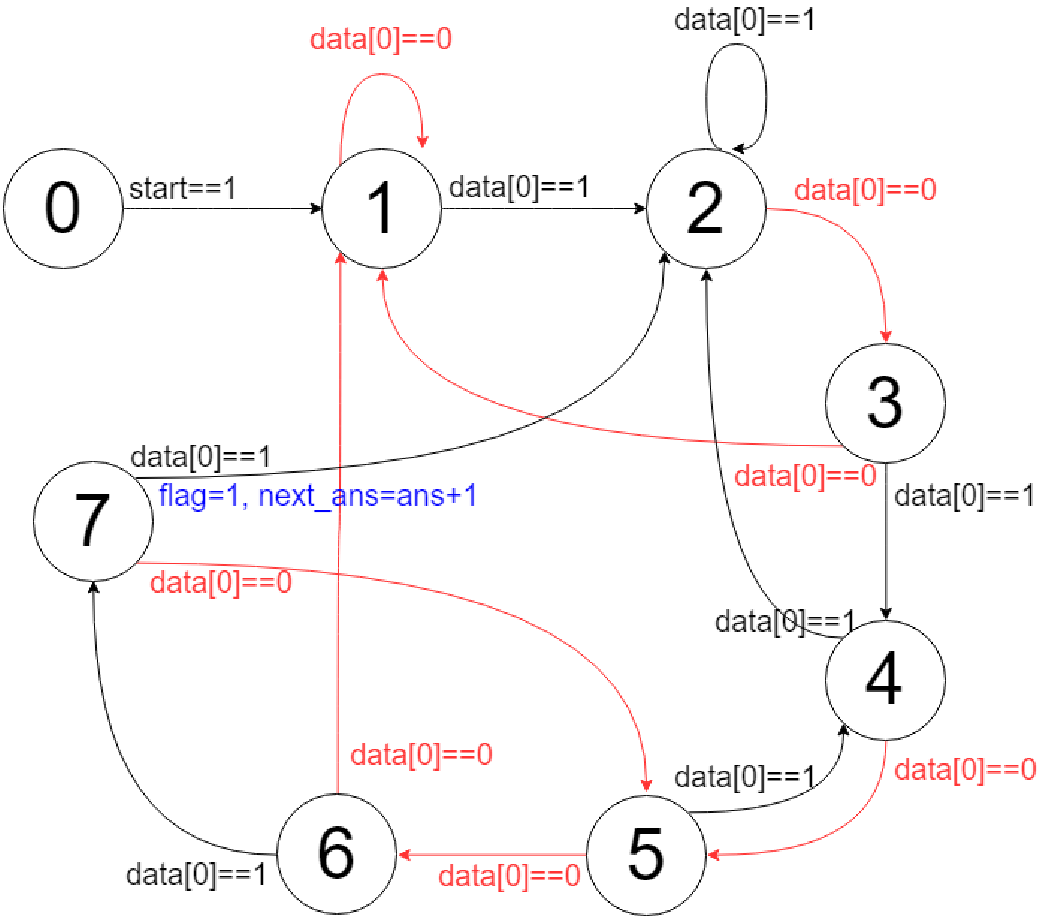
# Lab3: PTM

## 1) 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

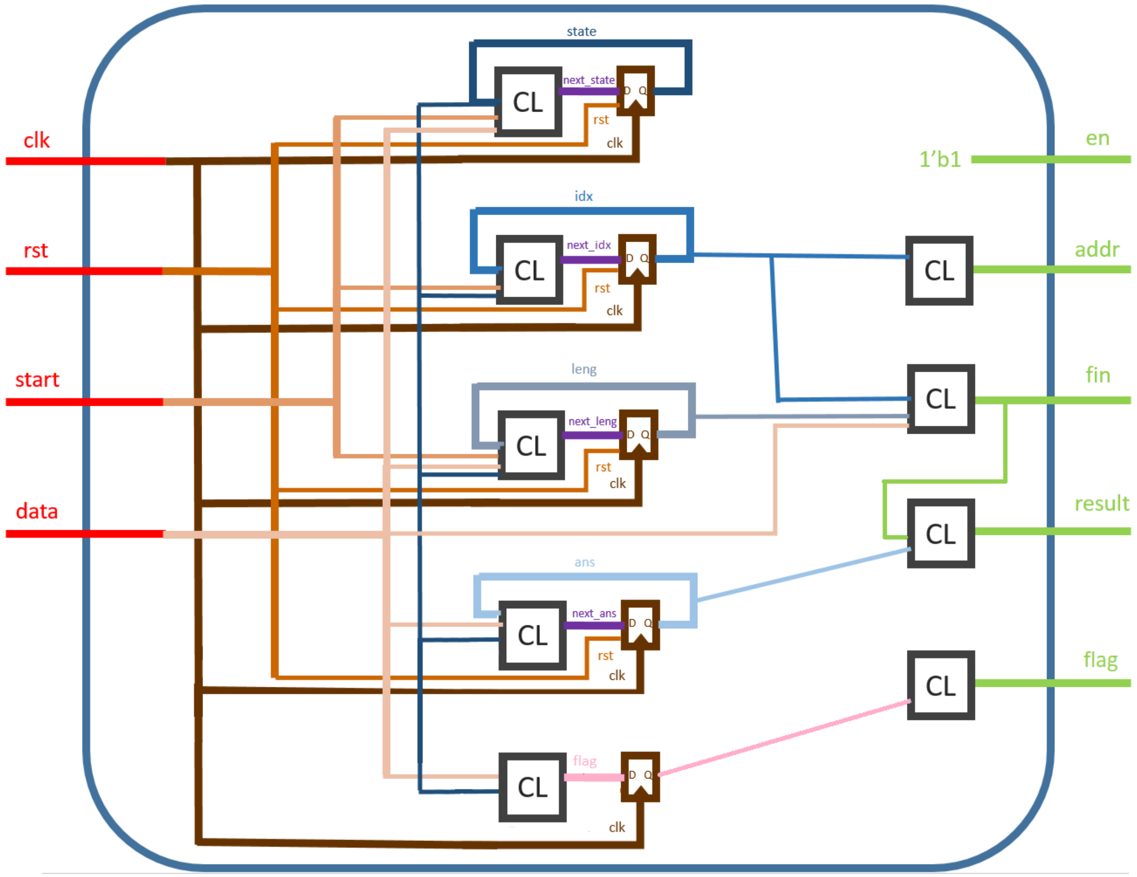
如果不符合，就按以下方式判斷：

| 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+4=4 |
