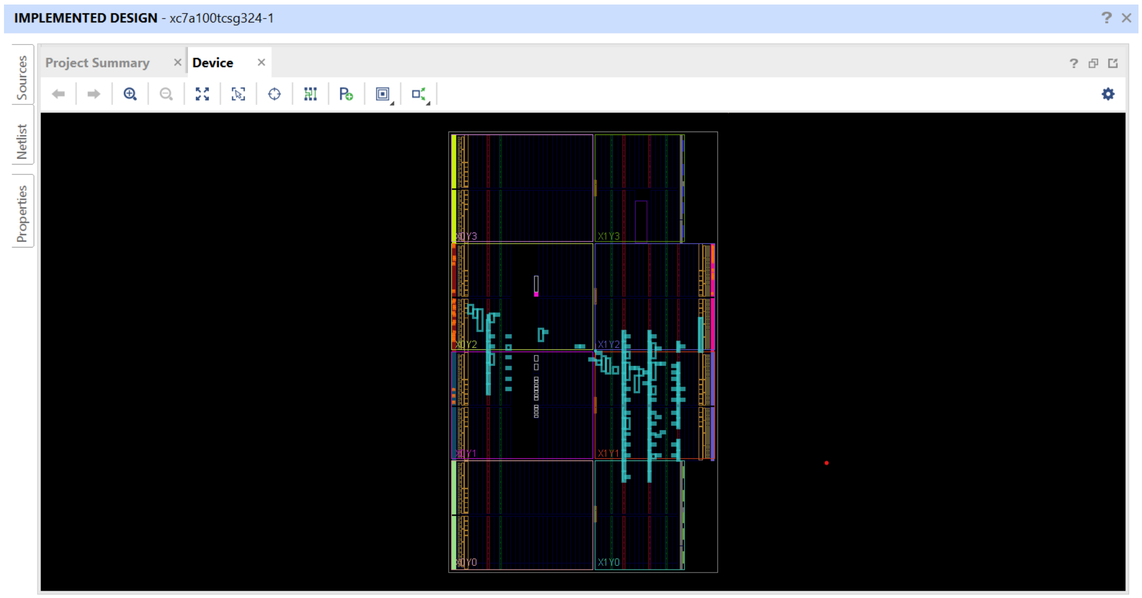
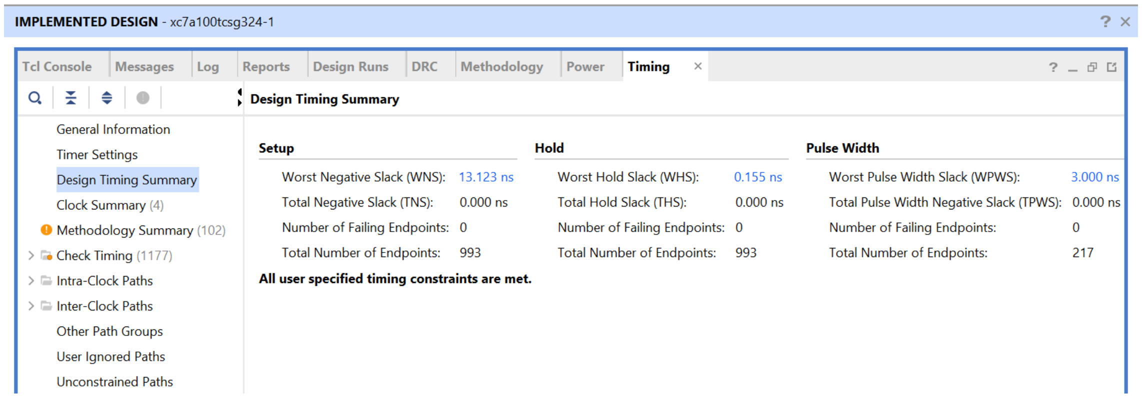


Figure 4.3: Layout of the implemented design



Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	13.123 ns	Worst Hold Slack (WHS):	0.155 ns	Worst Pulse Width Slack (WPWS):	3.000 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	993	Total Number of Endpoints:	993	Total Number of Endpoints:	217

All user specified timing constraints are met.

Figure 4.4: Implementation Report showing WNS of 13ns

Merits and demerits which were observed are as follows:

<u>Merits:</u>

- Faster Processing
- Minimum Latency
- Entire Processing on single board

<u>Demerits:</u>

- More wires connected to camera
- Noise disturbance due to multiple wired
- RGB mixing is not up to the mark.

V) <u>CONCLUSIONS</u>

The OV 7670 CMOS camera was successfully interfaced with NEXYS A7 100-T FPGA. The VHDL code was revised and some sections were added as per the board requirements. One major block of dual port RAM was designed successfully as per the required parameters where the data is initially stored with the help of Intellectual Property core of FPGA. The output video was observed through VGA on computer monitor with maximum delay of **13 nanoseconds** which met the expectation of the project to larger extent. Also there is scope to improve the mixing of shades red, green and blue in order to get better video quality.

REFERENCES

[1] FPGACam: A FPGA based efficient camera interfacing architecture for real time video processing Sayantam Sarkar1 | Satish S. Bhairannawar 2 | Raja K.B.3 19 March 2021

[2]Digilent NexysVideo FPGA Board User Guide. https://reference. digilentinc.com/reference/programmablelogic/nexysvideo/ referencemanual. Accessed 26 Mar 2020

[3] https://www.xilinx.com/products/design-tools/vivado.html

[4] https://digilent.com/reference/programmable-logic/nexys-a7/reference-manual

[5] Basys 3 FPGA OV7670 Camera - FPGA4student.com

[6] http://web.mit.edu/6.111/www/f2016/tools/OV7670_2006.pdf

[7] http://www.dejazzer.com/eigenpi/digital_camera/digital_camera.html