VHDL Infinite Runner

CLAY SHUBERT, ZACH WILLIAMS, ROBERT GRADY WILLIAMS

VGA Implemented in Infinite Runner

- Use coordinate system of display to detect player collisions with obstacles.
  - Potential Issues: Could be potential clipping of obstacles depending on how frequently we check for collisions.
- Some games we are trying to emulate are those such as Flappy Bird or Google's offline dino game

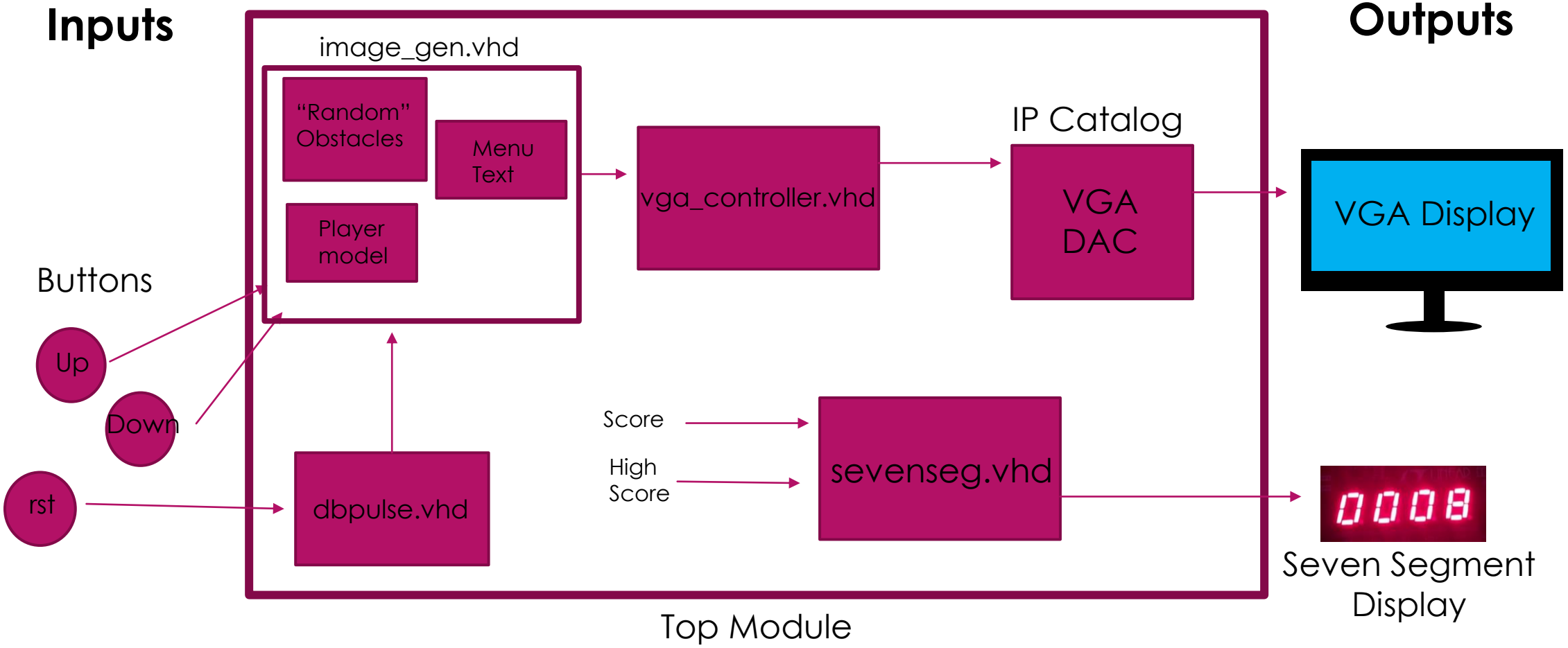
Implementation

- VGA monitor as game display
- Random number generation for seemingly random obstacle placement
- 7 segment display to show score based off running time
- Buttons for continuous up and down movement of player
- Separate debounced button for restart/menu entry
- (If time) Sound generation (background music/game over)