| 6 | 101000 | 1010011 | 0 | 0+1=1 |

7：如果是 0 -> 回到 (100)+ 的循環。
5 6 7

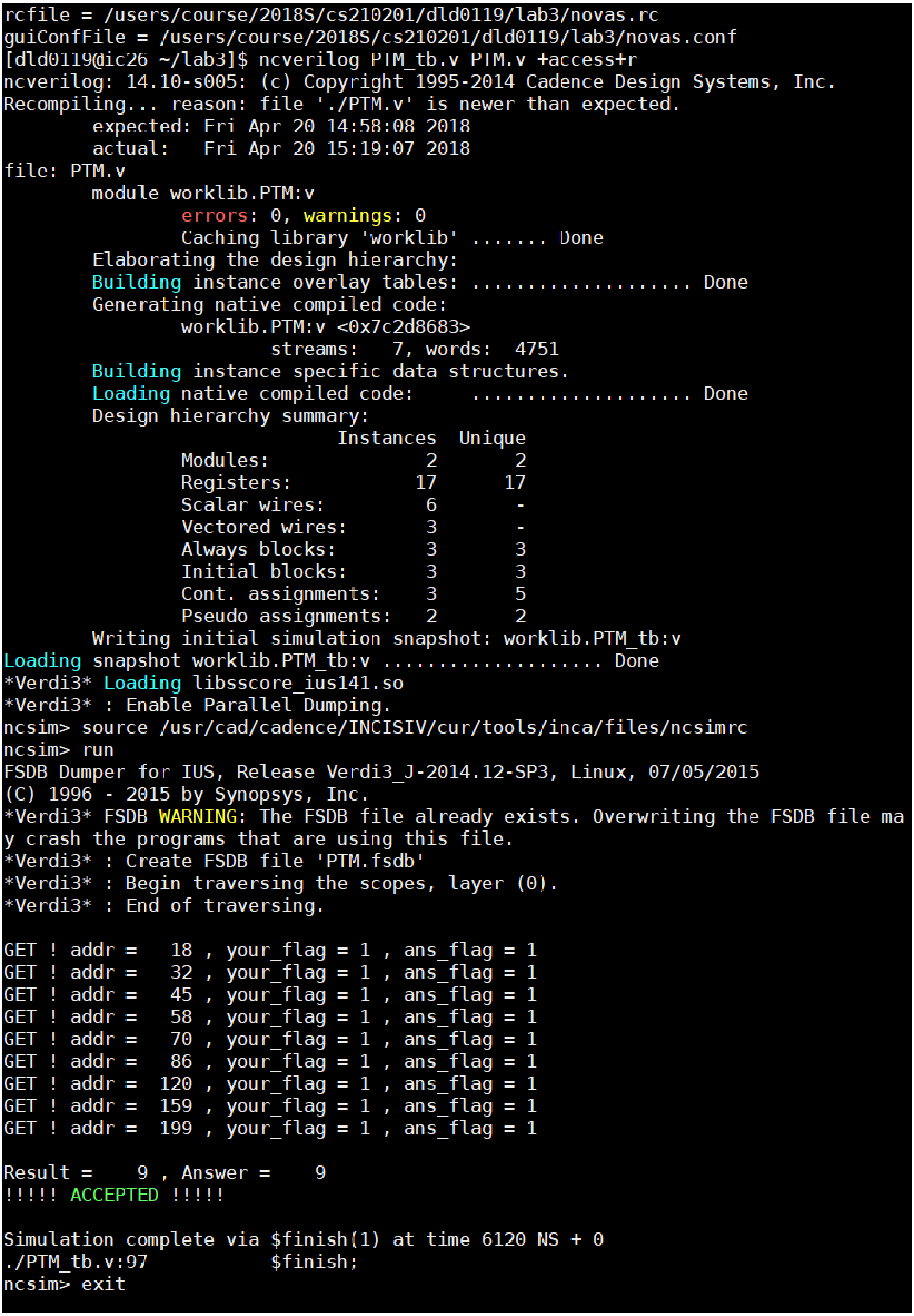
如果是 1 表示偵測到符合的字串，flag 設為 1，ans 也 +1。同時 1 可以作為下個字串的開頭，回到 state 2。

