Lab3 Pattern Matching

105031212 吳紹齊

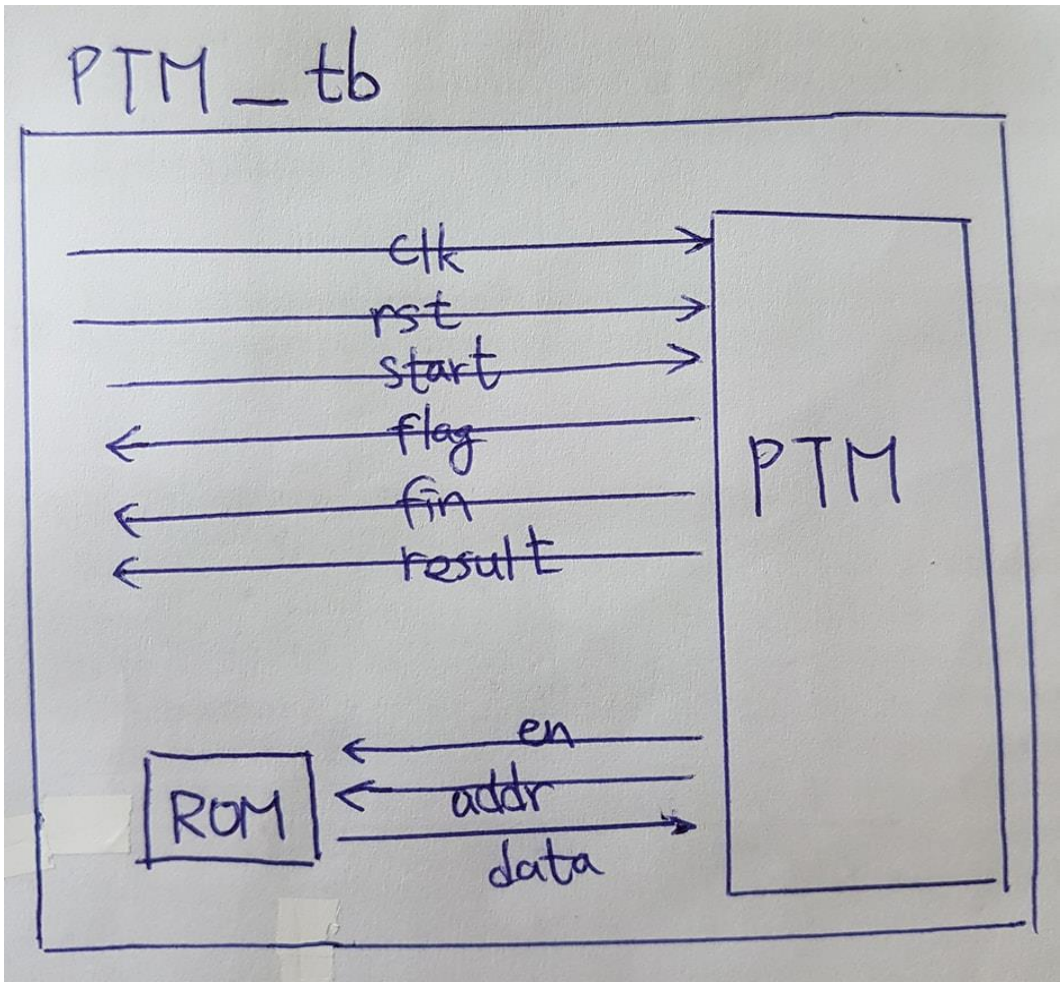
一、目標

請找符合以下特徵的字串

$$10(100)^{+}11$$ (+ 表示出現次數 >= 1)