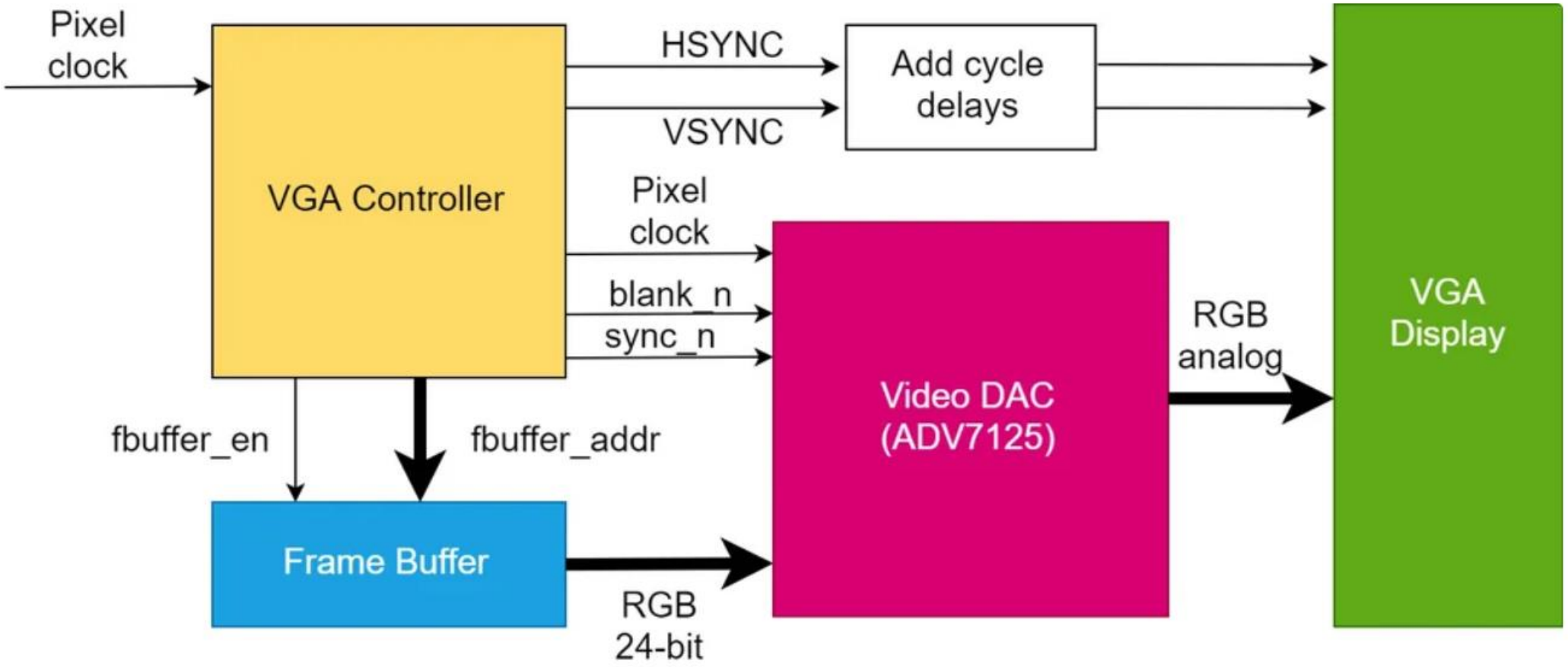
Procedure

- Use Github for version control
- Research VGA display in more detail to have “moving” obstacles on the screen.
- Introduce “random” obstacle placement with random number generation.
- Create button press signals to allow button hold for continuous player movement.
- Once the VGA is implemented. Use seven segment display for running score based on running time. Also display high score on left two bytes of display.
- (If time) Play music or sound effect after multiple of points. Game over sound.

