

COL215 Lab Assignment 8: FPGA Car Game (Part I)

Shivanshu Aryan (2024CS10237), Arnav Meena (2024CS50218)

October 2025

1 Introduction

This report details Part I of Lab Assignment 8, focusing on interfacing the VGA port of the Basys 3 FPGA board to display a static image consisting of a road background and a car sprite. The report covers the design, implementation, simulation, and verification of pixel-level transparency using ROM-based sprites.

2 Design and Implementation

ROM Modules

Two ROM modules are used:

- `bgrom`: A 12-bit wide ROM with depth of 38400, initialized with road background image data.
- `maincarrom`: A 12-bit wide ROM with depth of 224, containing the car sprite pixel data.

VGA Controller

`VGAdriver.v` is used to generate standard VGA signals (HSYNC and VSYNC) for 640x480 resolution at 60Hz.

Transparency Logic

The final pixel output uses conditional multiplexing to handle transparency: if the sprite pixel is pink (binary `12'b101000001010`), the background pixel is displayed instead, otherwise the sprite pixel is shown.

```
always @(*) begin
    if (sprite_pixel != 12'b101000001010)
        vga_output = sprite_pixel;
    else
        vga_output = bg_pixel;
end
```

3 Simulation Results

Simulation verified these key features:

- Correct output from both ROM modules for pixels

- Proper HSYNC and VSYNC signal timing and synchronization
- Validation of transparency logic via MUX
- Final pixel color output confirms correct sprite over background composition

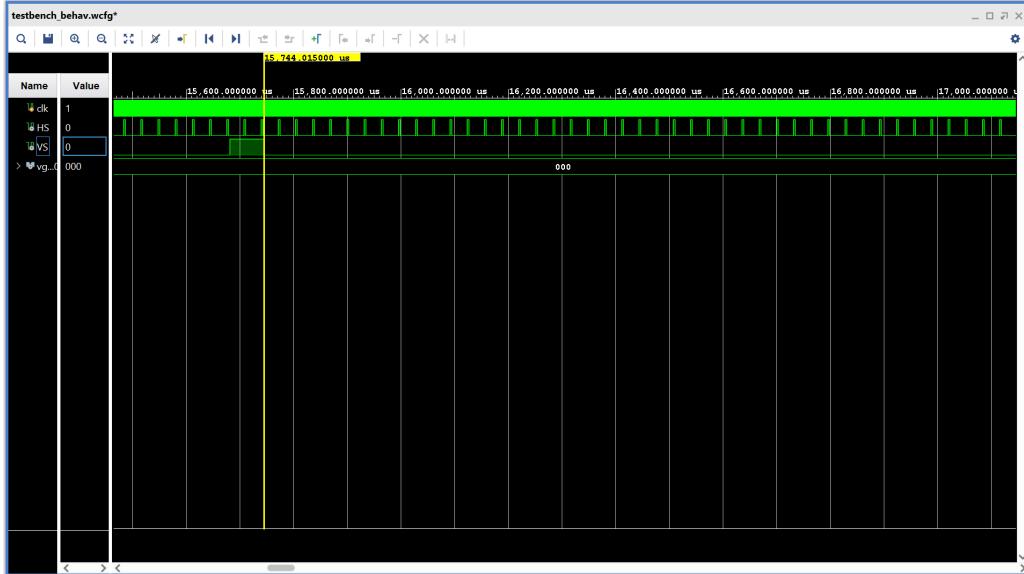


Figure 1: Simulation waveform: VGA signal generation and transparency logic output.

4 HSYNC and VSYNC Timing Verification

Used an analysis submodule in the testbench to measure clock cycle delay between consecutive HSYNC and VSYNC rising edges, confirming timing matches VGA 640x480@60Hz specs.

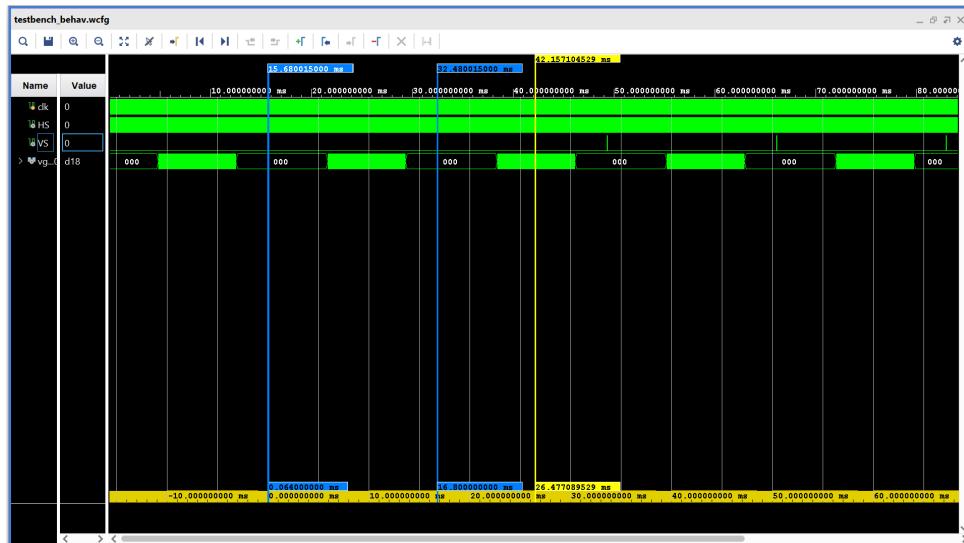


Figure 2: Timing analysis waveform showing VSYNC interval.

