VL505 System Design with FPGA

Course Project

Lakshmi Sai Niharika Vulchi MT2023513

Valipireddy Pranathi MT2023510

GitHub link: https://github.com/V-Pranathi/FPGA_PROJECT

Guidance: Prof. Nanditha Rao

Project statement:

Read in the image from a laptop-camera, store it in block RAM, display it on VGA. Transform the image and display it in real-time on the VGA monitor.

Project hierarchy:

1. Implementation of VGA controller with FPGA
2. Interfacing the UART for transferring the image to the FPGA
3. Implementing a real time parallel interface for the image taken from laptop camera to be displayed on the screen using VGA controller
4. Transforming the image, applying image blurring using image convolution, and displaying it on FPGA

Section 1: Project Description and Implementation

VGA controller

VGA, which stands for Video Graphics Array, is a standard interface used for connecting computers to displays, such as monitors and projectors.

We implemented VGA for 1280x1024 display with 60Hz frequency. The standard horizontal and vertical porch values for this specific display resolution are as follows:

a. Horizontal front porch, retrace, back porch = 48, 112, 248 respectively
b. Vertical front porch, retrace, back porch = 1,3, 38 respectively

Horizontal sync and vertical sync signals are the control signals which are given to the VGA controller. These timing signals are asserted after the completion of horizontal **row** of pixels and the entire **frame** respectively.

RGB pixel data each of 4 bit is sent to the VGA interface for the display.

VGA clock frequency: We are using a display resolution of 1280x1024, to match the display frequency of 60 frames per second, each pixel in the image should be transmitted at a frequency of **108 MHz**.

1 frame = ((1280+48+112+248) x (1024 +1+3+38)) pixels: 60 Hz

1 pixel: 107.964 MHz → 108 MHz

We are displaying the image in 320x240 dimension and 12 bit pixel data on the 1280x1024 screen, thus the remaining screen except that of the image is made black.

Storing the image on FPGA

The image pixels each of length 12 bits, where each RGB component is of 4 bits is stored in a block RAM. Basys 3 board can be used to store only one image of 320x240 size where each pixel is 12 bit width or two images of 320x240 size where each pixel is 11 bit width. A coe file is created using python which converts the image pixel data to 12 bit.

UART interface

A parallel UART interface is made which transfers the image taken from the laptop camera to FPGA. Python file reads the data from the image and writes it to the serial port which is connected to the laptop. This data is then sent it to the FPGA using USB, on the Basys 3 board FTDI FT2232 chip converts the USB signal to UART and this is sensed by FPGA using the specific pins.

A **two state FSM** is designed to handle the operation of the module. FSM states:

- LOW_BYTE: for the lower 8 bits of data – 6 bits of pixel data and 2 safety bits
- HIGH_BYTE: for the higher 8 bits of data

The **clock frequency** used is 50 MHz and **baud rate** of 1834200 which enables very fast transmission of image from laptop to the board.

Baud rate is chosen to be such that we can transfer 2 images of 320x240 of 11bit data width for pixel.

Oversampling of data is done to reduce the interference of noise and make sure that the right data is transferred over UART, here the oversampling factor is 15, which can be chosen as your wish

UART_frequency = oversampling_factor * baud rate * clock_divider_factor

<u>Image transformation (blurring)</u>

We are applying image transformation on the received image from UART, technique used is image blurring.

The image which is stored in the first block ram is read and then the convolution operation is operated on each pixel and its surrounding pixel by the weights of the kernel and then the data is stored in the second block ram.

The kernel we used is a 3x3 kernel with all the values as 1/9

The data which is being written into the second block ram is read and sent to the VGA for the image display on the screen.

Below figure is the block diagram detailing our project work.

