



Lab6_Team32_Report

組別：Team32

組長：蘇勇誠 (108062373)

組員：張晏瑄 (108062273)

Dual FPGA Communication

Design

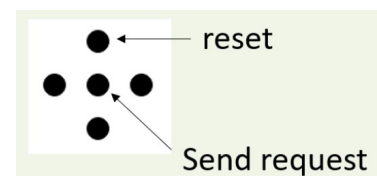
以網路中 **2-phase handshaking** 的概念去設計此題。要傳送資料前，Master 會先傳送 request 信號給 Slave，若有收到訊號，此時 Slave LED 燈亮一秒後會回傳 ACK 給 Master，Master 若有接收到訊號，則其 LED 燈也會亮一秒，並將資料傳送給 Slave，Slave 若有收到訊號，則回傳 ACK 完成此次資料傳輸。要注意的是，傳輸端在傳訊號時要一直傳送直到有收到接收端的回傳訊號再停止。

Detail

Master

Master 為老師提供的 Sample Code，並無修改

- **top:** 將 FPGA 的 button 所產生的 `rst_n` (傳至其他 module 時有做 inverter) 和 `request` 做 debounce 和 onepulse、接收 FPGA 上的 switches 所產生的 data (0~7)，以及分配訊號要傳給哪個 module



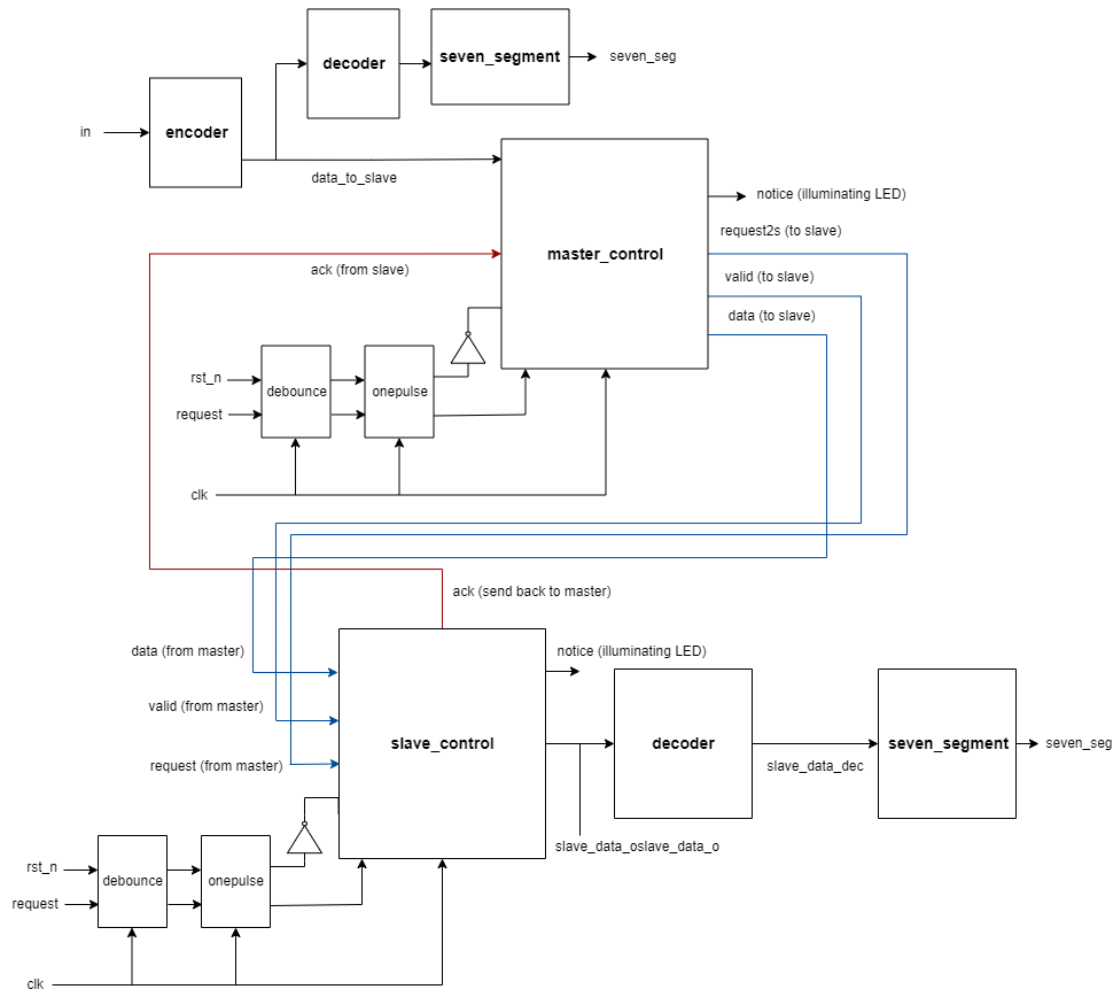
- **counter:** 計算 LED[0] 要亮的時長，當 `start` 為 1 時開始數，到達 1 秒時，則將 `done` 設為 1 並回傳信號
- **encoder & seven_seg:** 處理 switches 和 7 Segment Display 數字顯示
- **master_control:**
 - request: 為當 user 按下按鈕時產生的訊號
 - ack: 由 slave 傳回來的訊號，當 ack 為 0 時，表 slave 還未收到 `request2s` 信號，故繼續傳送直到收到 ack 為止
 - notice: 收到從 slave 傳回來的 `ack` 信號後會啟動 counter，當數了一秒時，`notice` 會設為 1，此時 LED[0] 會亮起一秒