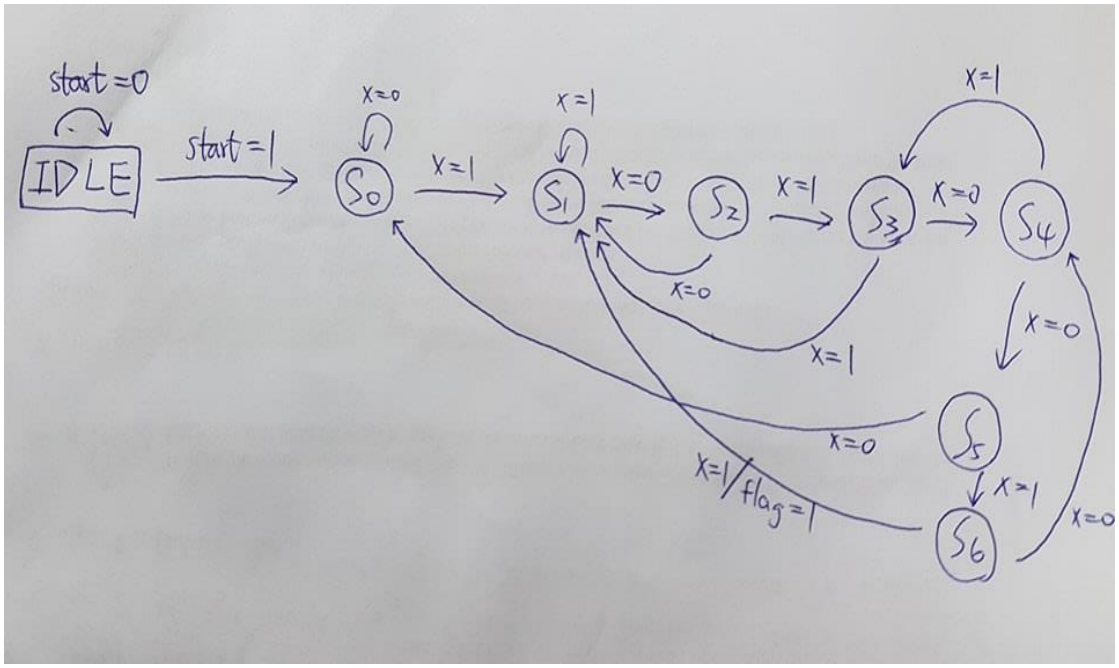
二、設計

這次的作業需要從 ROM 叫資料出來。Pattern Matching 電路(PTM)、ROM、Testbench(PTM_tb)的關係如下圖所示：



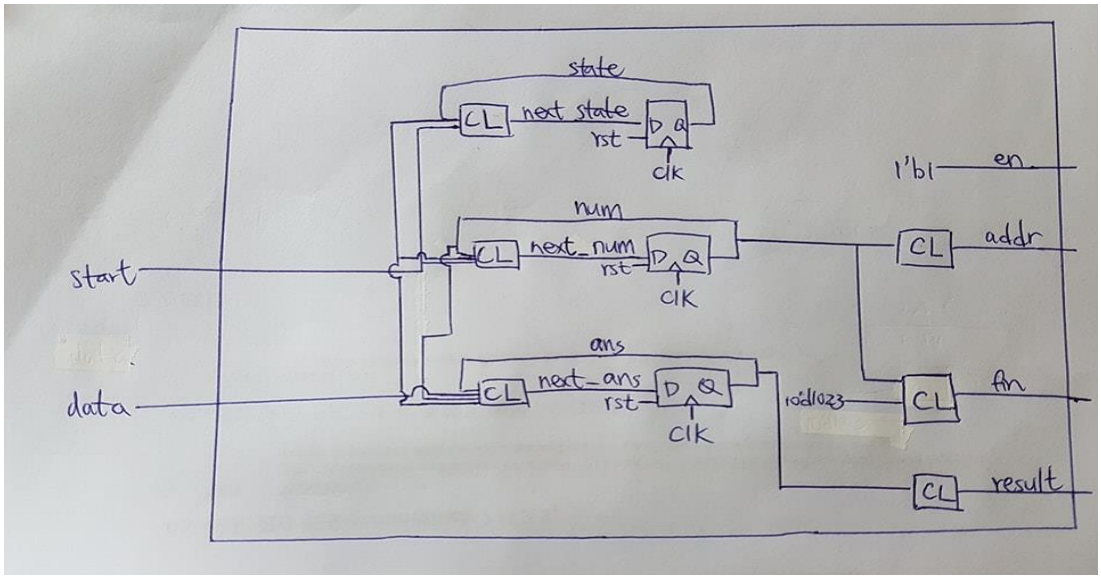
我設計了一個 Mealy Machine 來達成題目的要求，以下是這個 PTM 的

