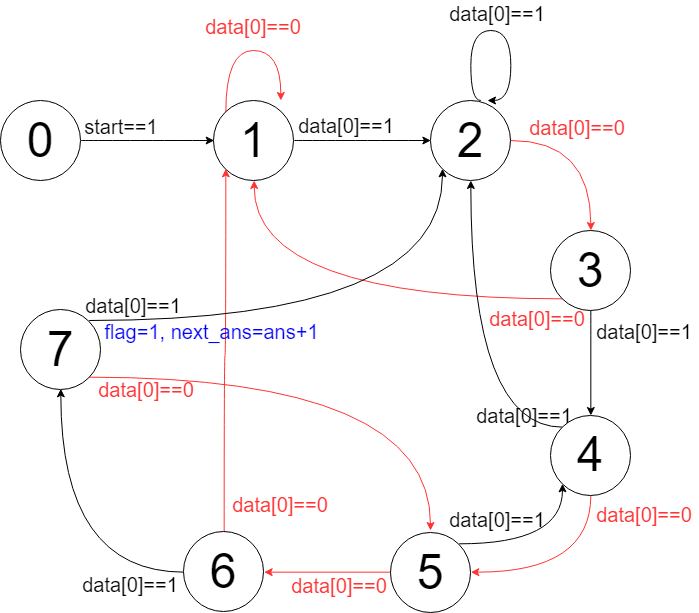
**Lab3: PTM**

# State Transition Graph



## Reason

0-1：0是初始值。Start跳到1之後，state從1開始。

1-6：每當分別符合101001中相應的數字，就往上個state。

1 2 3 4 5 6

如果不符合，就按以下方式判斷：

**state 不符合的結果 1010011 match next state**

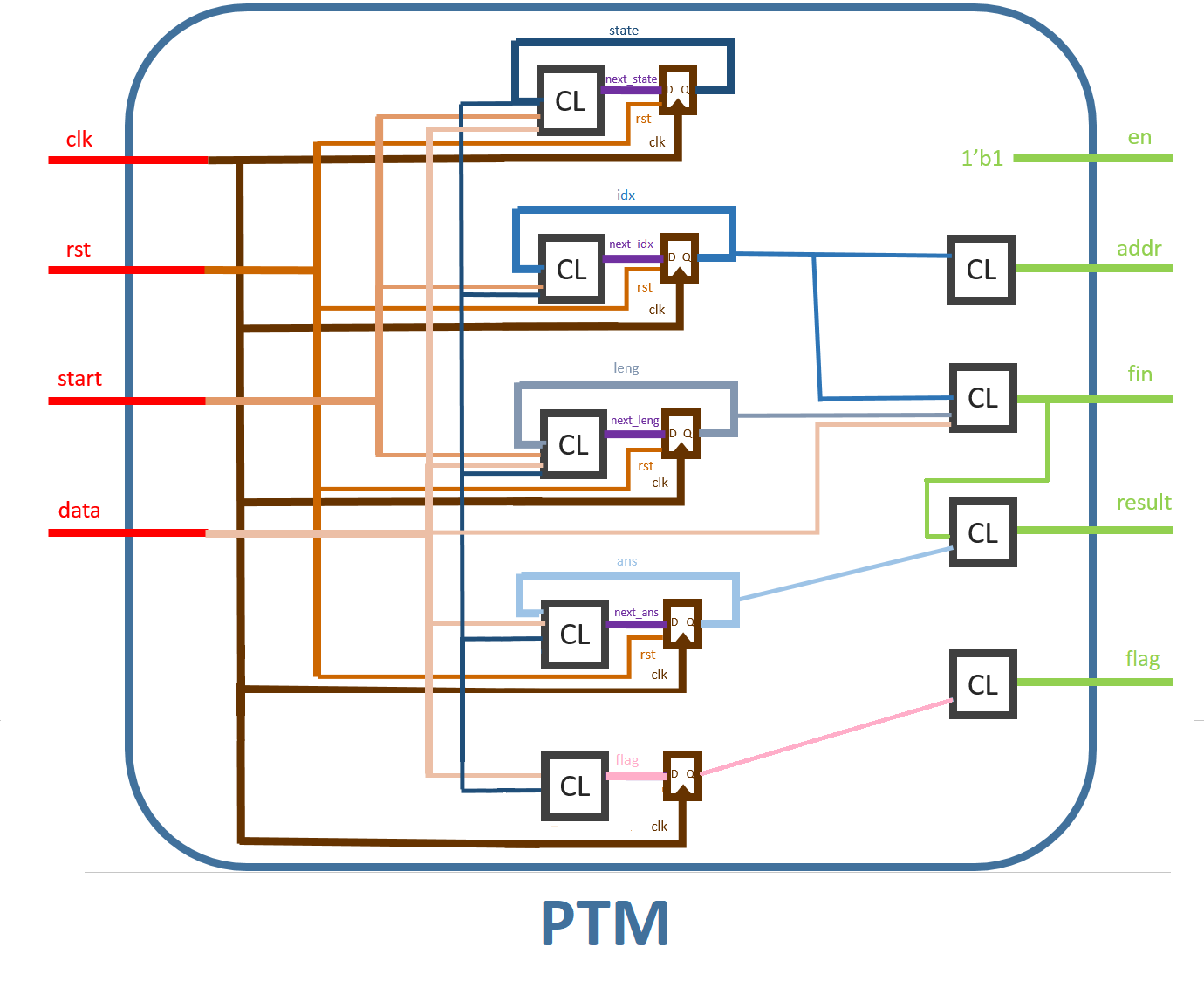
1. 0 1010011 0 0+1=1
2. 11 1010011 1 1+1=2
3. 100 1010011 0 0+1=1
4. 1011 1010011 1 1+1=2
5. 10101 1010011 3 3+1=4
6. 101000 1010011 0 0+1=1

7：如果是0 -> 回到 (100)+ 的循環。

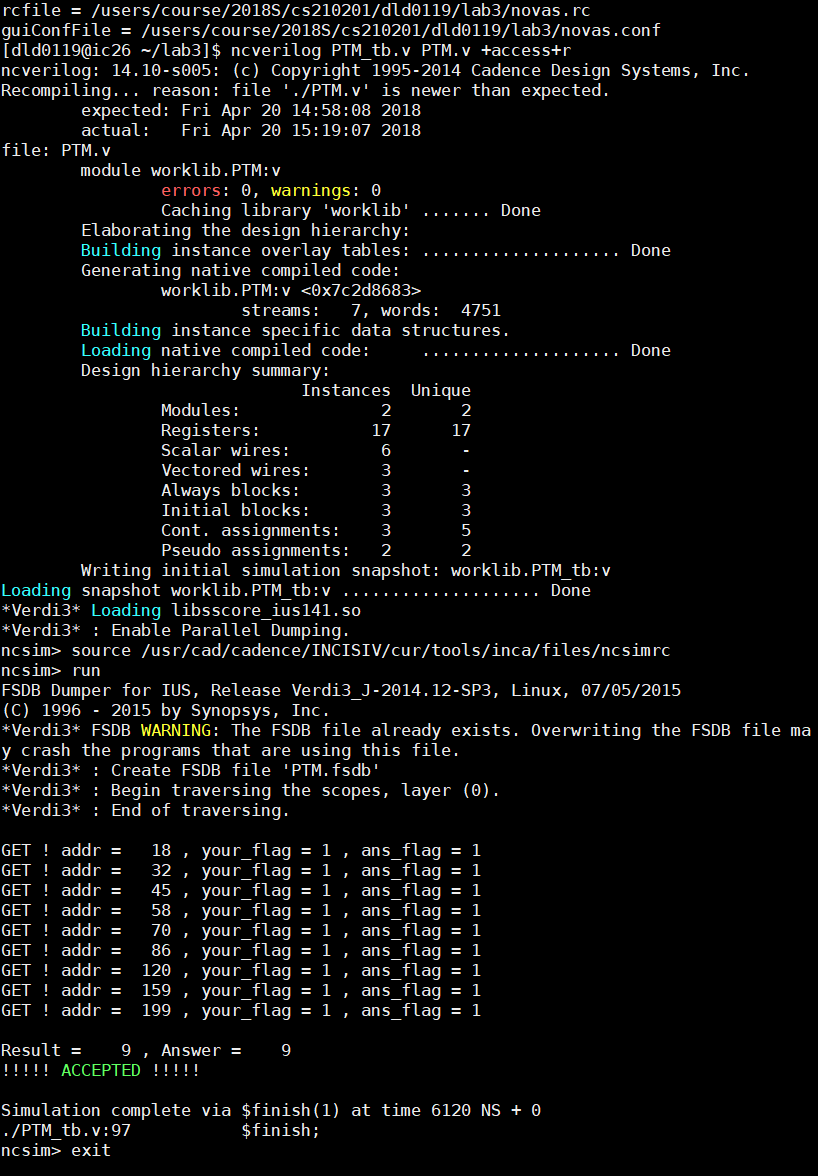
5 6 7

如果是1表示偵測到符合的字串，flag設為1，ans也 +1。同時1可以作為下個字串的開頭，回到state 2。

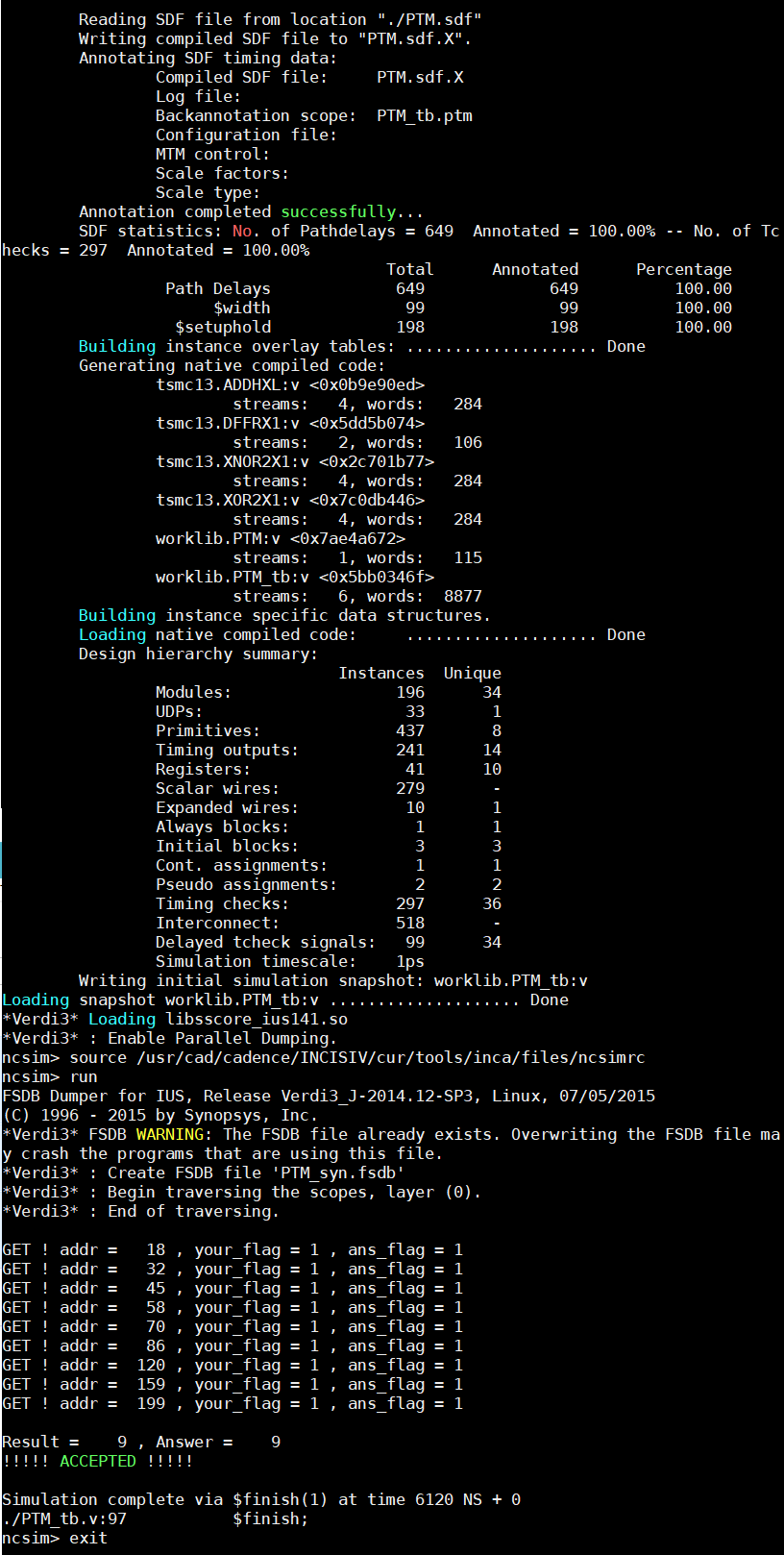
# Block Diagram



# ncverilog simulation (sim)



# ncverilog simulation (syn)



# Discussion

## Initial state

剛開始在initial state直接判斷第一個字有沒有符合字串，結果在讀leng的時候遇到問題。

3'd0:begin

next\_state = (start == 1'b1 && data[0] == 1'b1)? state + 3'd1: 3'd0;

next\_idx = (start == 1'b1)? idx +10'd1: 10'd1023;

next\_leng = (start == 1'b1)? leng: data;

end

後來把initial state跟判斷第一個字分開，就不會有這個問題。

3'd0:begin

next\_state = (start == 1'b1)? 3'd1 : 3'd0;

next\_idx = (start == 1'b1)? 10'd0 : 10'd1023;

next\_leng = (start == 1'b1)? data : leng;

end

3'd1:begin

next\_state = (data[0] == 1'b1)? state + 3'd1 : state;