- data: 要傳給 slave 的 data (0~7)
- valid: 當 counter 數了一秒後，`valid` 會被設為 1，此時表可以開始傳送 `data` 給 slave
- request2s: 對 slave 要求要開始傳送 data

Slave

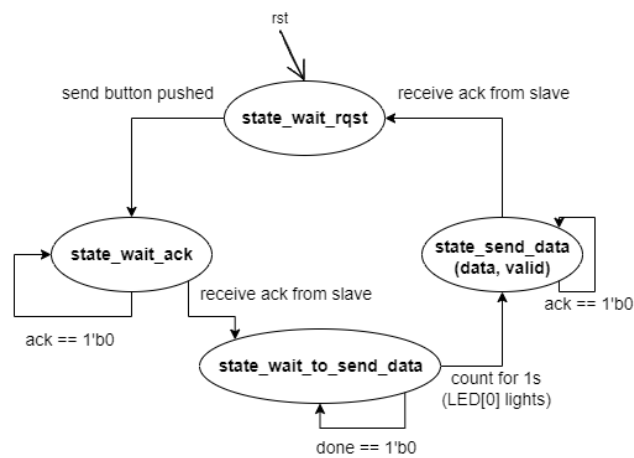
- **top:** 將 FPGA 的 button 所產生的 `rst_n` (傳至其他 module 時有做 inverter) 做 debounce 和 onepulse，以及分配訊號要傳給哪個 module
- **counter:** 計算 LED[0] 要亮的時長，當 `start` 為 1 時開始數，到達 1 秒時，則將 `done` 設為 1 並回傳信號
- **seven_seg:** 在 7 Segment Display 上顯示從 master 傳過來的數字
- **slave_control:**
 - notice: 當 counter 數了一秒時，`notice` 會設為 1，此時 LED[0] 會亮起一秒
 - ack: 當收到 master 的訊號後，會回傳 `ack`
 - data: master 傳過來的資料數字 (0~7)
 - start: 當收到從 master 傳過來的 `request` 信號會將 `start` 設為 1，此時啟動 counter 數一秒