## 2) Block Diagram



**PTM**

## 3) ncverilog simulation (sim)



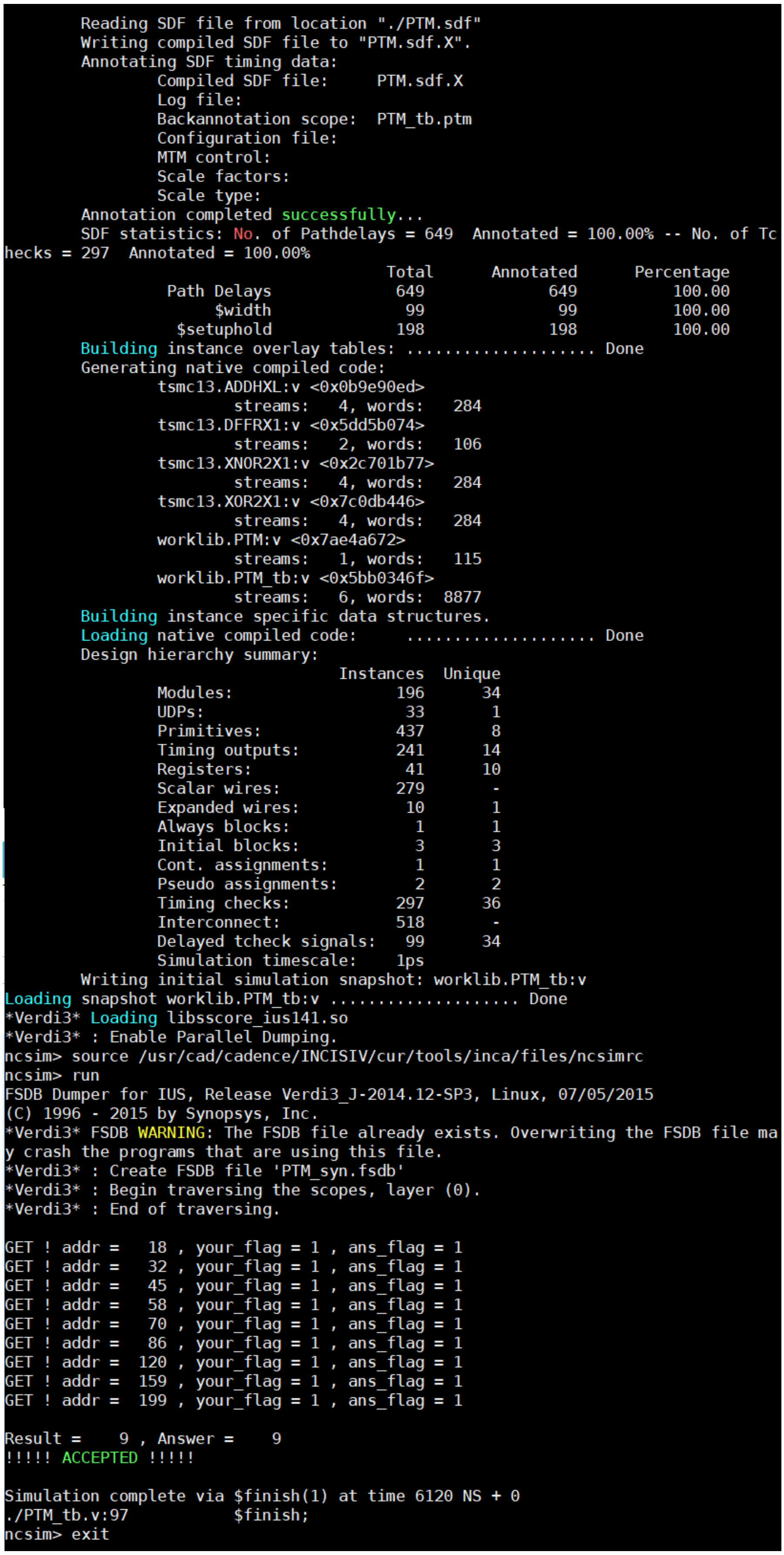
```
rcfile = /users/course/2018S/cs210201/dld0119/lab3/novas.rc
guiConfFile = /users/course/2018S/cs210201/dld0119/lab3/novas.conf
[dld0119@ic26 ~/lab3]$ ncverilog PTM_tb.v PTM.v +access+r
ncverilog: 14.10-s005: (c) Copyright 1995-2014 Cadence Design Systems, Inc.
Recompiling... reason: file './PTM.v' is newer than expected.
        expected: Fri Apr 20 14:58:08 2018
        actual:   Fri Apr 20 15:19:07 2018
file: PTM.v
        module worklib.PTM:v
                errors: 0, warnings: 0
                Caching library 'worklib' ....... Done
        Elaborating the design hierarchy:
        Building instance overlay tables: .................... Done
        Generating native compiled code:
                worklib.PTM:v <0x7c2d8683>
                        streams:   7, words:  4751
        Building instance specific data structures.
        Loading native compiled code:     .................... Done
        Design hierarchy summary:
                             Instances  Unique
                Modules:             2       2
                Registers:          17      17
                Scalar wires:        6       -
                Vectored wires:      3       -
                Always blocks:       3       3