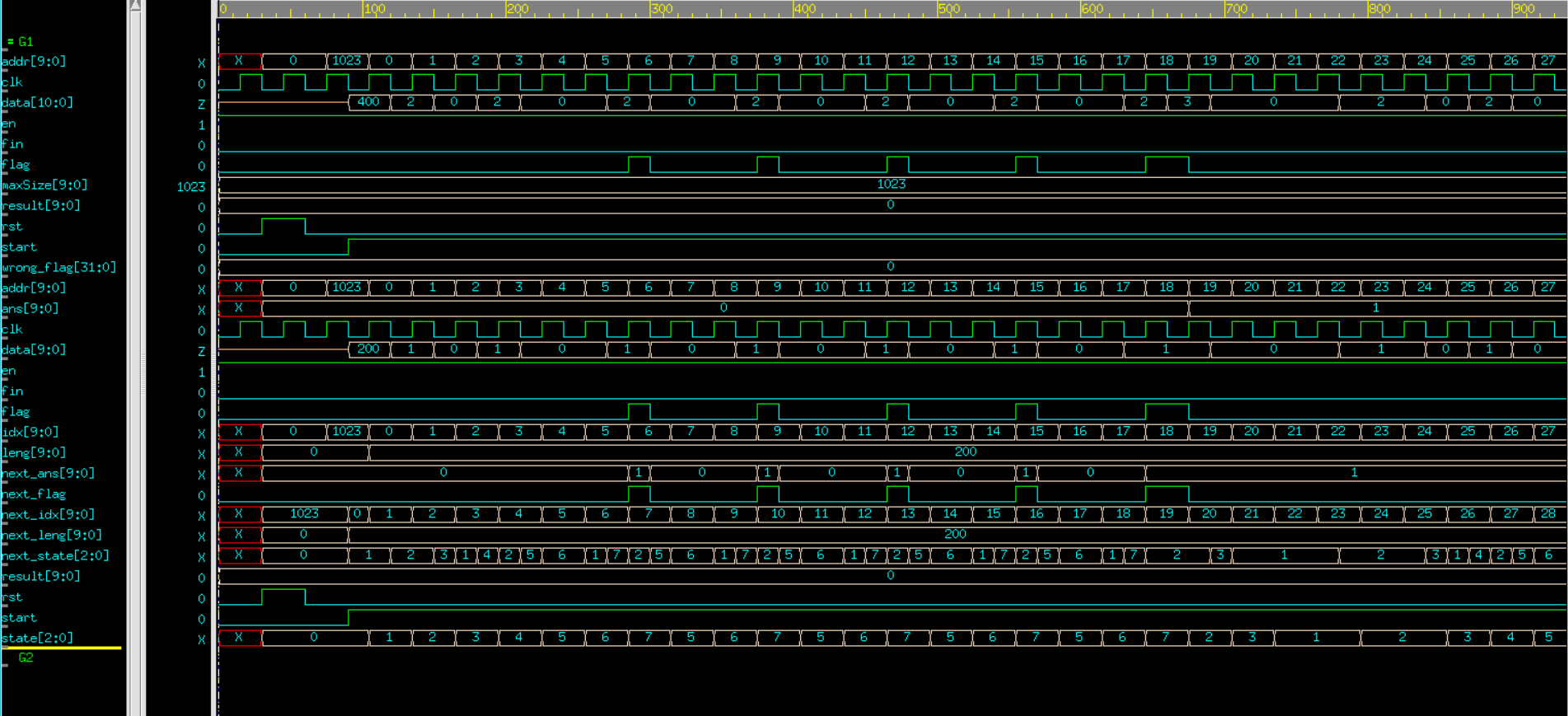
next\_idx = (idx == 10'd1023)? 10'd0 : idx + 10'd1;

next\_cont = (data[0] == 1'b1)? cont : 1'b0;

end

## Always block中變數亂跳的問題

在檢查nWave的時候發現以下情況：



在clk==0的時候變數（flag、next\_ans、flag、next\_idx、next\_state）會在if else的結果之間跳，然而由於DFF的原因，輸出的ans、idx、state卻是穩定的。

如果加上clk == 1'b0的條件就不會這樣了。

flag = (data[0] == 1'b1 && clk == 1'b0)? 1'b1 : 1'b0;

