

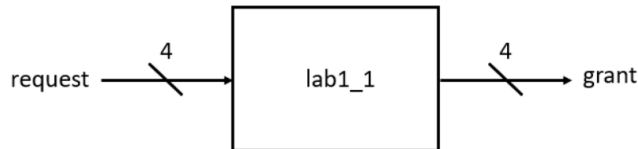
# Lab 1

學號: 110062131

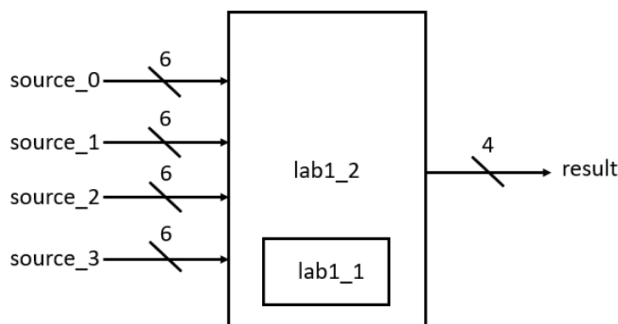
姓名: 馬毓昇

## A. Lab Implementation

Lab1\_1: 用 if-else statement 從最高位判斷到最低位來決定 arbitration。



Lab1\_2: 用每個 source 最高兩位給做 request 的個別 bit 讓 lab1\_1 判斷要處理哪個 source，再以 function 來做出 result。



Lab1\_2 segments:

1. input/output

2. 依 source 的最高兩位做 request

```

15     assign request[3] = source_3[5] | source_3[4];
16     assign request[2] = source_2[5] | source_2[4];
17     assign request[1] = source_1[5] | source_1[4];
18     assign request[0] = source_0[5] | source_0[4];
  
```

3. lab1\_1 做出 grant 代表要處理哪個 source，再跑 case 來處理

```

20     lab1_1 arb(.request(request), .grant(grant));
22     always @* begin
23         result = 4'b0000;
24         #0.5;
25         case(grant)
26             4'b1000: result = process(source_3);
27             4'b0100: result = process(source_2);
28             4'b0010: result = process(source_1);
29             4'b0001: result = process(source_0);
30         endcase
31     end
  
```

4. 因為處理的 code 重複性很高，所以寫成一個 function

```

33     function [3:0] process;
34         input [5:0] source;
35         begin
36             process = 4'b0000;
37             case(source[5:4])
38                 2'b01: process = source[3:0] & 4'b1010;
39                 2'b10: process = source[3:0] + 4'd3;
40                 2'b11: process = source[3:0] << 2;
41             endcase
42         end
43     endfunction
  
```

用到了 Lecture01 怎麼裝和設置 vivado，還有 Lecture02 怎麼用 vivado 跑模擬。主要還是老師的 Verilog Series，Modeling 跟 Blocks 的影片讓這次寫 code 順利許多。

## B. Questions and Discussions

A: 因為 grant 變化會有 Gate delay，要用 #DELAY 等變化結束再做判斷才是對的。

B: 是，MUX，屬於 Combinational Circuit，因為不需記憶。

C: if-else statement 是很多個 2-1MUX 接著，case statement 是一個 N-1MUX。

## C. Problem Encountered

1. 一開始遇到 "No Change in HDL. Linking previously generated obj files to create kernel" 的問題，後來重開 project 就解決了。

2. 有遇到 "concurrent assignment to a non-net 'grant' is not permitted" 的 error，網路上說是因為不能把 reg 型賦值給另一 module 的輸出 reg 型，改成 wire 型就可以了。

```
12     wire [3:0] request;  
13     wire [3:0] grant; //reg [3:0] grant;  
14  
15     assign request[3] = source_3[5] | source_3[4];  
16     assign request[2] = source_2[5] | source_2[4];  
17     assign request[1] = source_1[5] | source_1[4];  
18     assign request[0] = source_0[5] | source_0[4];  
19  
20     labl_1 arb(.request(request), .grant(grant));
```

## D. Suggestions

老師的 verilog 教學影片很棒，我花了一學期還搞不懂的 verilog 僅用短短幾部影片就讓我學懂了，受益良多。

胡瓜到科技業面試。

主管：「你懂製程嗎？」

胡瓜：「不是，我胡瓜。」



換董至成到科技業面試

主管：你懂製程嗎？

董至成：對

主管：你懂什麼？

董至成：志成

主管：什麼製程？

董至成：董

主管：來，下面一位～