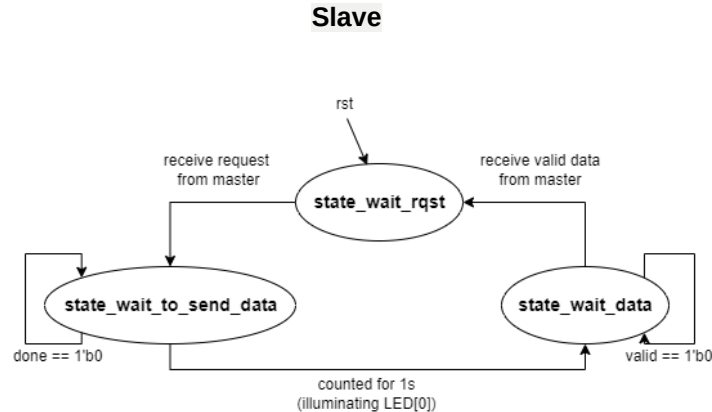
Block Diagram



State Transition Diagram

Master





The Slot Machine

Design

Sample code的slot machine只有下拉功能，而下拉的功能實作於state_control中，state_control的output—A_v_count、B_v_count、C_v_count為up counter。因此，若要設計上拉的slot machine只需要將state_control的output—A_v_count、B_v_count、C_v_count改為down counter即可。最後，再將A_v_count、B_v_count、C_v_count改為同時可進行up count和down count，並利用mux選擇此時要用up counter或down counter，即可實現同時有上拉及下拉功能的slot machine。

Detail

- top module：串接所有module，詳細串接方式見block diagram。
- state_control：

每次要進行上拉或下拉時，A_state、B_state、C_state會依序從STOP → SLOW → MID → FAST → MID → SLOW → STOP。

而進行上拉或下拉時，A_state為最慢回到STOP的state，因此若要進行下一次上拉或下拉需等待A_state回到STOP state（counter≥1000），此功能設計見底下 combinational circuit of sel。

Sample code的A_v_count、B_v_count、C_v_count為upward counter，因此sample code的slot machine只有下拉功能。故若要實現上拉功能，需讓A_v_count、B_v_count、C_v_count為downward counter。最後，再利用mux（利用sel訊號選擇），選擇A_v_count、B_v_count、C_v_count要進行downward counting或upward counting，此功能設計見底下combinational circuit of up counter and down counter和combinational circuit of selecting up_counter or down_counter。

- sequential circuit

```

always@(posedge clk)begin
    if(rst)begin
        A_state <= `STOP;
        B_state <= `STOP;
        C_state <= `STOP;
        counter <= 10'd0;
        A_v_count <= 10'd0;
        B_v_count <= 10'd0;
        C_v_count <= 10'd0;
        sel <= 1'b0;
    end
    else begin
        A_state <= next_A_state;
        B_state <= next_B_state;
        C_state <= next_C_state;
        counter <= next_counter;
        A_v_count <= next_A_v_count;
        B_v_count <= next_B_v_count;
        C_v_count <= next_C_v_count;
        sel <= next_sel;
    end
end
end

```

- combinational circuit of up counter and down counter

Up_counter : $\text{next_X_UP_count} = \text{X_v_count} + \text{X_state}$ (up_counter為v_count + state)

Down_counter : $\text{next_X_DOWN_count} = \text{X_v_count} - \text{X_state}$ (down_counter為v_count - state)

```

assign next_A_UP_v_count = (A_v_count + A_state >= 10'd240)? A_v_count + A_state - 10'd240: A_v_count + A_state;
assign next_B_UP_v_count = (B_v_count + B_state >= 10'd240)? B_v_count + B_state - 10'd240: B_v_count + B_state;
assign next_C_UP_v_count = (C_v_count + C_state >= 10'd240)? C_v_count + C_state - 10'd240: C_v_count + C_state;

```

```

//A_v_count - A_state < 10'd0, B_v_count - B_state < 10'd0, C_v_count - C_state < 10'd0
assign next_A_DOWN_v_count = (A_v_count < A_state)? A_v_count - A_state + 10'd240: A_v_count - A_state;
assign next_B_DOWN_v_count = (B_v_count < B_state)? B_v_count - B_state + 10'd240: B_v_count - B_state;
assign next_C_DOWN_v_count = (C_v_count < C_state)? C_v_count - C_state + 10'd240: C_v_count - C_state;

```

- combinational circuit of selecting up_counter or down_counter :

當sel == 0, 為up_counter

當sel == 1, 為down_counter

```

always @(*) begin
    if(sel == 1'b0) begin
        next_A_v_count = next_A_UP_v_count;
        next_B_v_count = next_B_UP_v_count;
        next_C_v_count = next_C_UP_v_count;
    end
    else begin
        next_A_v_count = next_A_DOWN_v_count;
        next_B_v_count = next_B_DOWN_v_count;
    end
end

