



Lab06 code review

Pattern

- 檢查design的錯誤
 - **Spec1** : reset後所有的output signal要歸零
 - **Spec2** : 計算完吐完值後4 cycles , out_valid要歸零 (out_valid只維持4 cycles)
 - **Spec3** : outvalid 不能跟 invalid重疊
 - **Spec4** : 100cycle內要計算完成(out_valid為high之前)
 - **Spec5** : function要對(前一頁的公式會有錯要檢查出來)
 - **Spec6** : 當out_valid是0時, out_result要歸零

產生clock

- Clock 要記得給初始值

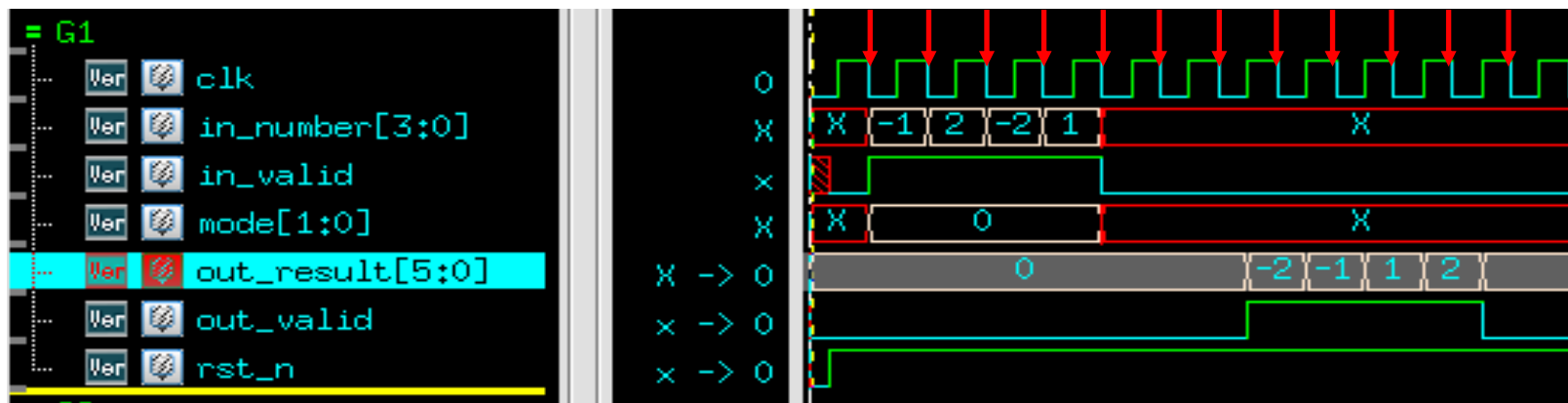
```
integer CYCLE = 10;  
always #(CYCLE/2.0) clk = ~clk;  
initial clk = 0;
```

如何產生signed的測資

- urandom_range要怎麼給signed的數
- test = \$urandom_range(15,0);
- 他會回傳0000~1111
- 所以只要你前面的變數宣告為4bits signed，他就會自動識別為-8~7

Pattern 給值與檢查值的時間

- Pattern 給值與檢查的時機都是在 `negedge clk`，目的是為了讓 design 在 `posedge clk` 取值時機與給值時間相隔半個 cycle，檢查時機也會等到 output 穩定半個 cycle 後才會檢查 output 值。
- 使用 `@(negedge clk);` 等到下一個 `negedge clk` 才繼續執行後面的 code，否則會直接執行下去



Sample code

- Initial 裡面的code是依序執行的，因此task的順序很重要
- Spec3(invalid和outvalid不能重疊)在input_task裡面檢查，邊給input邊檢查outvalid是否為1

```
initial begin
    reset_task;
    in_valid = 0;
    in_number = 4'dx;
    mode = 2'dx;
    total_latency = 0;
    for (i=0;i<=PATNUM;i=i+1)begin
        input_task;
        gen_ans;
        wait_outvalid;
        check_ans;
        outvalid_rst;
        @(negedge clk);
    end
    #(1000)
    YOU_PASS_task;
    $finish;
end
```

Sample code

- Spec5 check answer只能檢查4個cycle
- 第5個cycle如果outvalid仍為1 算 spec2(outvalid沒有歸零)

```
task outvalid_rst;begin
    if( out_valid != 0 )begin
        //fail;
        $display ("-----");
        $display ("
        $display ("
        $display ("-----");
        #(100);
        $finish ;
    end
```

```
task check_ans ; begin
    for(cnt = 0;cnt<4;cnt=cnt+1 ) begin
        if(d_mode == 0) begin
            golden = d_number[cnt];
        end
        else if(d_mode == 1) case (cnt) ...
        else if(d_mode == 2) begin...
        end
        else if(d_mode == 3)begin...
        end
        if(out_result!= golden)begin
            //fail;
            $display ("-----");
            $display ("
            $display ("
            $display ("
            $display ("-----");
            #(100);
            $finish;
        end
    end
end
@ (negedge clk);
```