                Initial blocks:      3       3
                Cont. assignments:   3       5
                Pseudo assignments:  2       2
        Writing initial simulation snapshot: worklib.PTM_tb:v
Loading snapshot worklib.PTM_tb:v .................... Done
*Verdi3* Loading libsscore_ius141.so
*Verdi3* : Enable Parallel Dumping.
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi3_J-2014.12-SP3, Linux, 07/05/2015
(C) 1996 - 2015 by Synopsys, Inc.
*Verdi3* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file ma
y crash the programs that are using this file.
*Verdi3* : Create FSDB file 'PTM.fsdb'
*Verdi3* : Begin traversing the scopes, layer (0).
*Verdi3* : End of traversing.

GET ! addr =   18 , your_flag = 1 , ans_flag = 1
GET ! addr =   32 , your_flag = 1 , ans_flag = 1
GET ! addr =   45 , your_flag = 1 , ans_flag = 1
GET ! addr =   58 , your_flag = 1 , ans_flag = 1
GET ! addr =   70 , your_flag = 1 , ans_flag = 1
GET ! addr =   86 , your_flag = 1 , ans_flag = 1
GET ! addr =  120 , your_flag = 1 , ans_flag = 1
GET ! addr =  159 , your_flag = 1 , ans_flag = 1
GET ! addr =  199 , your_flag = 1 , ans_flag = 1

Result =     9 , Answer =     9
!!!!! ACCEPTED !!!!!

Simulation complete via $finish(1) at time 6120 NS + 0
./PTM_tb.v:97           $finish;
ncsim> exit
```

