

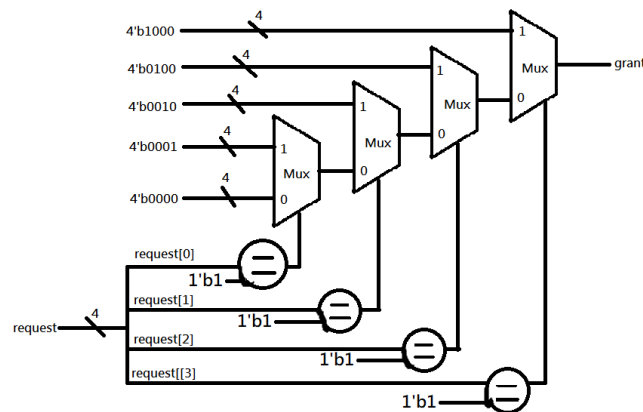
# Lab 1

學號: 109062110

姓名: 祝語辰

## A. Lab Implementation

Lab1\_1:

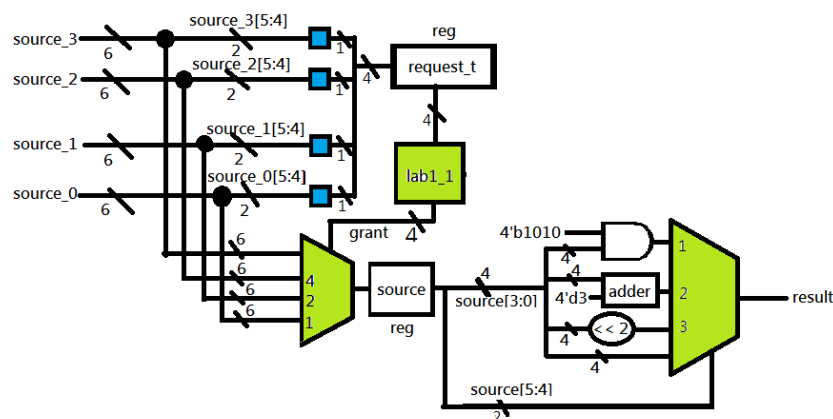


Request 進入後，會直接控制各個 mux，應當通過哪個值。

```
always@(*)begin
    if(request[3] == 1'b1)
        grant = 4'b1000;
    else if(request[2] == 1'b1)
        grant = 4'b0100;
    else if(request[1] == 1'b1)
        grant = 4'b0010;
    else if(request[0] == 1'b1)
        grant = 4'b0001;
    else
        grant = 4'b0000;
end
```

使用 if-else 來實作

Lab1\_2:



Source\_0~3 進入電路，分別進到各自的 block 處理，判斷說是否有 request，以此設定 request\_t。將 4 個 bits 的 request\_t 傳入 lab1\_1 module，決定 grant，並以此來決定 source 的

值，最後進去 source 中 control 的 part，來控制最後的 result。

```
always @(*) begin
    case (source_0[5:4])
        2'b00: request_t[0] = 1'b0;
        default: request_t[0] = 1'b1;
    endcase
end
```

此圖示為 black diagram 中藍色的部分，檢視 control 的部分，來決定是否有 request。

```
always @(*)begin
    case(grant)
        4'b0001: source = source_0;
        4'b0010: source = source_1;
        4'b0100: source = source_2;
        default: source = source_3;
    endcase
end
```

上圖為從 lab1\_1 中的回傳值，來決定誰要通過 mux，進入 source

```
always @(*)begin
    case(source[5:4])
        2'b01: result = source[3:0] & 4'b1010;
        2'b10: result = source[3:0] + 4'd3;
        2'b11: result = source[3:0] << 2;
        default: result = 4'b0;
    endcase
end
```

上圖為另外一個 mux，利用 case 來根據不同 source[5:4]的值，來決定輸出(result)

## B. Questions and Discussions

Please answer the following questions in your report.

- A. In the testbench lab1\_1\_t.v, please explain why we place #DELAY between input assignment and output verification. Hint: Gate delay.

因為訊號在電路裡不一定會是一致的，通過不同的 block 的時間會不同，在處理時，如果沒有考慮這個問題，首先一開始可能會讀不到值，其次是有可能讀到錯誤的值。

- B. In the testbench lab1\_1\_t.v, does the Verilog keyword function synthesizable? If yes, what kind of circuit will it be?

是可以合成的，合成出來的是 combinational circuit.

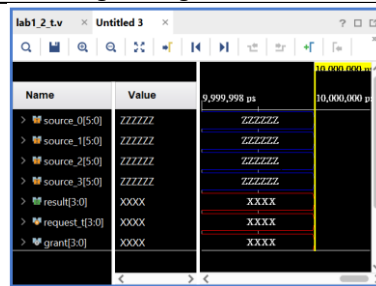
- C. Please explain the difference between if-else and case statements after the synthesis at the gate/cell-level. Hint: priority encoder

If-else 合成出來的好幾個 2 to 1 的 mux。

Case statements 合成出來的是一個 multi-mux，控制多個值，誰要往前。

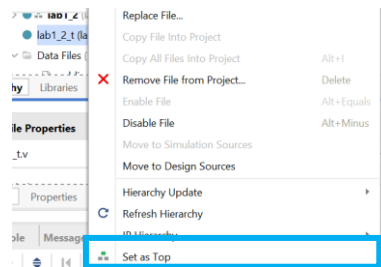
## C. Problem Encountered

在 lab1\_2 首次 simulation 的時候，不知道為什麼跑出來 wave form 中沒有讀到值(顯示 ZZ)。



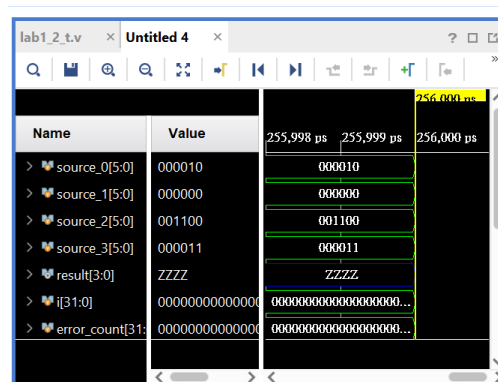
(非當下情境，乃事後重現當時情況)

檢查 testbench 有確定放入 project 中，後來從 error message 上發現是 coding 的時候，沒有 connect module, 因此導致 Vivado 無法判斷哪一個才是 top module，導致設定跑掉。



(非當下情境，乃事後重現當時情況)

選擇 testbench 中的 lab1\_2\_t 作為 top module 之後仍會發現最後的 result 沒有讀到值，



(非當下情境，乃事後重現當時情況)

再檢查 code，將 module connect 上去就沒問題了。

此外在 debug 時，發現在最後產出 result 時出錯，後來上網查才知道寫一個"&"才對(我原本寫"&&"但那是其他意思)

```
always @(*)begin
    case(source[5:4])
        2'b01: result = source[3:0] & 4'b1010;
        2'b10: result = source[3:0] + 4'd3;
        2'b11: result = source[3:0] + 4'd3;
        2'b00: result = source[3:0] + 4'd3;
    endcase
end
```

## D. Suggestions

無