

組合語言與微處理機

Lab4

ISS&C2A



組別： 第 3 組

組員： 林鑫彤 黃翊 許睿玲

學號： B013040024 B013040049 B013040051

指導老師： 張雲南 老師

➤ 實驗內容

實作題

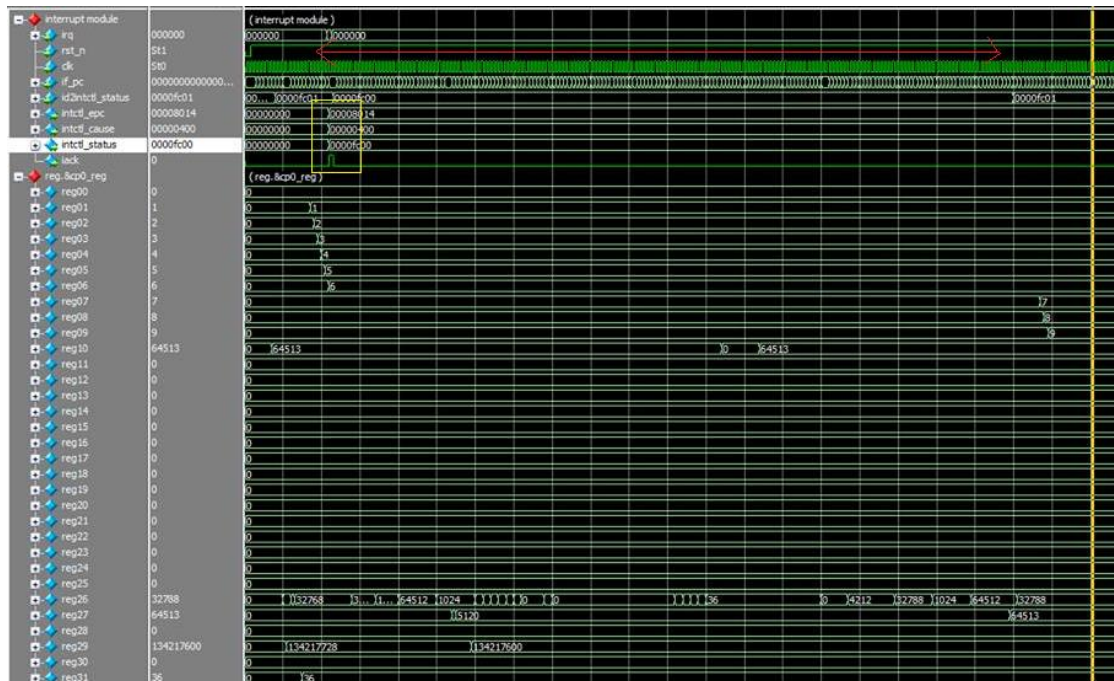
```
always@(negedge clk or negedge rst_n)
begin
    if(~rst_n) //暫存器的初始值設定
    begin
        iack <= 1'b0;
        intctl_epc <= 32'd0;
        intctl_status <= 32'd0;
        intctl_cause <= 32'd0;
    end
    else if ( (irq[5]&id2intctl_status[15])|(irq[4]&id2intctl_status[14])|(irq[3]&id2intctl_status[13])
    |(irq[2]&id2intctl_status[12])|(irq[1]&id2intctl_status[11])|(irq[0]&id2intctl_status[10])) & id2intctl_status[0] ) //判斷硬體中斷是否可被接受
    begin
        intctl_epc <= if_pc - 32'd12; //計算中斷結束後的返回的PC
        intctl_status <= (intctl_status[31:1],1'b1); //disable interrupt, 避免其他中斷要求的干擾
        intctl_cause <= {16'b0,irq,10'b0}; //記錄中斷原因 (=irq[5:0])
        iack <= 1'b1; //發出中斷認可
    end
    else
    begin
        if(if_pc == 32'h00001000) //判斷是否已進入中斷服務程式(ISR)的指令位址
        begin
            iack <= 1'b0; //中斷認可訊號拉回0
        end
        else intctl_epc <= intctl_epc;
    end
end
```