# 4) ncverilog simulation (syn)



```
        Reading SDF file from location "./PTM.sdf"
        Writing compiled SDF file to "PTM.sdf.X".
        Annotating SDF timing data:
                Compiled SDF file:     PTM.sdf.X
                Log file:
                Backannotation scope:  PTM_tb.ptm
                Configuration file:
                MTM control:
                Scale factors:
                Scale type:
        Annotation completed successfully...
        SDF statistics: No. of Pathdelays = 649  Annotated = 100.00% -- No. of Tc
hecks = 297  Annotated = 100.00%
                                        Total      Annotated      Percentage
                 Path Delays             649             649          100.00
                      $width              99              99          100.00
                  $setuphold             198             198          100.00
        Building instance overlay tables: .................... Done
        Generating native compiled code:
                tsmc13.ADDHXL:v <0x0b9e90ed>
                        streams:   4, words:   284
                tsmc13.DFFRX1:v <0x5dd5b074>
                        streams:   2, words:   106
                tsmc13.XNOR2X1:v <0x2c701b77>
                        streams:   4, words:   284
                tsmc13.XOR2X1:v <0x7c0db446>
                        streams:   4, words:   284
                worklib.PTM:v <0x7ae4a672>
                        streams:   1, words:   115
                worklib.PTM_tb:v <0x5bb0346f>
                        streams:   6, words:  8877
        Building instance specific data structures.
        Loading native compiled code:     .................... Done
        Design hierarchy summary:
                                   Instances  Unique
                Modules:                 196      34
                UDPs:                     33       1
                Primitives:              437       8
                Timing outputs:          241      14
                Registers:                41      10
                Scalar wires:            279       -
                Expanded wires:           10       1
                Always blocks:             1       1
                Initial blocks:            3       3
                Cont. assignments:         1       1
                Pseudo assignments:        2       2
                Timing checks:           297      36
                Interconnect:            518       -
                Delayed tcheck signals:   99      34
                Simulation timescale:    1ps
        Writing initial simulation snapshot: worklib.PTM_tb:v
Loading snapshot worklib.PTM_tb:v .................... Done
*Verdi3* Loading libsscore_ius141.so
*Verdi3* : Enable Parallel Dumping.
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi3_J-2014.12-SP3, Linux, 07/05/2015
(C) 1996 - 2015 by Synopsys, Inc.
*Verdi3* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file ma
y crash the programs that are using this file.
*Verdi3* : Create FSDB file 'PTM_syn.fsdb'
*Verdi3* : Begin traversing the scopes, layer (0).
*Verdi3* : End of traversing.

GET ! addr =   18 , your_flag = 1 , ans_flag = 1
GET ! addr =   32 , your_flag = 1 , ans_flag = 1
GET ! addr =   45 , your_flag = 1 , ans_flag = 1
GET ! addr =   58 , your_flag = 1 , ans_flag = 1
GET ! addr =   70 , your_flag = 1 , ans_flag = 1
GET ! addr =   86 , your_flag = 1 , ans_flag = 1
GET ! addr =  120 , your_flag = 1 , ans_flag = 1
GET ! addr =  159 , your_flag = 1 , ans_flag = 1
GET ! addr =  199 , your_flag = 1 , ans_flag = 1

Result =    9 , Answer =    9
!!!!! ACCEPTED !!!!!

Simulation complete via $finish(1) at time 6120 NS + 0
./PTM_tb.v:97            $finish;
ncsim> exit
```