General Module Design



Driving the VGA Display



Driving the VGA Display: Pixel Clock, HSYNC, VSYNC

- Frequency of Pixel Clock needed depends on 3 parameters: **Total Horizontal Pixels, Total Vertical Pixels, Screen Refresh Rate.**
  - *F = THP * TVP * Refresh Rate*
- The VGA Display we are using has a refresh rate 30-60hz.
- The timing of HSYNC and VSYNC signals depend on number of parameters: **Horizontal and Vertical Frontporch, Horizontal and Vertical Backporch, Horizontal and Vertical Display Pixels, Horizontal and Vertical Sync Pulse Widths and Polarities.**
  - Essentially HSYNC and VSYNC are vertical and horizontal timings that are depended on the display sizes.

Driving the VGA: Frame Buffer

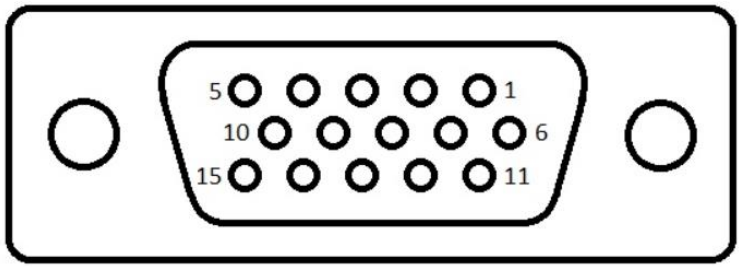
- Each pixel is stored in a Frame Buffer memory location.
- Suppose an image to be displayed is 640x480 pixels.
- Therefore, the Frame Buffer needed is 800x600 = 480000 x 24 bits memory
  - Total size of the memory is 800x600x24 = 1400 kB approx.
  - If black and white image, 800x600x1 = 60 kB approx.
- Block RAMs maybe used to represent a Frame Buffer

Driving the VGA: Video DAC

- Digital to Analog Converter that converts RGB digital data and drives the VGA Display with RGB analog signals at appropriate voltage level.
- Allows signal to be sent to the VGA display.
- These signals are stored in the Frame Buffer (BRAM) which are then sent through the Video DAC to be displayed on the VGA Display.
- This can be included as an IP catalog in our project

