CECS 361 – Digital Design Techniques and Verification

Spring 2019 – Final Project

By

Justin Maeder - 015906629

May 9, 2019

Final Project: Report

The first purpose of the final project, single player pong game, is to create a Video Graphics Array or VGA synchronization module for a 640-by-480-pixel screen in order to interface our Nexys4 DDR FPGA to a computer monitor. The second purpose was to display one static object: a wall and two animated objects: a paddle and a ball. The third purpose of the final project was to create a self-checking test bench to ensure that all 12 requirements in the VGA sync were met. The final purpose of this final project was to follow the 11 given requirements to ensure that our animated VGA design does what it is supposed to do.

The first step before writing the test bench is making sure you have a complete understanding of how the VGA synchronization and the graphic generation modules function. The six outputs of VGA synchronization unit are the horizontal sync, vertical sync, video on, pixel_x, pixel_y, and p_tick. The only output of the graphics generation unit is the 12-bit graph_rgb. The horizontal sync is the amount of time required to transverse a row, and the vertical sync is the time required to transverse the entire screen.

The first set of requirements are to verify the VGA sync module does what it is supposed to do. The VGA uses a 25 MHz clock to produce 25 M pixels per second for our display. Since our Nexys4 DDR board produces a 100 MHz clock, I created a ticker module that produces a 25 MHz clock. The 25 MHz clock pulses once every four clock cycles from the onboard clock. Although not required, I incorporated the Asynchronous-In Synchronous-Out or AISO module to prevent all flops in our design from reaching a metastable state when reset is released. This module resets all flops in the design in sync when reset is released. Reset brings the VGA Synchronization circuit to a known state with all outputs inactive. The horizontal scan count is updated at the 25 MHz rate and ranges from 0 to 799. The horizontal sync signal is low active and is active from horizontal scan count 656 through 751. The horizontal video on signal is high active and is active from horizontal scan count 0 through 639. The vertical scan count is updated at the completion of a horizontal scan and ranges from 0 to 524. The vertical sync signal is low active and is active from vertical scan count 490 to 491. The vertical video on signal is high active and is active from vertical scan count 0 through 479. The video on signal is high active and is active when horizontal video on and vertical video on are active at the same time. Lastly, the RGB signals are driven while video on is active; therefore, when video on signal is inactive the RGB signals are held at 0. With the 12-bit RGB signals, I can produce 4096 or 2^12 different color options.

The second set of requirements are to verify the graphic generation with an object-mapped scheme does what it is supposed to do. First, I was required to display three objects on the display in the regions specified by these requirements: a Wall, a Bar, and a Ball. Second, each object will have a specified region and a unique color to be selected by the designer. Third, the wall shall occupy the region from horizontal scan count 32 through 35. Fourth, the paddle shall occupy the region from horizontal scan count 600 through 603 and vertical scan count 204 to 276. Lastly, the ball shall occupy the region from horizontal scan count 580 through 588 and vertical scan count 238 through 246.

The finals set of requirements for this design are to verify the paddle and ball are animated properly. The first requirement was to create a 60 Hz enable tick, reft_tick, which shall be asserted once clock period per 60 Hz cycle. Second, replace the paddle top constant with a register that will allow the paddle to roam from the top to bottom of the display. The third requirement is the paddle bottom should be computed using the top register and the paddle length. The fourth requirement is the design shall add two momentary buttons as inputs to control the paddle up and down. The fifth requirement is when a button is pressed during the reft_tick the paddle shall be moved +/- 4 pixels depending on the button. The sixth requirement is if there is no room in the direction requested then the paddle shall not move. The seventh requirement is the ball position shall be tracked by two registers: ball position left and ball position y. The eighth requirement is the area covered by the ball shall be computed using the current position of the ball and the defined size of the ball. The ninth requirement states that the ball shall move with a constant velocity with equal distance in either direction (x = x+/- 1, y = y+/- 1). The tenth requirement is when the ball hits the right wall, the game is over and should restart. The final requirement of the final set of requirements is when the ball hits the paddle, top, bottom, or left wall it shall reflect according to the “Ball Design 3” diagram.

Picture of finished game: