LAB 6 REPORT

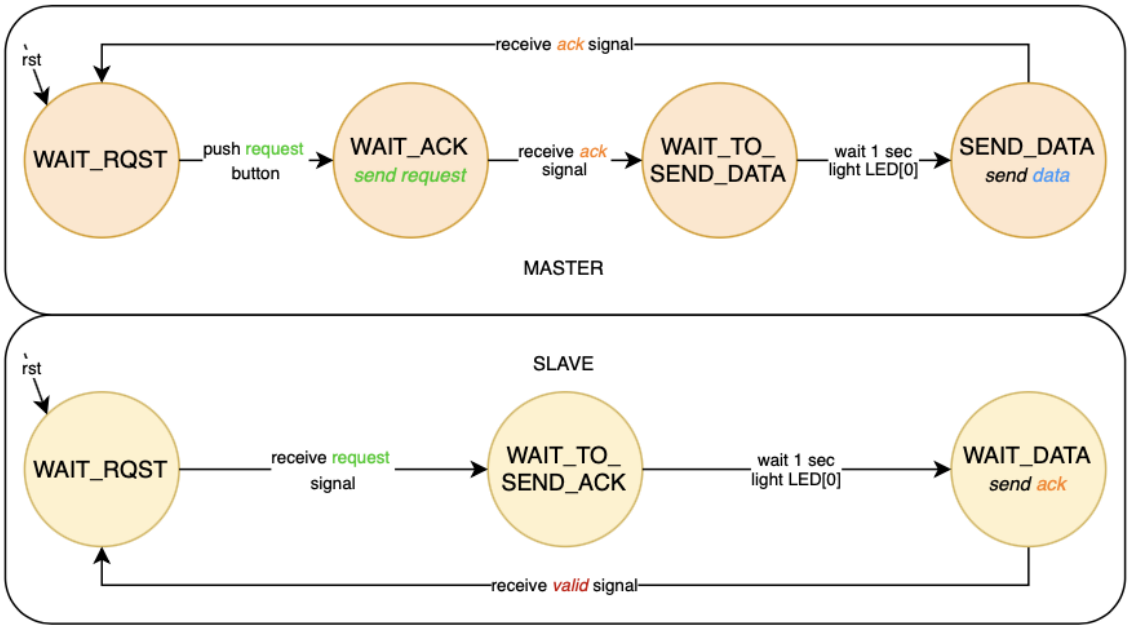
(FPGA) Dual FPGA communication

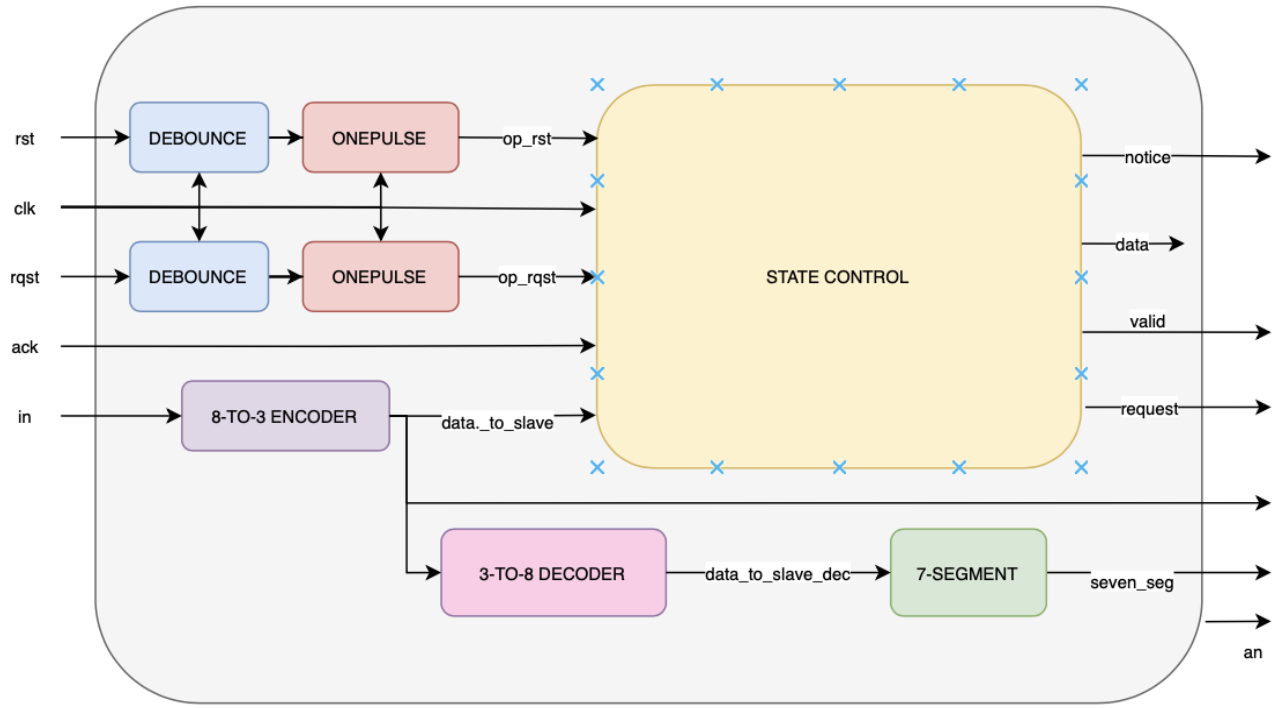
(FPGA) The Slot Machine

Team 37:

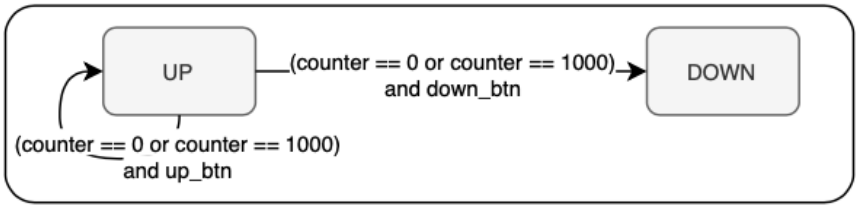
1. 徐美妮 Mary Madeline Nicole 109006205
2. 林之耀 Kevin Richardson Halim 109006277

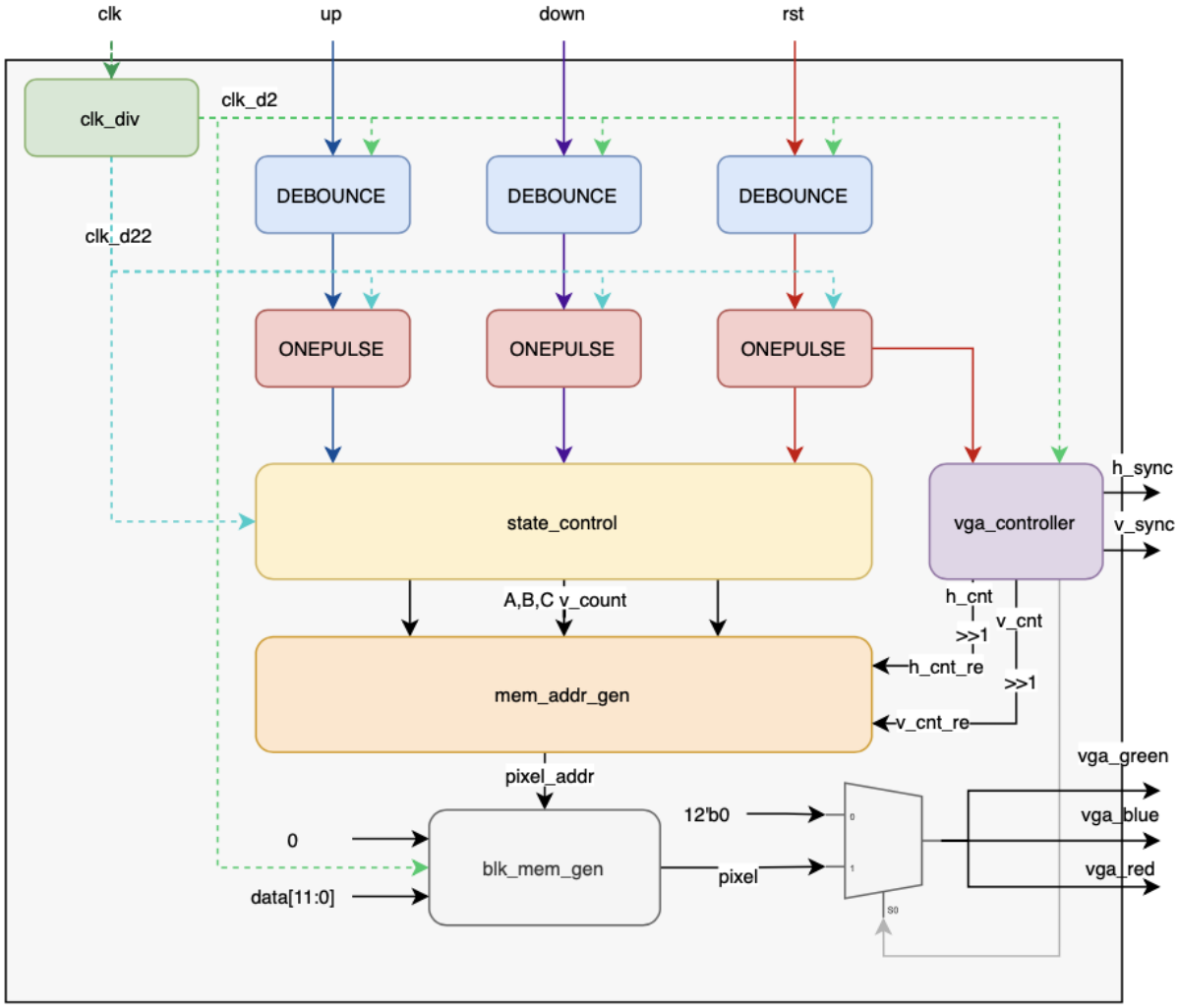
I. Block diagrams and state transition diagrams





1.1. Dual FPGA Communication





1.2. The Slot Machine

II. Design Process

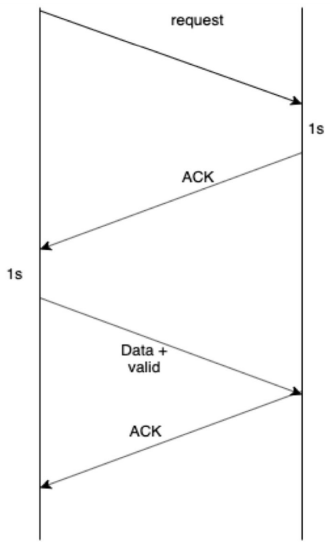
a) Dual FPGA Communication

Communication between FPGA using the pmod port and jumper as the intermediate. This design has a purpose to build a method to deliver a reliable data transmission between FPGAs. Designing the method by stopping and waiting for the requirement signal, such as the ***request***, ***ack***, ***data***, ***valid***.

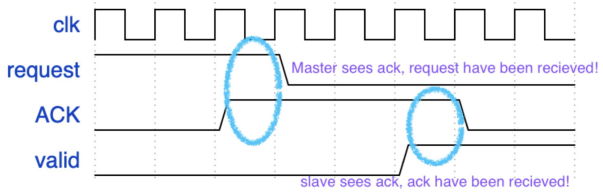
The code which has been provided by the TAs contained the completed ***master*** (data sender FPGA) and partially completed ***slave*** (data reciever FPGA) design. Using state and the input/output communication between two designs will carry out all the requirements to be a reliable communication method.

```
// Master's param
parameter state_wait_rqst = 2'b00;
parameter state_wait_ack = 2'b01;
parameter state_wait_to_send_data = 2'b10;
parameter state_send_data = 2'b11;
```

```
// Slave's param
parameter state_wait_rqst = 2'b00;
parameter state_wait_to_send_ack = 2'b01;
parameter state_send_ack = 2'b10;
parameter state_wait_data = 2'b11;
```



The input of ***request***, ***ack***, ***valid*** will determine the success of communication transfer. The *"hand-shaking"* method is a method which the ***master*** send a signal named ***request*** to the ***slave*** and waits the slave to send back a signal named ***ack***. When the *"hand-shaking"* method has been executed properly. The ***data*** will be transferred until the ***valid*** signal given to the slave from the master.



2.1. Hand-shaking Method

b) The Slot Machine

The slot machine module given by the TA plays in downward direction as the start button is pressed, and is able to be reset. However, the given slot machine's design cannot play continuously after a performance is pulled off. It could only be played again after pressing reset after every run. Therefore in this module we were meant to implement the other direction, which is the upward direction of the slot machine, and should modify the code so that the slot machine can be played again without reset. Using extra state and some if-else statements, the requirement can be fulfilled.

Up-down direction states were needed in this design. Therefore, we design this module to fetch the input+ direction in a register named ***dir*** stands for direction and some parameter to denote up named ***UP*** (1'b1) and down named ***DOWN*** (1'b0). When ***dir*** is up, then the vertical counter will be subtracted by the state for each image and vice versa when the ***dir*** is down.

Each image in this design is represented as A, B and C. Therefore the variables named ***A_v_count***, ***B_v_count***, ***C_v_count***, ***A_state***, ***B_state***, and ***C_state***. We bound each counter to not pass more than 240 or pass below 0. The global counter is also needed to keep track of the global timing to pull off the animation differently, but keeping sure every action is in sync. The variable named ***counter*** is used to handle this. The original design did not reset the ***counter*** back to 0, therefore another performance can not be executed after a perform has been pulled off.

III. Design Test

We tested the success of our program by testing it directly on the FPGA. For the master and slave, we try to send the request signal one by one from 0 to 7, and reset it. While for the slot machine, we tested it by directly playing the slot machine on the FPGA connected with the screen using the VGA cable.

After testing and demo, our module works as expected.

IV. What we have learned from Lab 6 FPGA

- ➔ Connect two FPGAs for dual communication
- ➔ Using VGA to display module

V. List of contributions

1. Mary Madeline Nicole 徐美妮 109006205
    - Module Chip2Chip
    - Draw state diagram and block diagram
    - Make report
2. Kevin Richardson Halim 林之耀 109006277
    - Module Slot Machine
    - Make report