```

```

        next_C_v_count = next_C_DOWN_v_count;
    end
end

```

- combinational circuit of sel :

當counter == 0 || counter ≥ 1000時，若start_up == 1（press the start_up button），則設sel為0（此時使用up_counter）

當counter == 0 || counter ≥ 1000時，若start_down == 1（press the start_down button），則設sel為1（此時使用down_counter）

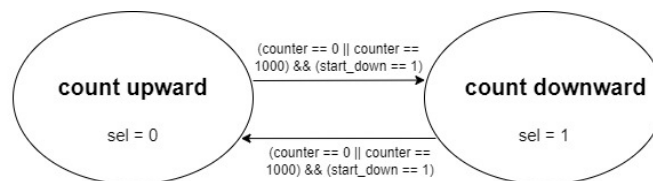
```

always @(*) begin
    if((counter == 10'd0) || (counter >= 10'd1000)) begin
        if(start_up == 1'b1)
            next_sel = 1'b0;
        else begin
            if(start_down == 1'b1) begin
                next_sel = 1'b1;
            end
            else begin
                next_sel = sel;
            end
        end
    end
    else begin
        next_sel = sel;
    end
end

```

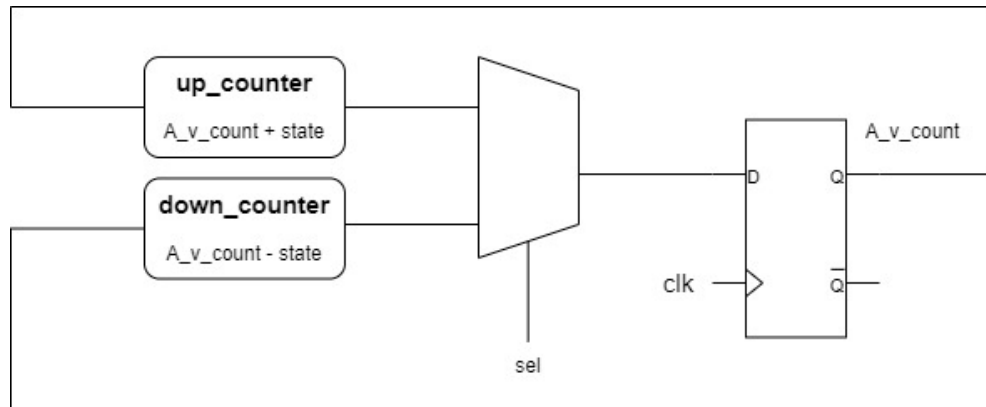
- mem_addr_gen：將h_cnt、v_cnt配合A_v_count、B_v_count、C_v_count map到pixel address。
- blk_mem_gen_0：Access FPGA的memory。
- vga_controller：output hsync、vsync（用以告訴FPGA開始、結束Display）signal給Screen，output valid（enable）signal給Screen，output h_cnt、v_cnt給mem_addr_gen以計算出data存在memory的位置。