State Transition Graph：

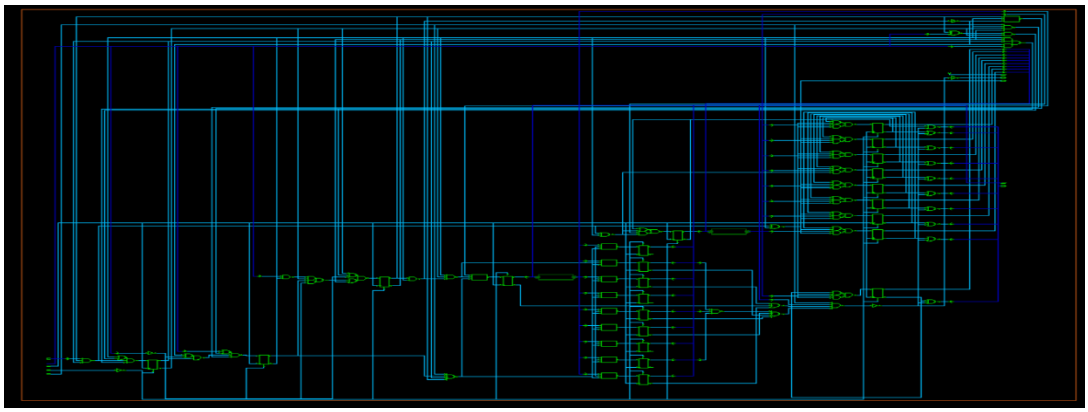


1. 若 Start =1，離開 IDLE 狀態，開始從 address[0] 讀 ROM 的 data。
   (透過 Counter (num) 讓我能夠從 address[0]往下讀到 address[1022])
2. 判斷 data[0]的值，訂出分別為 1 和 0 時，下一個 state 該如何變化。
3. 抵達 S6 時，若輸入 data[0]為 1，完整 pattern 產生，flag 升起(=1)。

Block Diagram：

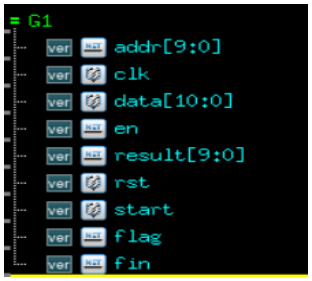


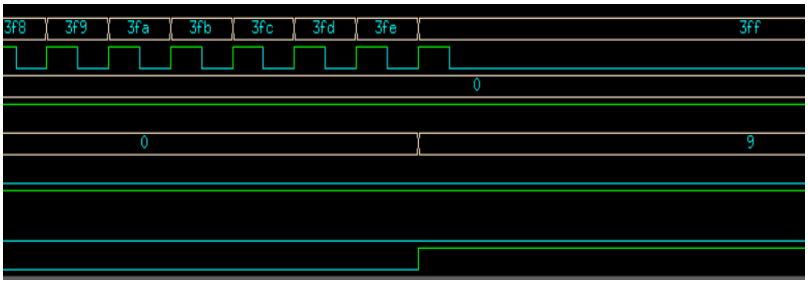
dv 模擬電路：



nWave 波形圖：

波形圖能夠輔助了解電路是否如預期運作，下面的例子為 finish 與 result 的互動關係。可以發現當 fin 拉起來時(=1)，電路輸出 result(=9)。





