COE3DQ5 – Lab #3 Report

Group 14

Mengjia Li (400011362)
Nicole Wu (400089778)
lim14@mcmaster.ca
wuz78@mcmaster.ca
Sep 28th, 2019

a. Welcome screen & vertical/horizontal colour bar alternation

The period of the H_SYNC is about 32us. There are 524 lines for the V_SYNC signal to process. Therefore, it takes 16.768ms to display one frame. To switch the scene every one second, 60 frames are displayed. *Counter* is the inside the *VGB_Controller* to record the number of periods that the V_SYNC signal gets. Once *counter* gets to 60, a *welcome_flag* output will be flipped and send out to the main programme. The horizontal colour bar is simply determined by the Y position. In the vertical colour bar, the specific position range is assigned to each colour.

b. Game activation by keyboard & A/S game control

game_start is used to detect the first input from the PS2. Once the code is detected (if statement is true), the game_start will always be assigned to 1 unless the system is reset again. *A_Key* and *S_key* are used to detect the A/S key. *PS2_controller* compare the *PS2_shift_reg* value with A/S PS2 code. To determine if the key is kept pressed, *A_Key* and *S_Key* are "and" with *PS2_make_code*.

c. 15 seconds messages display after game over

A FSM FLASH_state is used for the counter to switch from S_FLASH_INIT (inactivated) to S_FLASH_DELAY (activated) two states. During S_FLASH_INIT, *counter* is asleep and reset to 0. When *lives_0* and *lives_1* are both 0, which means game over, the state will change from INIT to DELAY and a 5 bit counter will start counting up to 16. When it reaches 16, state switches back to INIT and counter is reset to 0.

Messages display: within the VGA_RGB *always_comb* logic, *lives_0* and *lives_1* are used to decide whether the message will be displayed or normal game is displayed. Based on the truth table of *OR* gate, when (*lives_0* || *lives_1*) is true (the game is not over), game *objects/bar* and *score/lives* will be on, while *lives_0* and *lives_1* are both 0, the three messages below is shown until counter is 15 and the system is initialized. Within the display text *alway_comb* logic a separate if statement is used for this message.

1. *LAST GAME SCORE WAS XX*
   XX is assigned to signals *score_0* and *score_1*, which is two BCD counters that BCD1 is increment based on BCD0.
2. *GAME YY SCORE ZZ*
   YY is assigned to signals *highest_game_0* and *highest_game_1* and *ZZ* is assigned to signals *highest_score_0* and *highest_score_1*. A parallel running *always_ff* logic is implemented with 3 conditions deciding if the current game has the highest score and update the *highest_score* and *highest_game* counters if possible.
3. *TIME LEFT WW*
   WW is assigned to signals *counter_0* and *counter_1*. These two signals are also two BCDs that are initialized to 15. Within the FSM FLASH_state, during S_FLASH_DELAY state, while the counter is counting, *counter_0* and *counter_1* are also doing subtraction logic until 00.