Comp Eng 3DQ5 Lab 3

Group 44

Sakib Reza (rezas2) – Dharak Verma (vermad1)

2019-10-03

To display the welcome screen, and essentially all the objects and colours, we got rid of the assign statements for VGA_red, VGA_green, and VGA_blue and created a new always_ff block to be updating based on what should be showing on the screen. Inside here there is a variable welcome_screen, and another variable, h_or_v which are used to know if the system should be displaying the welcome screen and which orientation the horizontal bars are. And If-elseif-else statement is used to create the 8 bars, both horizontally and vertically. To ensure they rotate every 60 frames, we implemented a counter that updates with V_sync, up to 60 times. Once that milestone is reached, h_or_v is flipped and the other mode is displayed. To start the game up from the welcome screen, we reused the PS2_controller code and copied some of the .qsf file from Lab 2 to start the game on button push.

We then used the PS2_code signal to change the bar movement to “A” and “S”. Since it was constantly receiving a make code if the button is held down, everything works as expected.

Once the number of lives reaches 0, we enter the game_over state by changing the value to 1’b1. In this state, nothing shows except the intended messages about the last game score, number of games played since async reset, the highest score since async reset, and time left until the game resets. All these numbers are stored in variables and modified in the block where the ball location is updated. These messages are coded in an always_comb with the Lives and Score to be displayed during the game, but only appear during game_over. Every register storing data like the lives left, current game score, highest game score, were changed to two different 4-bit registers supported up to 99 with an MSB and LSB to be used for BCD conversion. We reused the HEX_to_char_rom to print these values.

The timer for the game ending messages was created using a separate counter that decrements every 60 frames, from 15 to 0. This is displayed to the screen by storing this information into two registers and displaying their MSB and LSB in BCD format.