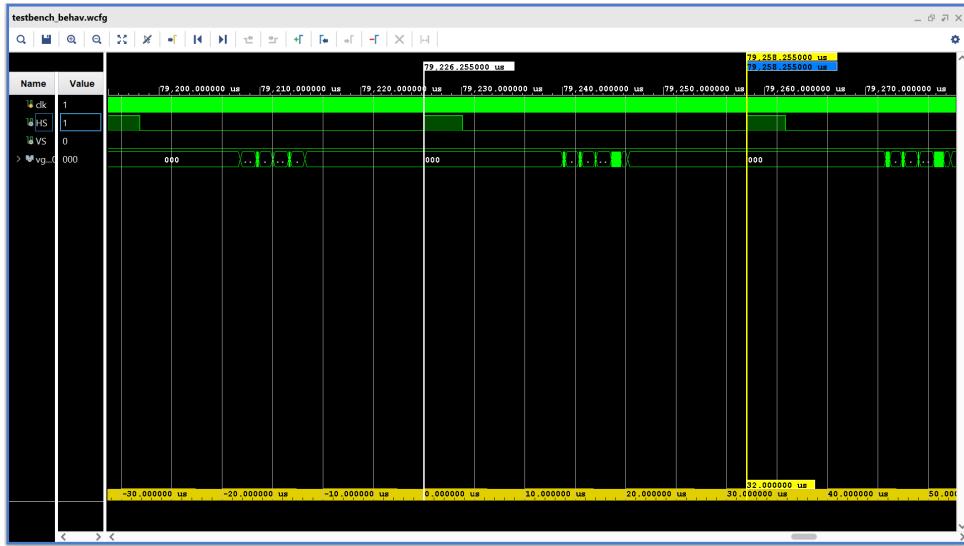


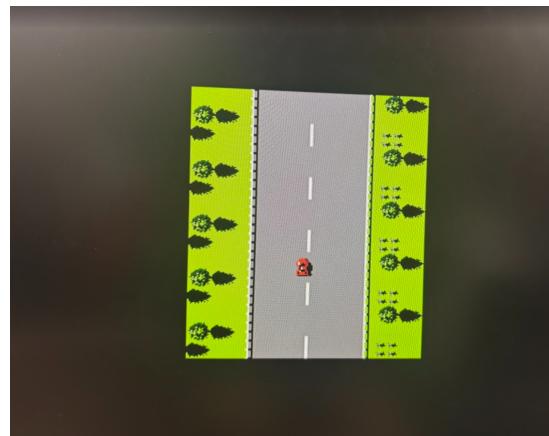
Figure 3: Timing analysis waveform showing HSYNC interval.

Table 1: Measured HSYNC and VSYNC Timing

Signal	Expected Delay	Measured Delay(consecutive)
HSYNC	800 clock cycles	32us
VSYNC	521 HSYNC periods	16.7ms

5 Rendered Image Output

The final output is a static frame combining the road background with a red car sprite. Transparency ensures pink pixels in the sprite are replaced with the road background pixels.



(a) Final VGA output image with transparency applied.

Figure 4: Monitor output showing composite image after transparency logic.

6 Synthesis and Implementation Report

Schematic Diagram

The top-level schematic for Part I of the design illustrates the interconnections between the VGA controller, ROM modules, and output signals. The VGA controller module receives a clock signal and generates the necessary synchronization (HSYNC, VSYNC) and pixel data outputs, while the ROM modules provide pixel data for both the background and the sprite. The final pixel output is determined through a multiplexer that handles sprite transparency.