進階題

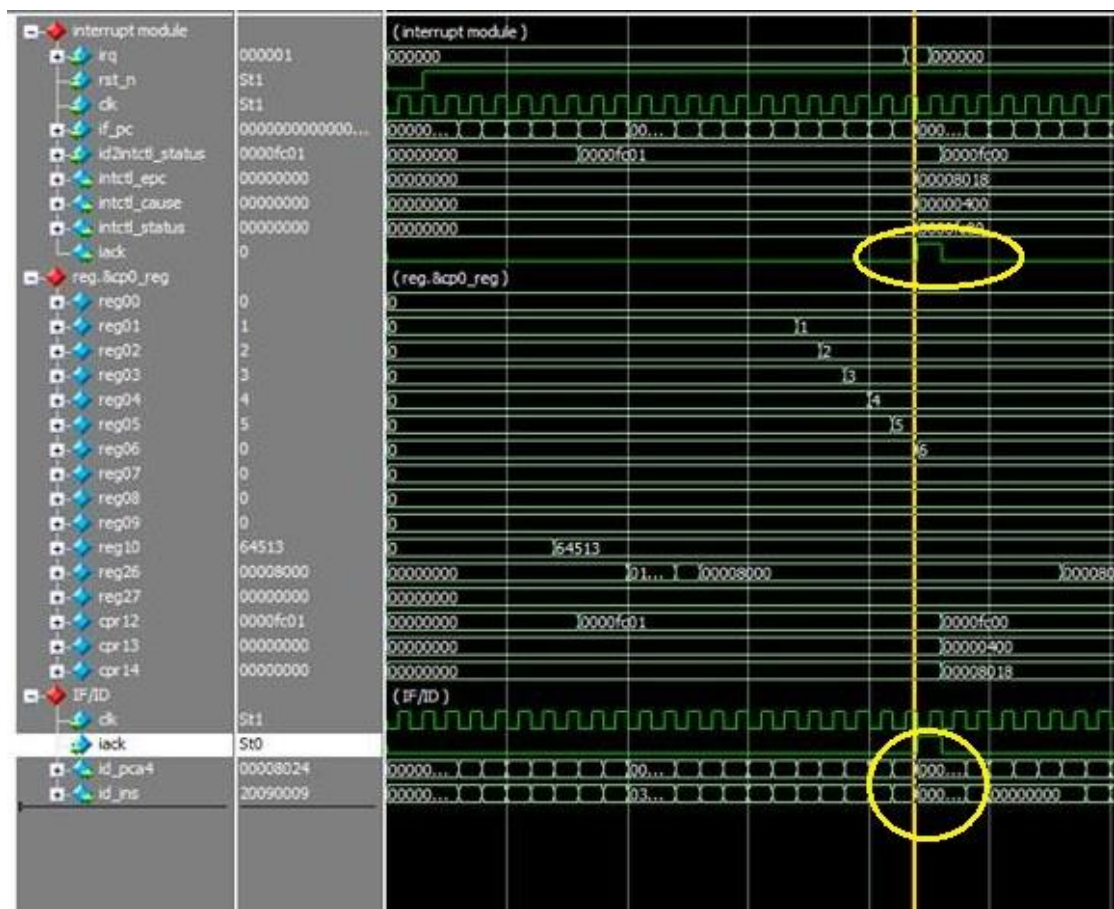
```
always@(negedge clk or negedge rst_n or posedge iack) //當正緣觸發iack or 負緣觸發rst_n or 負緣觸發clk時
begin
    //若rst_n == 0 或 iack == 1時, 設dout = 0(flush動作)
    if(~rst_n || iack)
    begin
        dout <= 32'b0;
    end
    else
    begin
        if(wren) begin dout <= din; $monitor("%0dms : \n$monitor: wren=%b ", $stime, wren); end
        else dout <= dout;
    end
end
```

➤ 驗證的方式與模擬的結果

實作題



進階題



➤ 心得與討論

從這次實驗學習到一些寫 verilog 的一些小技巧，像是怎麼樣把 register 橋接起來，如在記錄中斷原因的時候，我們其實只要中間的 `irq[5:0]` 就好，但是因為 register 是 32 位元所以我們必須把其他地方填滿 0，一開始卡這邊卡很久是因為不知道怎麼把其他地方都填 0，後來經由助教的教學，讓我們知道可以用這樣 `{16'b0,irq,10'b0};` 的方式來完成，其他部分照著投影片教學，還有 code 裡面所打好的註解就可以一步一步完成。進階題部分是要實作 flush 功能，一開始有想一下，再上網 google 查詢 flush 的功能之後，把題目中的 ?? 設為 iack 就成功了，還有因為 modlesim 出了點問題，感謝學長的幫忙，才能順利完成這份作業。