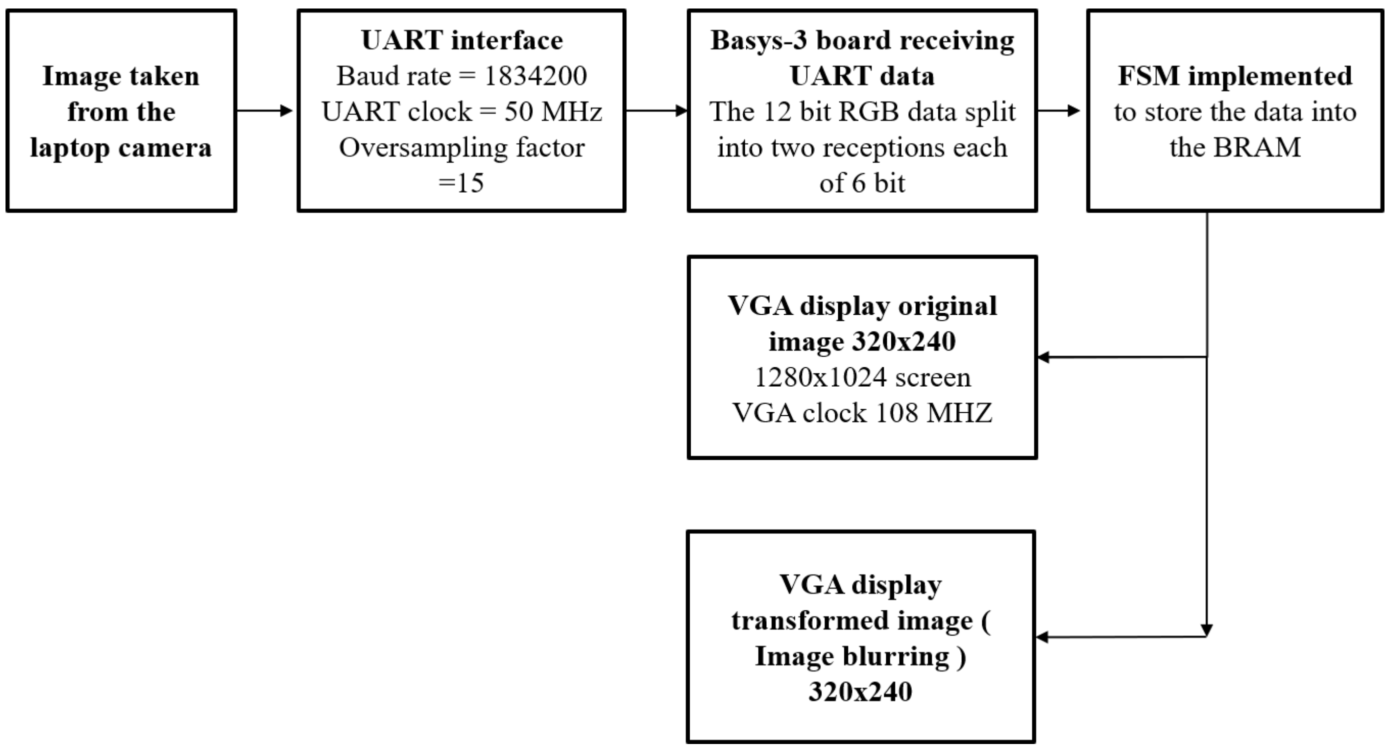
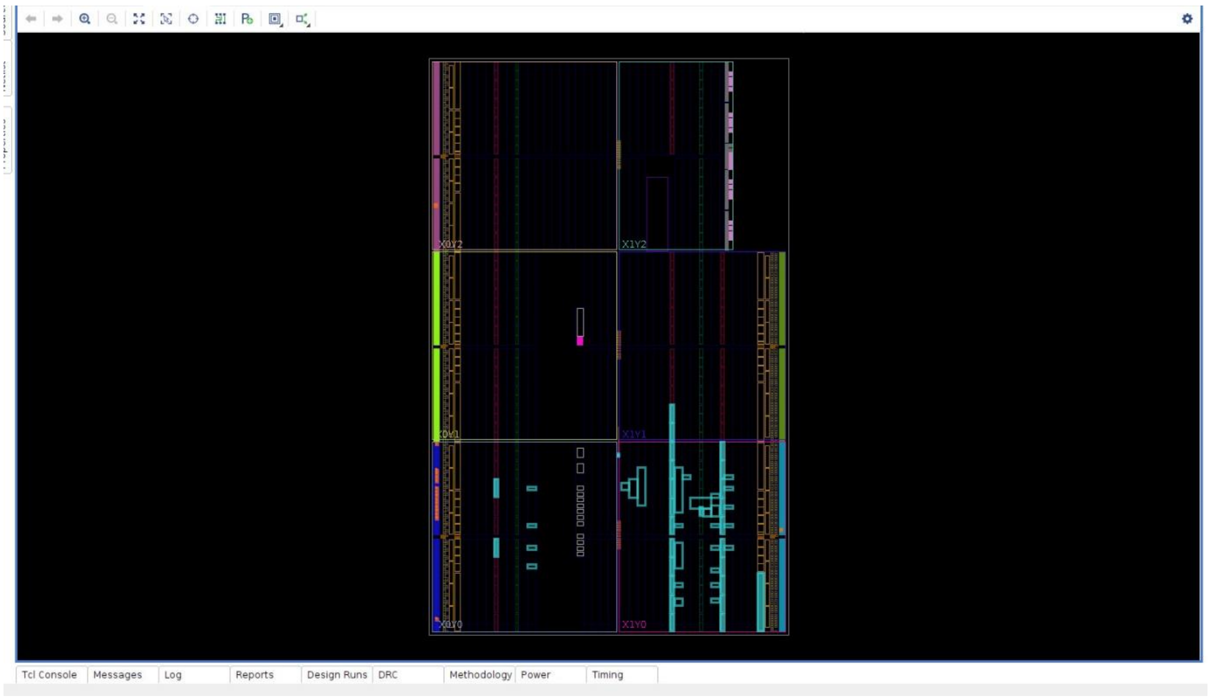


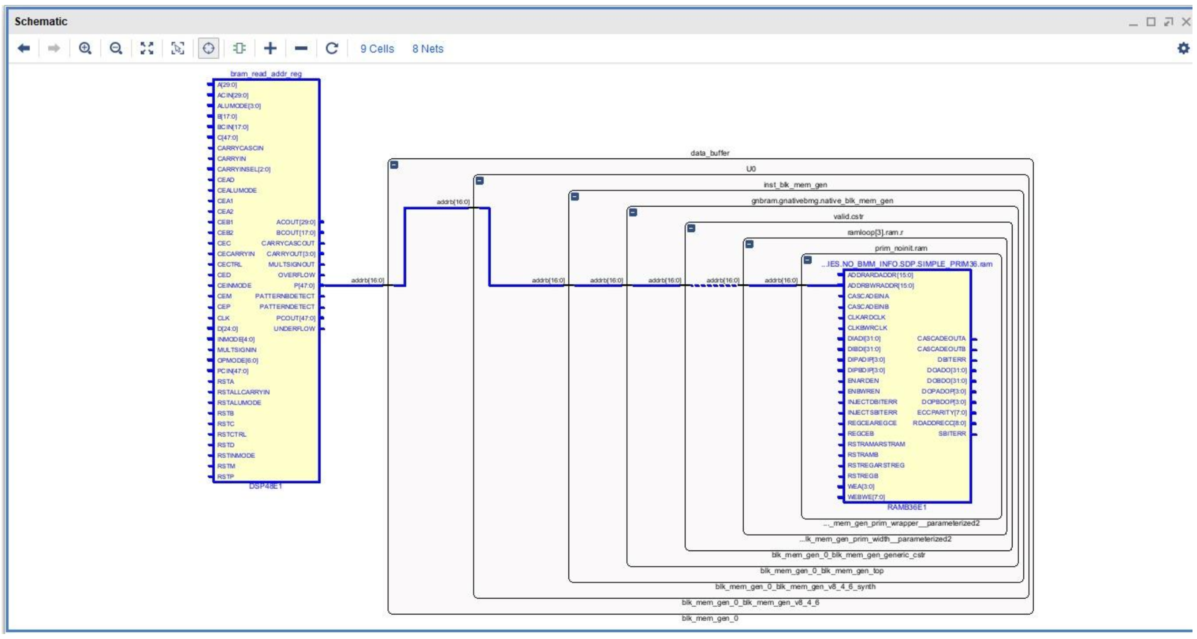
Image taken from the laptop camera
UART interface
Baud rate = 1834200
UART clock = 50 MHz
Oversampling factor =15
Basys-3 board receiving UART data
The 12 bit RGB data split into two receptions each of 6 bit
FSM implemented to store the data into the BRAM
VGA display original image 320x240
1280x1024 screen
VGA clock 108 MHZ
VGA display transformed image (Image blurring) 320x240