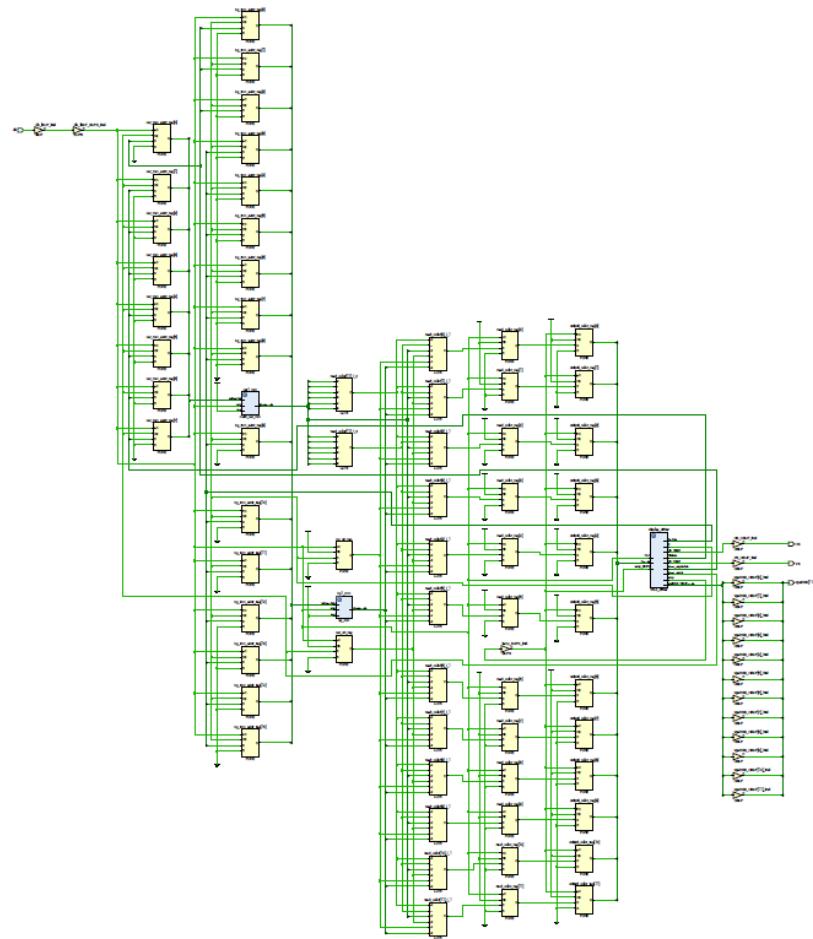


Figure 5: Schematics of Synthesis RTL analysis

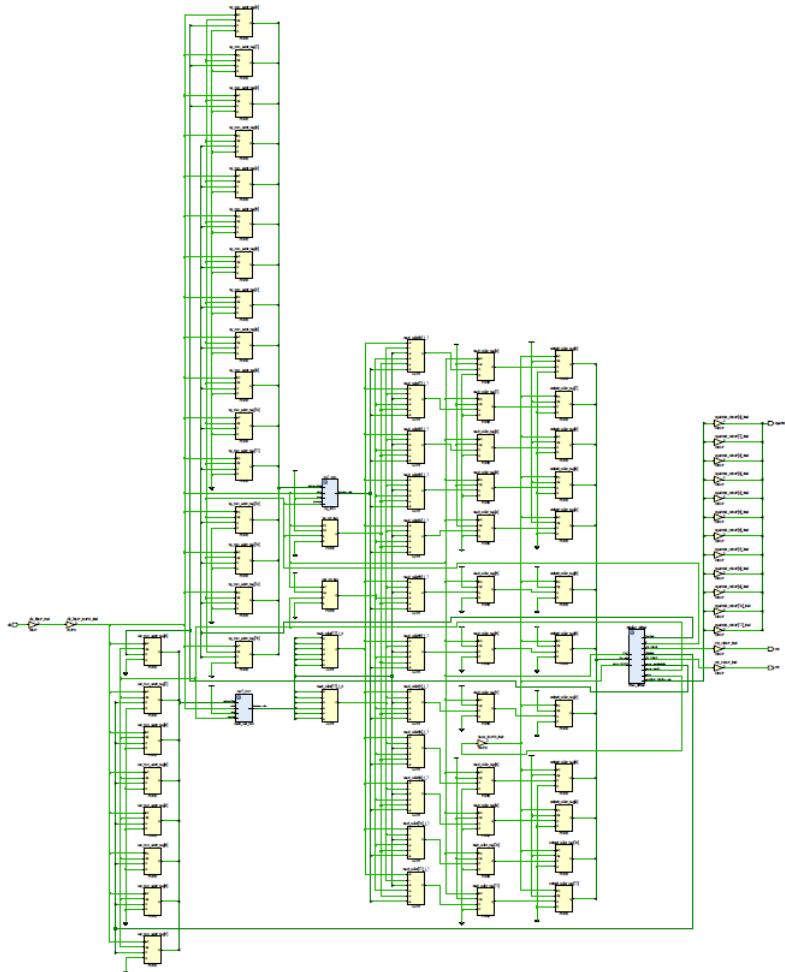


Figure 6: RTL schematics

Table 2: FPGA resource utilization for synthesis and implementation on xc7a35tcpg236-1

Resource Type	Used (Synth)	Used (Impl)	Available	Utilization (%)
Logic Resources				
Slice LUTs	587	586	20,800	2.82%
LUT as Logic	587	586	20,800	2.82%
LUT as Memory	0	0	9,600	0.00%
Slice Registers (FFs)	406	407	41,600	0.98%
F7 Muxes	107	107	16,300	0.66%
F8 Muxes	18	18	8,150	0.22%
Memory Resources				
Block RAM Tiles	0	0	50	0.00%
RAMB36/FIFO	0	0	50	0.00%
RAMB18	0	0	100	0.00%
DSP Resources				
DSPs	0	0	90	0.00%
I/O and Clock Resources				
Bonded IOBs	27	27	106	25.47%
BUFG (Clock Buffers)	1	1	32	3.13%

7 Conclusion

Part I of the assignment successfully demonstrates VGA output programming on the Basys 3 board with pixel-level transparency logic. The design meets timing requirements and produces the expected composite image output.