Ebrahim Simmons - simmoe1 - 400200042
Bilal Yusuf - yusufb1 - 400185626

Lab 5 Report

The new pixel address is set to (0) for R-value. Then we cycle through SRAM going red(0), green even (38400), blue even(57600), green odd (76800) and blue odd (96000). Then increment SRAM by 1 if returning SRAM red address. Blue and green increments after the red SRAM address have been incremented twice. Store the data in the register at each STAM address. Data is stored in the register 3 states after SRAM is changed.

- SRAM address is changed to red address
- SRAM address is changed to green address (even)
- SRAM address is changed to blue address (even)
- SRAM address is changed to green address (odd)
- SRAM address is changed to blue address (old)
    - Red address is read
- SRAM changed to red address
    - Green address (even) is read
- SRAM changed to green address (even)
    - Blue address (even) is read
- SRAM changed to blue address (even)
    - Green address (odd) is read
- SRAM address changed to green address (odd)
    - Blue address (odd) is read
- SRAM address changed to blue address (odd)
    - Red address is read
- Repeat from top

Blue and green addresses are read after 2 address cycles if SRAM addresses are incremented. Data is then sent to VGA like experiment 5. Data is buffered to avoid data being overwritten. Odd pixel data is changed to blue even state while even pixel data is changed to red state. SRAM_data registers are initialized [5,0] has 6 registers with 16 bits each. To bring down resource usage we remove redundant registers by not initializing large buffer registers excluding our SRAM data buffers that we have. According to the compilation report in quartus we use a total of 326 registers most of them due to SRAM_data registers and the rest due to the counter, SRAM_address etc. from the timing analyzer we can see the worst case slack is around 11.3 ns while the average data delay is 8.

Register Chart:

Module (Instance)	Register	Bits	Description
PB_controller (PB_unit)	debounce_shift_reg[0]	10	Shift register used for debouncing the push-button 0
PB_controller (PB_unit)	clock_kHz_div_count	16	Clock division conter needed to reduce the sample rate for the push-buttons from 50 MHz to 1 KHz
VGA_controller (VGA_unit)	H-Cont	10	Column counter - keeps track of the pixel position within a line
VGA_controler (VGA_unit)	V_Cont	10	Line counter - keeps track of the line position within a frame
Top-level experiment 1	SRAM_address	18	Address register used for accessing the external memory (i.e., the external SRAM organized as 218 x 16)
Top-level experiment 1	VGA_sram_data[5]	16	Buffer register - holds the Red and Green, Blue data for the even and odd pixels before it is ready to be transferred to the VGA controller.
Top-level experiment 1	state	4	State register used by the FSM that coordinates the data transfers to/from the external SRAM and to the VGA controller