Section 2: Vivado Implementation results

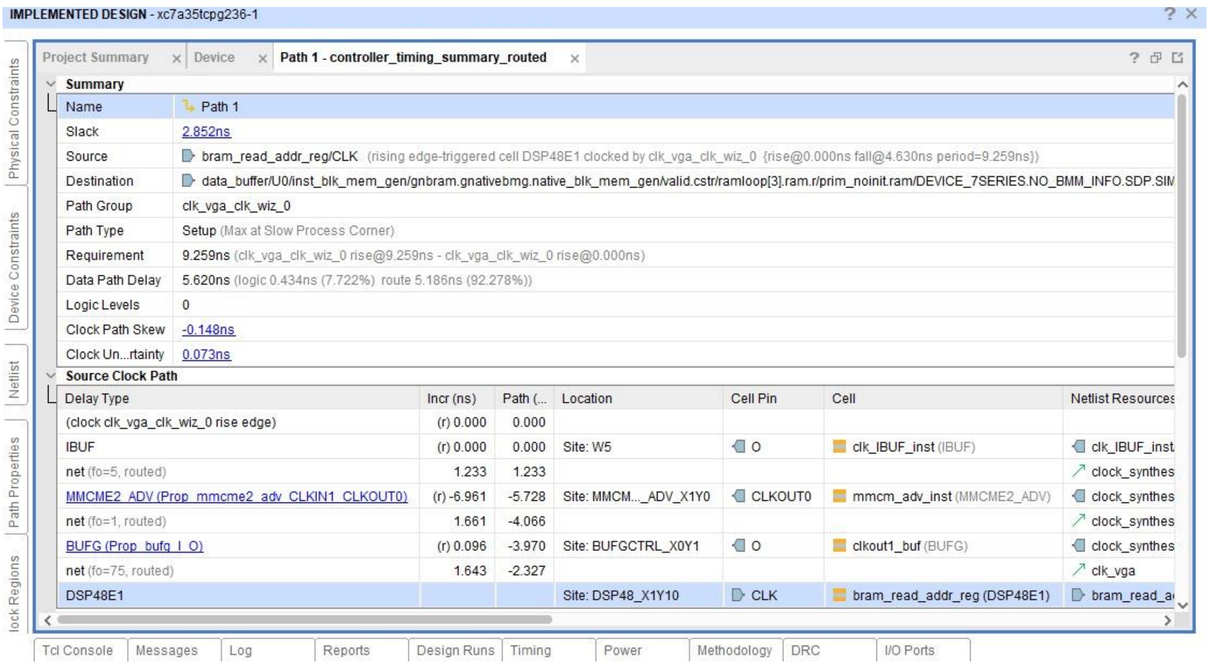
Post implementation floorplan:



Post implementation schematic with critical path highlighted


Schematic
9 Cells
8 Nets
bram_read_addr_reg
DSP48E1
addrb[16:0]
data_buffer
U0
inst_blk_mem_gen
gnbram.gnativebmg.native_blk_mem_gen
valid.cstr
ramloop[3].ram.r
prim_noinit.ram
...IES.NO_BMM_INFO.SDP.SIMPLE_PRIM36.ram
RAMB36E1
..._mem_gen_prim_wrapper__parameterized2
...lk_mem_gen_prim_width__parameterized2
blk_mem_gen_0_blk_mem_gen_generic_cstr
blk_mem_gen_0_blk_mem_gen_top
blk_mem_gen_0_blk_mem_gen_v8_4_6_synth
blk_mem_gen_0_blk_mem_gen_v8_4_6
blk_mem_gen_0