## 5) 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。

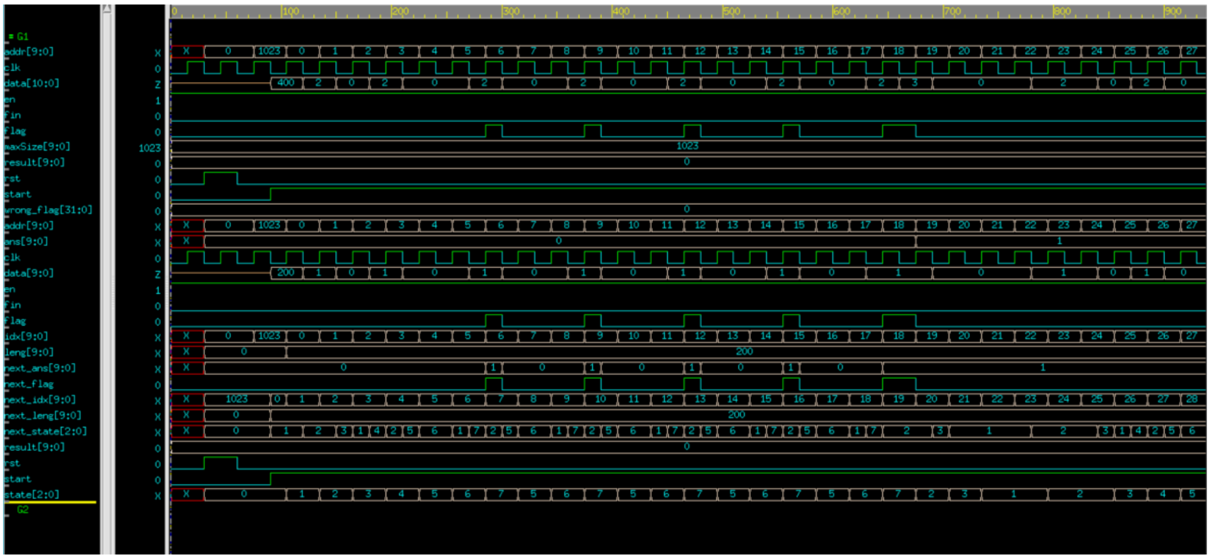
```
3'd0:begin
	next_state = (start == 1'b1 && data[0] == 1'b1)? state + 3'd1: 3'd0;
	next_idx = (start == 1'b1)? idx +10'd1: 10'd1023;
	next_leng = (start == 1'b1)? leng: data;
end
```

後來把 inittial 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

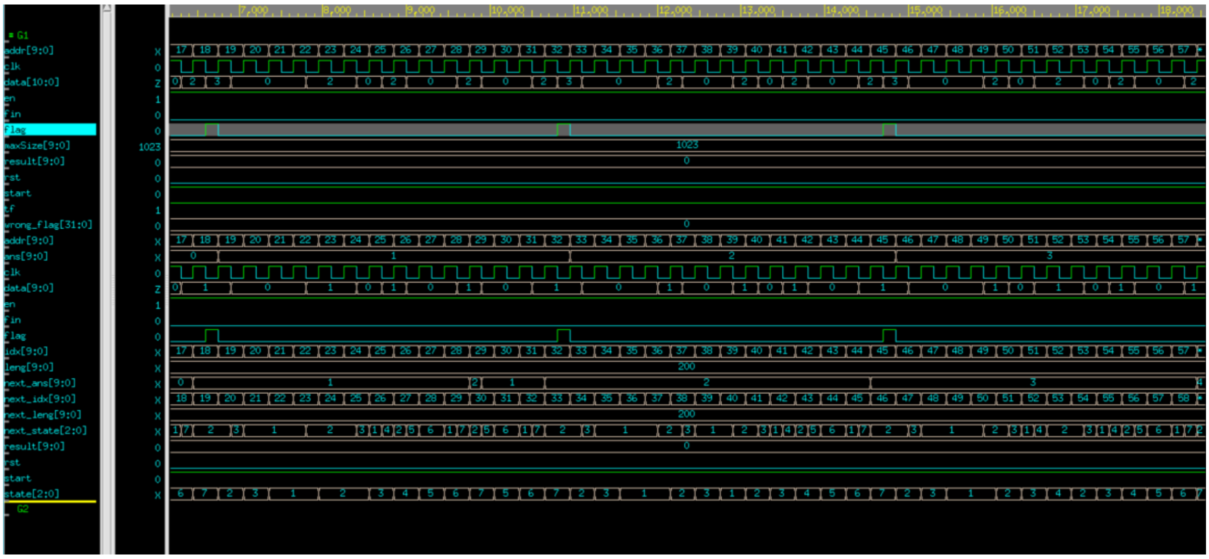
在檢查 nWave 的時候發現以下情況：



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;
```


G1
addr[9:0]
clk
data[10:0]
en
fin
flag
maxSize[9:0]
result[9:0]
rst
start
tf
wrong_flag[31:0]
addr[9:0]
ans[9:0]
clk
data[9:0]
en
fin
flag
idx[9:0]
leng[9:0]
next_ans[9:0]
next_idx[9:0]
next_leng[9:0]
next_state[2:0]
result[9:0]
rst
start
state[2:0]
G2