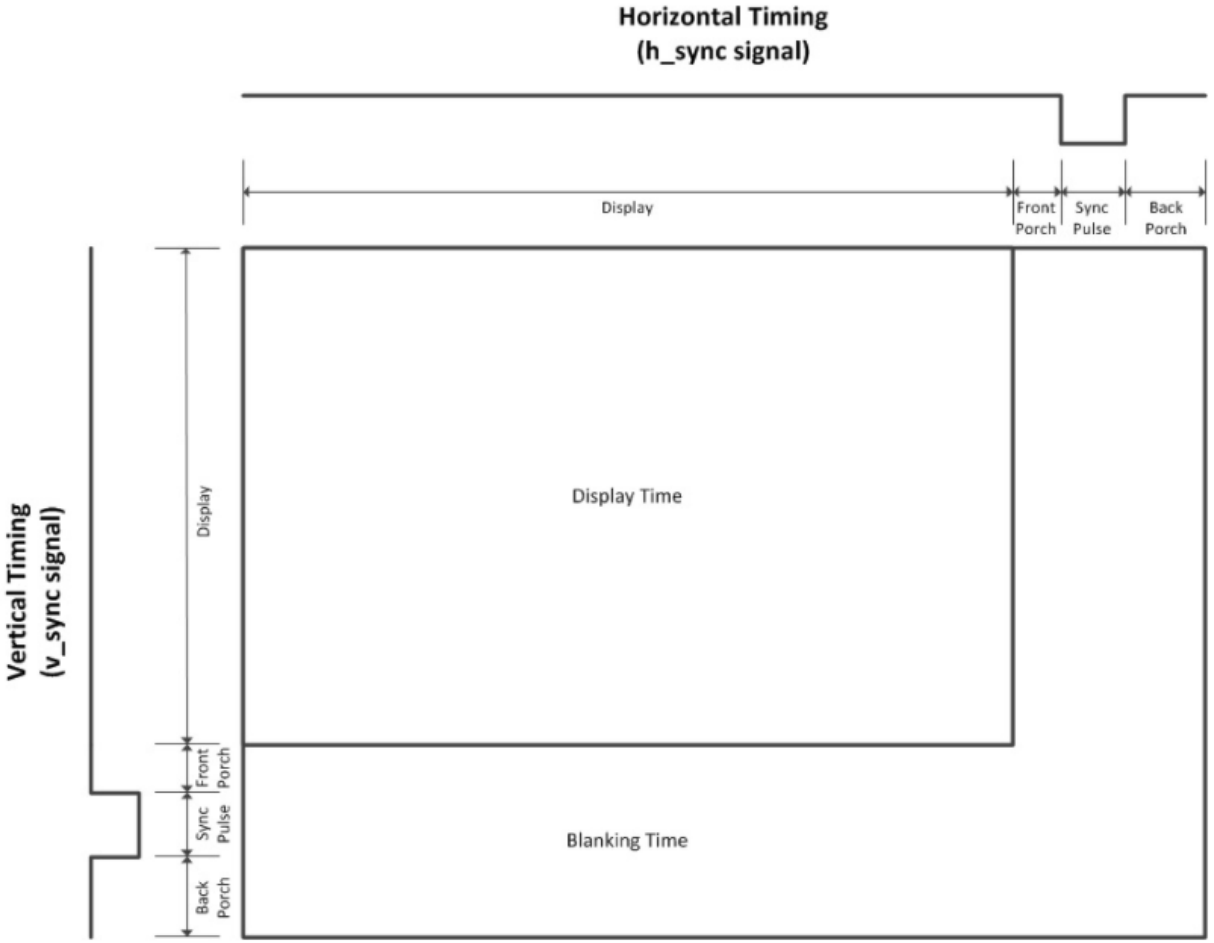
Driving the VGA: VGA Display 15-pin Map

Pin	Signal	Description	Connection
1	R	analog red, 0-0.7V	DAC output
2	G	analog green, 0-0.7V or 0.3-1V (if sync-on-green)	DAC output
3	B	analog blue, 0-0.7V	DAC output
4	EDID Interface	function varies depending on standard used	no connect
5	GND	general	GND
6	GND	for R	GND
7	GND	for G	GND
8	GND	for B	GND
9	no pin	or optional +5V	no connect
10	GND	for h_sync and v_sync	GND
11	EDID Interface	function varies depending on standard used	no connect
12	EDID Interface	function varies depending on standard used	no connect
13	h_sync	horizontal sync, 0V/5V waveform	FPGA output
14	v_sync	vertical sync, 0V/5V waveform	FPGA output
15	EDID Interface	function varies depending on standard used	no connect



- Important ports that are driven in the vhd vga controller are pins 1-3 for RGB analog signals and pins 13-14 for h_sync and v_sync.

VGA Timing (more detail)



- Timing with the VGA display has two dependencies on both vertical and horizontal.
- The controller contains two counters. One counter increments on pixel clocks and controls the timing of the h_sync (horizontal sync) signal, the other increments after each row is completed and controls the timing of v_sync.
  - h_sync
    - By setting it up such that the display time starts at counter value 0, the counter value equals the pixel's column coordinate during the display time. The horizontal display time is followed by a blanking time, which includes a horizontal front porch, the horizontal sync pulse itself, and the horizontal back porch, each of specified duration. At the end of the row, the counter resets to start the next row.
  - v_sync
    - Again, this is set up such that the display time starts at counter value 0, so the counter value equals the pixel's row coordinate during the display time. As before, the vertical display time is followed by a blanking time, with its corresponding front porch, sync pulse, and back porch. Once the vertical blanking time completes, the counter resets to begin the next screen refresh.

References

- VGA Controller: https://forum.digikey.com/t/vga-controller-vhdl/12794
- VHDL Text Display Reference: https://github.com/Derek-X-Wang/VGA-Text-Generator