Post implementation critical path analysis with source and destination



IMPLEMENTED DESIGN - xc7a35tcpg236-1

Path 1 - controller_timing_summary_routed

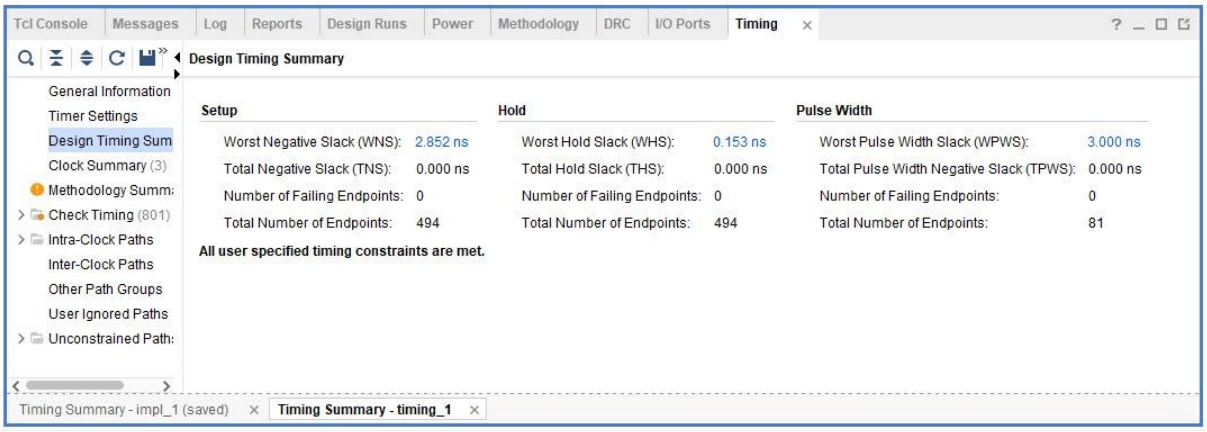
Summary

Name	Path 1
Slack	2.852ns
Source	bram_read_addr_reg/CLK (rising edge-triggered cell DSP48E1 clocked by clk_vga_clk_wiz_0 {rise@0.000ns fall@4.630ns period=9.259ns})
Destination	data_buffer/U0/inst_blk_mem_gen/gnbram.gnativebmg.native_blk_mem_gen/valid.cstr/ramloop[3].ram.r/prim_noinit.ram/DEVICE_7SERIES.NO_BMM_INFO.SDP.SIM
Path Group	clk_vga_clk_wiz_0
Path Type	Setup (Max at Slow Process Corner)
Requirement	9.259ns (clk_vga_clk_wiz_0 rise@9.259ns - clk_vga_clk_wiz_0 rise@0.000ns)
Data Path Delay	5.620ns (logic 0.434ns (7.722%) route 5.186ns (92.278%))
Logic Levels	0
Clock Path Skew	-0.148ns
Clock Un...rtainty	0.073ns

Source Clock Path

Delay Type	Incr (ns)	Path (...	Location	Cell Pin	Cell	Netlist Resources
(clock clk_vga_clk_wiz_0 rise edge)	(r) 0.000	0.000				
IBUF	(r) 0.000	0.000	Site: W5	O	clk_IBUF_inst (IBUF)	clk_IBUF_inst
net (fo=5, routed)	1.233	1.233				clock_synthes
MMCME2_ADV (Prop_mmcme2_adv_CLKIN1_CLKOUT0)	(r) -6.961	-5.728	Site: MMCM..._ADV_X1Y0	CLKOUT0	mmcm_adv_inst (MMCME2_ADV)	clock_synthes
net (fo=1, routed)	1.661	-4.066				clock_synthes
BUFG (Prop_bufg_I_O)	(r) 0.096	-3.970	Site: BUFGCTRL_X0Y1	O	clkout1_buf (BUFG)	clock_synthes
net (fo=75, routed)	1.643	-2.327				clk_vga
DSP48E1			Site: DSP48_X1Y10	CLK	bram_read_addr_reg (DSP48E1)	bram_read_a

Post implementation timing report:

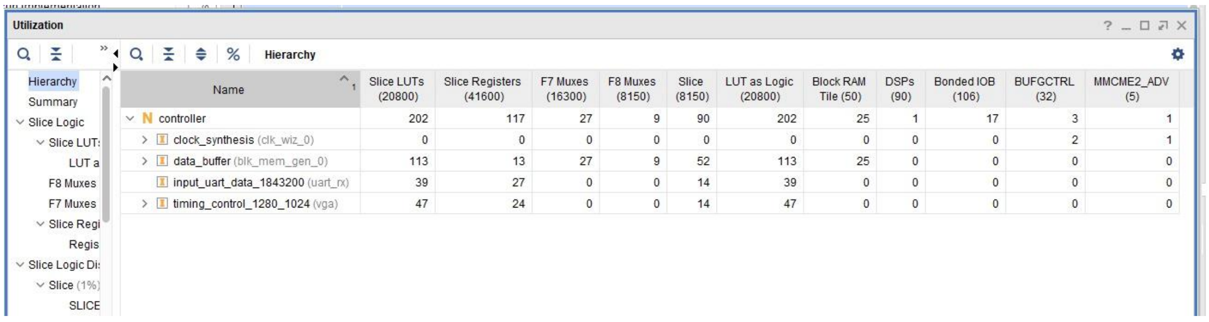


Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	2.852 ns	Worst Hold Slack (WHS):	0.153 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:	494	Total Number of Endpoints:	494	Total Number of Endpoints:	81

All user specified timing constraints are met.

Post implementation utilization report:



Utilization

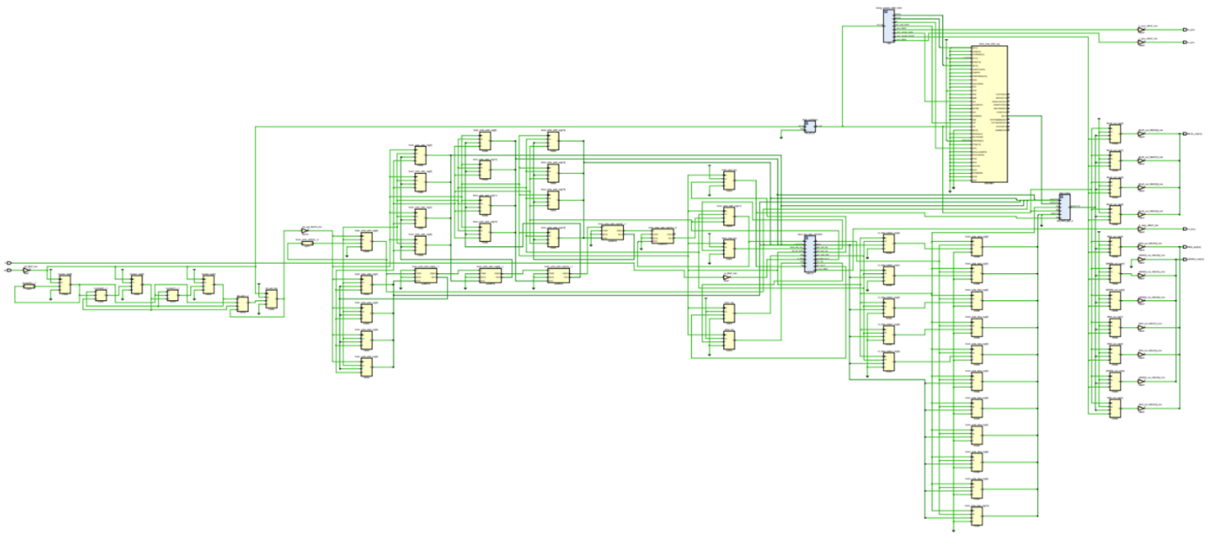
Hierarchy

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	F8 Muxes (8150)	Slice (8150)	LUT as Logic (20800)	Block RAM Tile (50)	DSPs (90)	Bonded IOB (106)	BUFGCTRL (32)	MMCME2_ADV (5)
controller	202	117	27	9	90	202	25	1	17	3	1
clock_synthesis (clk_wiz_0)	0	0	0	0	0	0	0	0	0	2	1
data_buffer (blk_mem_gen_0)	113	13	27	9	52	113	25	0	0	0	0
input_uart_data_1843200 (uart_rx)	39	27	0	0	14	39	0	0	0	0	0
timing_control_1280_1024 (vga)	47	24	0	0	14	47	0	0	0	0	0

Timing analysis:

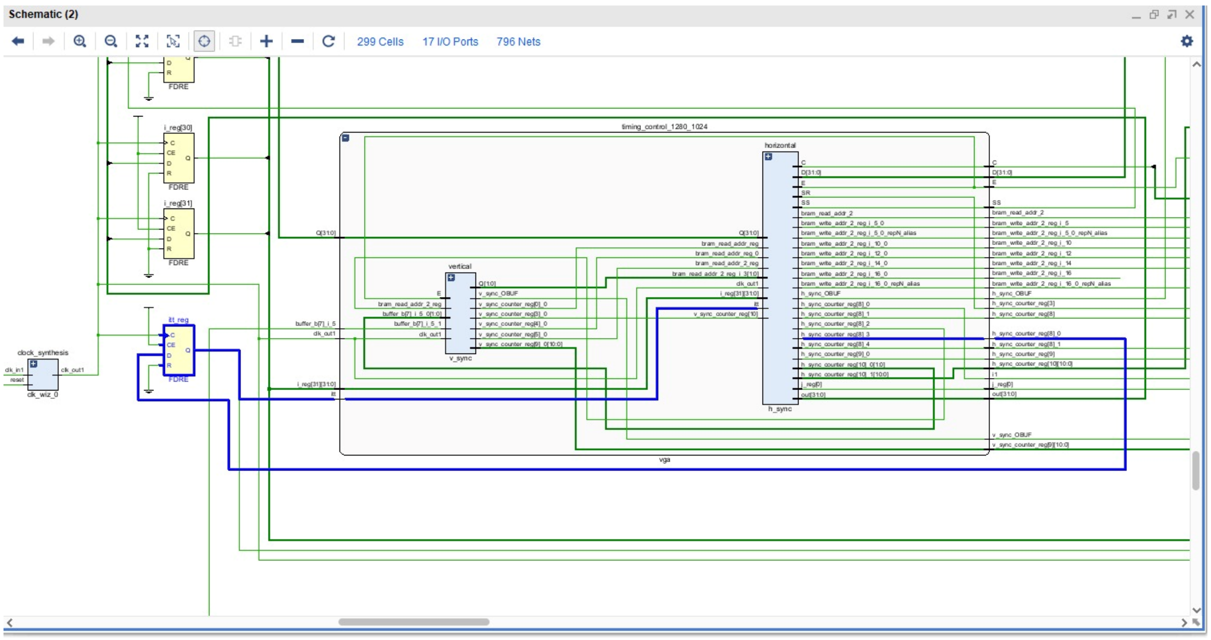
- Worst slack: 2.852 ns for 100MHz clock frequency
- Maximum operating frequency obtained: 139.9 MHz

Post implementation schematic:

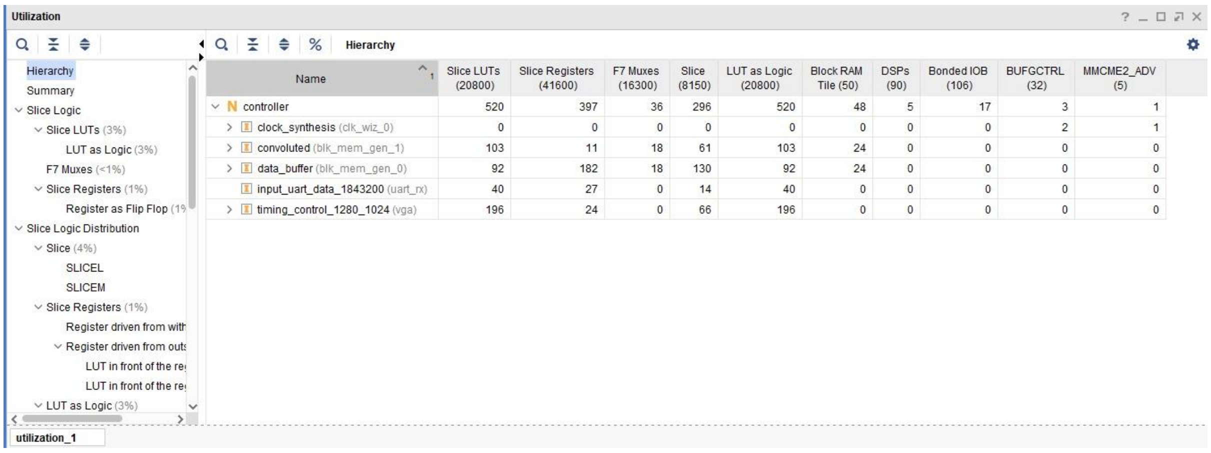


Post implementation results for Image Blurring

Schematic showing critical path

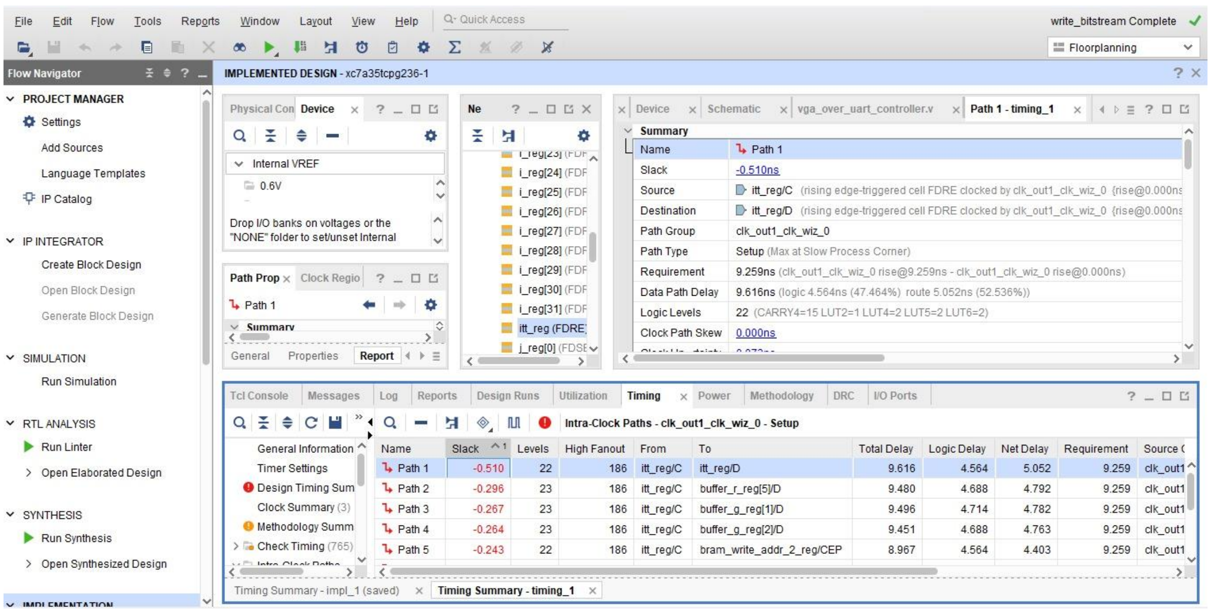


Resource utilization



Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slice (8150)	LUT as Logic (20800)	Block RAM Tile (50)	DSPs (90)	Bonded IOB (106)	BUFGCTRL (32)	MMCME2_ADV (5)
controller	520	397	36	296	520	48	5	17	3	1
clock_synthesis (clk_wiz_0)	0	0	0	0	0	0	0	0	2	1
convoluted (blk_mem_gen_1)	103	11	18	61	103	24	0	0	0	0
data_buffer (blk_mem_gen_0)	92	182	18	130	92	24	0	0	0	0
input_uart_data_1843200 (uart_rx)	40	27	0	14	40	0	0	0	0	0
timing_control_1280_1024 (vga)	196	24	0	66	196	0	0	0	0	0

Timing report showing critical path



IMPLEMENTED DESIGN - xc7a35tcpg236-1

write_bitstream Complete

Path 1 - timing_1

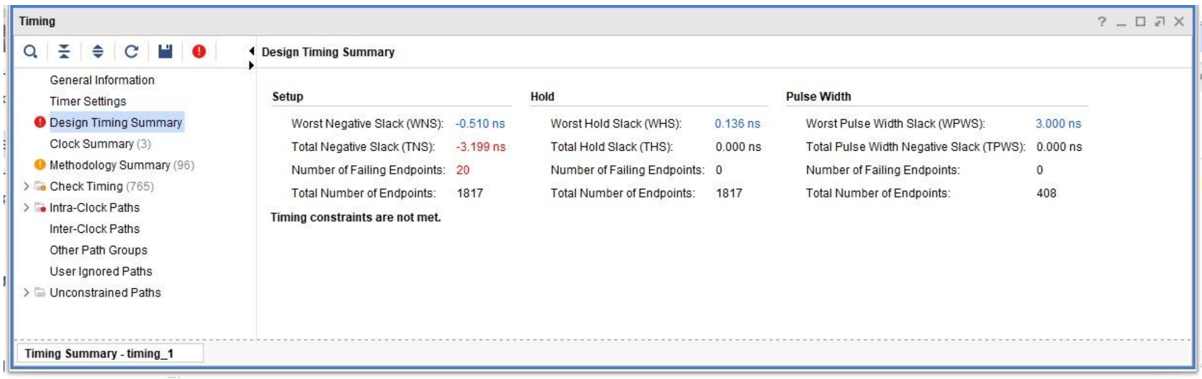
Summary

Name	Path 1
Slack	-0.510ns
Source	itt_reg/C (rising edge-triggered cell FDRE clocked by clk_out1_clk_wiz_0 {rise@0.000ns
Destination	itt_reg/D (rising edge-triggered cell FDRE clocked by clk_out1_clk_wiz_0 {rise@0.000ns
Path Group	clk_out1_clk_wiz_0
Path Type	Setup (Max at Slow Process Corner)
Requirement	9.259ns (clk_out1_clk_wiz_0 rise@9.259ns - clk_out1_clk_wiz_0 rise@0.000ns)
Data Path Delay	9.616ns (logic 4.564ns (47.464%) route 5.052ns (52.536%))
Logic Levels	22 (CARRY4=15 LUT2=1 LUT4=2 LUT5=2 LUT6=2)
Clock Path Skew	0.000ns