State Transition Diagram

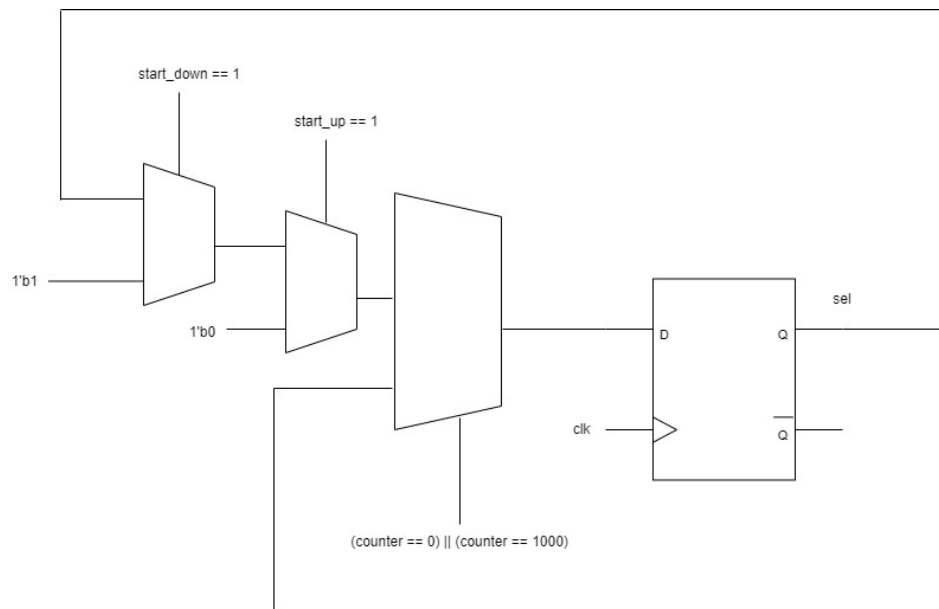


block diagram

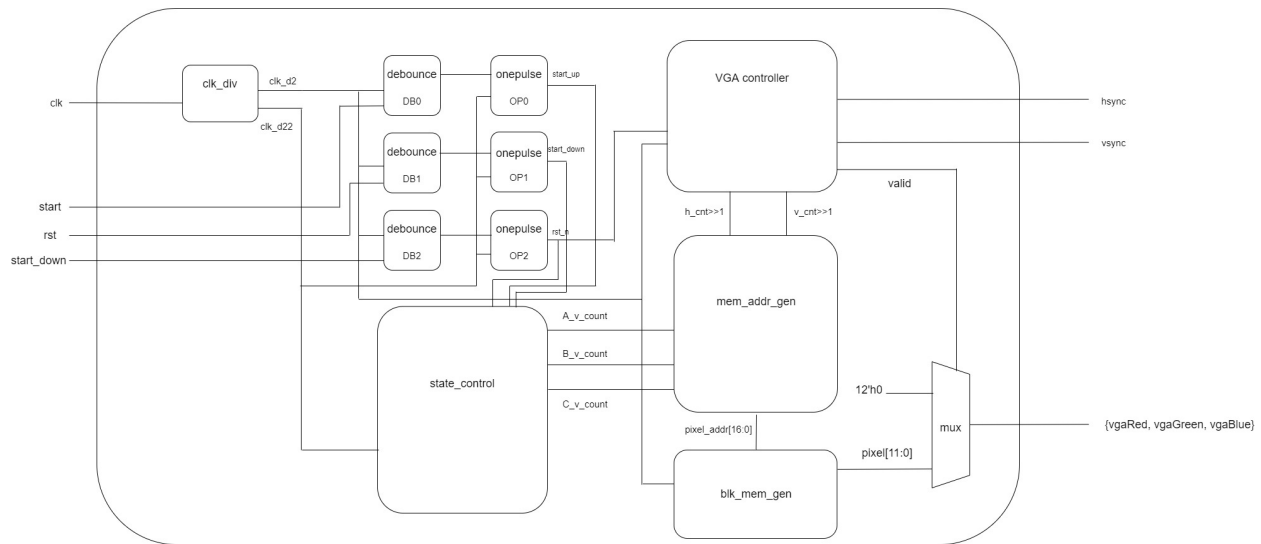
circuit of state_control A_v_count、B_v_count、C_v_count（以下僅列出A_v_count，B_v_count、C_v_count為相同概念）



circuit of sel



circuit of top module



Learning

- 接線的時候避免搞亂線的順序，故有使用 excel 檔紀錄哪個 data 是屬於哪個顏色的線

	A	B	C	D
1				
2	A14	紅		data_in[2]
3	A16	黃		data_in[1]
4	B15	紫		data_in[0]
5		白	GND	
6	A15	褐		request
7	A17	橘		ack
8	C15	藍		valid
9		綠		

- 了解如何利用FPGA將畫面過VGA顯示在螢幕之流程，如：hsync、vsync signal在一開始要顯示以及顯示完成的時候會發出signal
- 了解兩個FPGA互相溝通的機制
- 了解如何使用FPGA的memory

分工

- 蘇勇誠 (108062373)

The Slot Machine

- 張晏瑄 (108062273)

Dual FPGA Communication