Intra-Clock Paths - clk_out1_clk_wiz_0 - Setup

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source C
Path 1	-0.510	22	186	itt_reg/C	itt_reg/D	9.616	4.564	5.052	9.259	clk_out1
Path 2	-0.296	23	186	itt_reg/C	buffer_r_reg[5]/D	9.480	4.688	4.792	9.259	clk_out1
Path 3	-0.267	23	186	itt_reg/C	buffer_g_reg[1]/D	9.496	4.714	4.782	9.259	clk_out1
Path 4	-0.264	23	186	itt_reg/C	buffer_g_reg[2]/D	9.451	4.688	4.763	9.259	clk_out1
Path 5	-0.243	22	186	itt_reg/C	bram_write_addr_2_reg/CEP	8.967	4.564	4.403	9.259	clk_out1

Timing summary

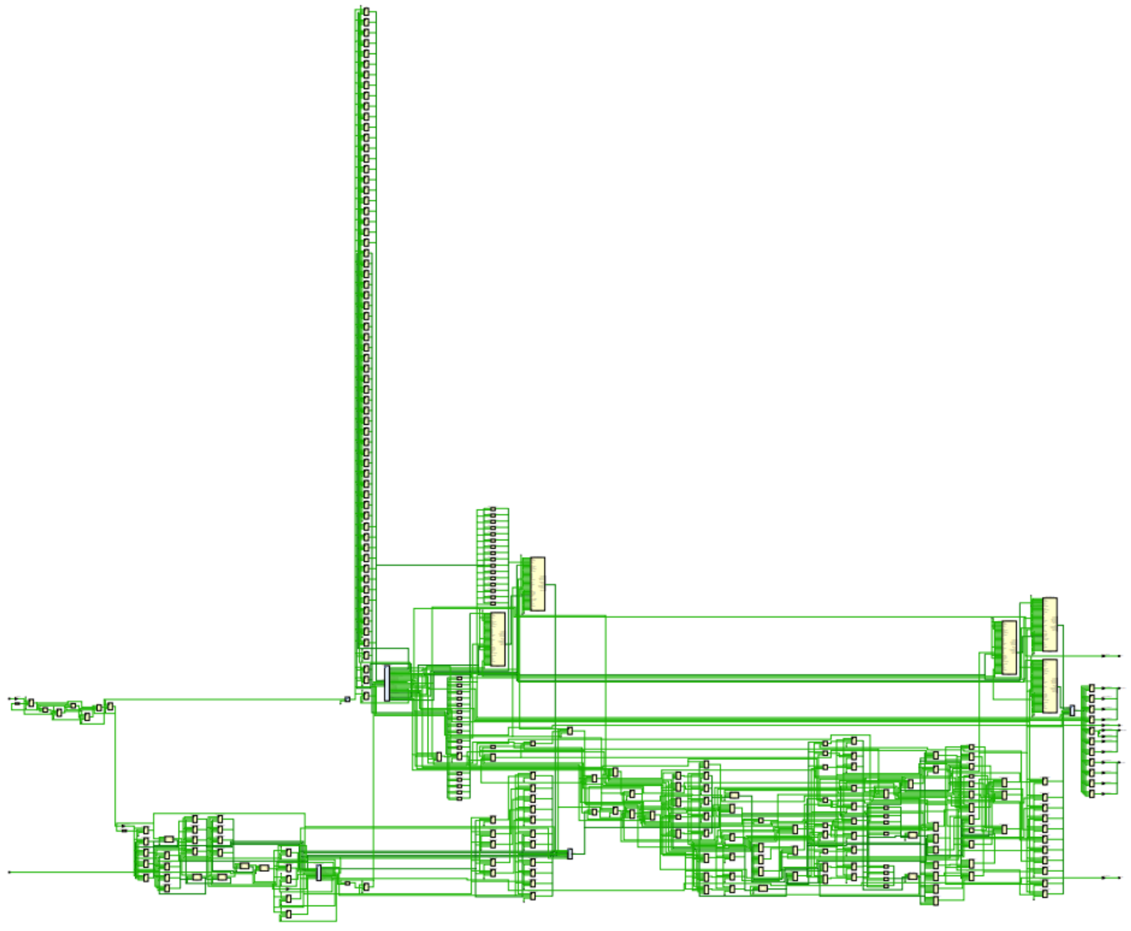


Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-0.510 ns	Worst Hold Slack (WHS):	0.136 ns	Worst Pulse Width Slack (WPWS):	3.000 ns
Total Negative Slack (TNS):	-3.199 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	20	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	1817	Total Number of Endpoints:	1817	Total Number of Endpoints:	408

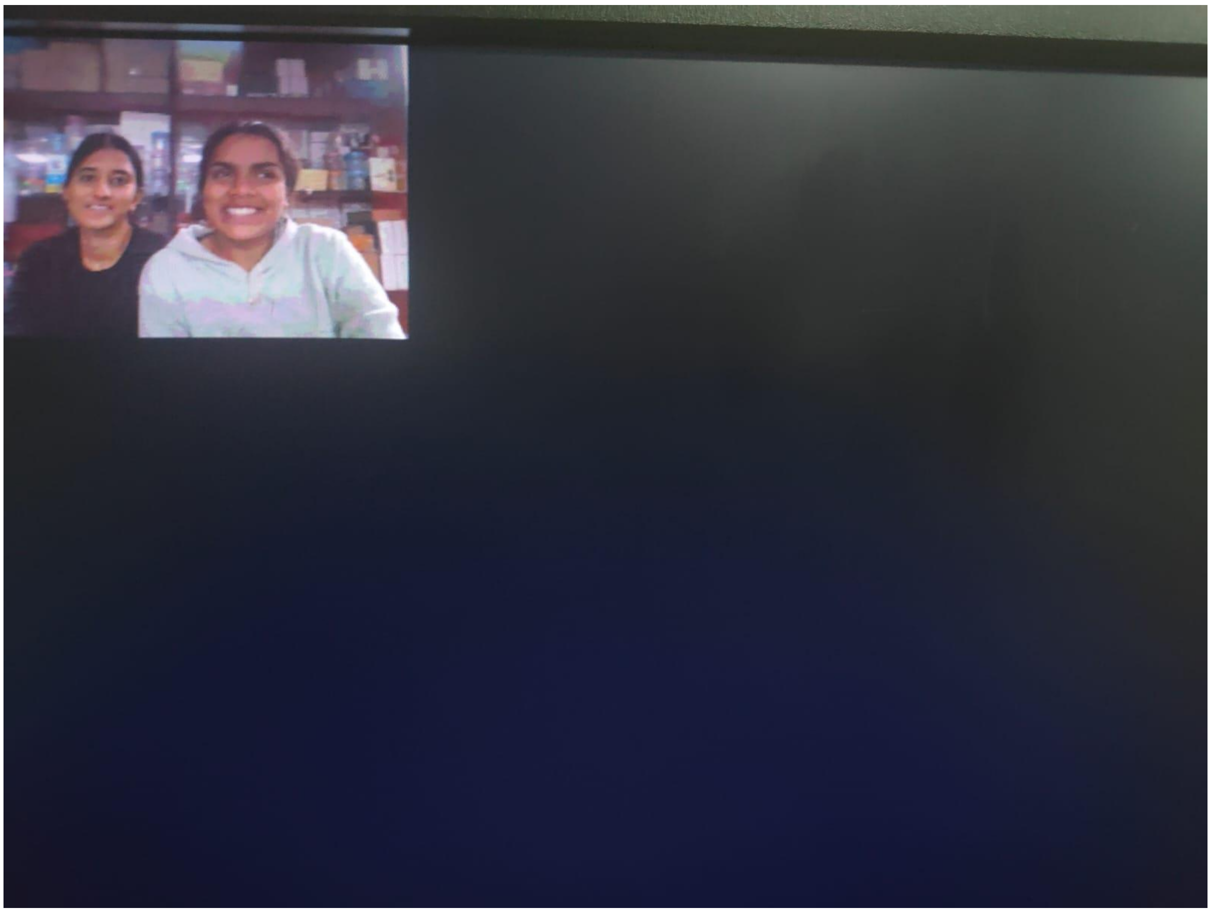
Timing constraints are not met.

Post implementation schematic



Section 3: VGA display

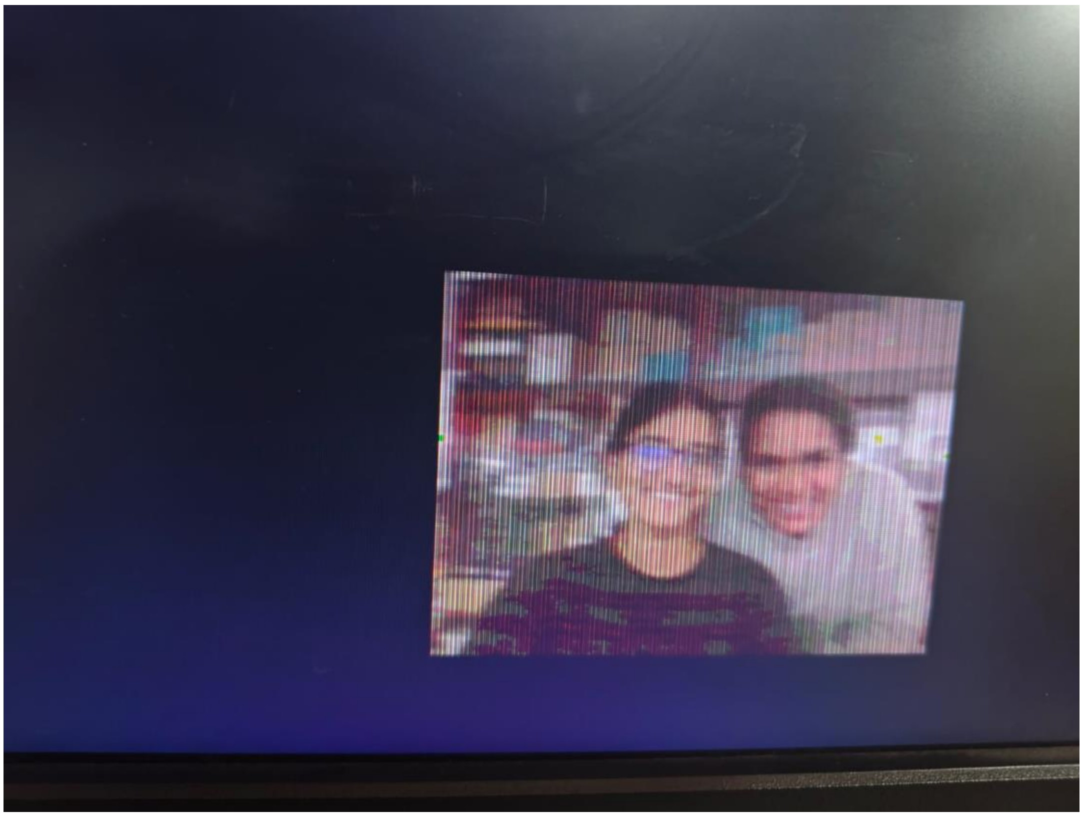
Real time image capture from laptop camera to FPGA to display on monitor through VGA



Video link :

https://github.com/V-Pranathi/FPGA_PROJECT/blob/main/fpga_demo.mp4

Image display after blurring :



Section 4: References

1. https://github.com/Shubhayu-Das/VL504-project/
2. https://www.youtube.com/watch?v=MVu5OAFZhKA&t=375s

Acknowledgments

Jay Shah, MS, IIIT- Bangalore