ncverilog 模擬結果：

(make sim)

```
[dld0026@ic26 ~/Lab_3]$ make sim
ncverilog header.v PTM.v PTM_tb.v +access+r
ncverilog: 14.10-s005: (c) Copyright 1995-2014 Cadence Design Systems, Inc.
file: header.v
file: PTM.v
        module worklib.PTM:v
                errors: 0, warnings: 0
file: PTM_tb.v
        module worklib.PTM_tb: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 <0x60a9adf1>
                        streams:   9, words:  4310
                worklib.PTM_tb:v <0x5f32e0aa>
                        streams:   6, words:  8378
        Building instance specific data structures.
        Loading native compiled code:     ..................... Done
        Design hierarchy summary:
                                Instances  Unique
                Modules:                2       2
                Registers:             15      15
                Scalar wires:           6       -
                Vectored wires:         3       -
                Always blocks:          5       5
                Initial blocks:         3       3
                Cont. assignments:      3       4
                Pseudo assignments:     2       2
                Simulation timescale:  100ps
        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* : 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 30810 NS + 0
./PTM_tb.v:97           $finish;
ncsim> exit
```

(make syn)

```
[dld0026@ic26 ~/Lab_3]$ make syn
ncverilog header.v PTM_syn.v PTM_tb.v -v /theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v +define+SDF +access+r
ncverilog: 14.10-s005: (c) Copyright 1995-2014 Cadence Design Systems, Inc.
file: PTM_syn.v
        module worklib.PTM_DW01_inc_0:v
                errors: 0, warnings: 0
        module worklib.PTM_DW01_inc_1:v
                errors: 0, warnings: 0
        module worklib.PTM:v
                errors: 0, warnings: 0
file: PTM_tb.v
        module worklib.PTM_tb:v
                errors: 0, warnings: 0
file: /theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v
        module tsmc13.AOI22X1:v
                errors: 0, warnings: 0
        module tsmc13.AOI32X1:v
                errors: 0, warnings: 0
        module tsmc13.AOI2BB1X1:v
                errors: 0, warnings: 0
        module tsmc13.NAND2X1:v
                errors: 0, warnings: 0
        module tsmc13.NAND3X1:v
                errors: 0, warnings: 0
        module tsmc13.NAND4X1:v
                errors: 0, warnings: 0
        module tsmc13.NOR2X1:v
                errors: 0, warnings: 0
        module tsmc13.NOR3X1:v
                errors: 0, warnings: 0
        module tsmc13.NOR4X1:v
                errors: 0, warnings: 0
        module tsmc13.OR4X1:v
                errors: 0, warnings: 0
        module tsmc13.OAI21XL:v
                errors: 0, warnings: 0
        module tsmc13.OAI211X1:v
                errors: 0, warnings: 0
        module tsmc13.OAI32X1:v
                errors: 0, warnings: 0
        module tsmc13.OAI2BB1X1:v
                errors: 0, warnings: 0
        module tsmc13.OAI2BB2XL:v
                errors: 0, warnings: 0
        module tsmc13.XOR2X1:v
                errors: 0, warnings: 0
        module tsmc13.CLKBUFX3:v
                errors: 0, warnings: 0
        module tsmc13.CLKINVX1:v
                errors: 0, warnings: 0
        module tsmc13.ADDHXL:v
                errors: 0, warnings: 0
        module tsmc13.DFFRX1:v
                errors: 0, warnings: 0
        module tsmc13.DFFSX1:v
                errors: 0, warnings: 0
        primitive tsmc13.udp_dff:v
                errors: 0, warnings: 0
                Caching library 'tsmc13' ....... Done
                Caching library 'worklib' ....... Done
```

```
        Elaborating the design hierarchy:
  DFFSX1 \num_reg[9]  ( .D(next_num[9]), .CK(clk), .SN(n54), .Q(addr[9]) );
                    |
ncelab: *W,CUVWSP (./PTM_syn.v,70|20): 1 output port was not connected:
ncelab: (/theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v,18500): QN

  DFFSX1 \num_reg[2]  ( .D(next_num[2]), .CK(clk), .SN(n54), .Q(addr[2]) );
                    |
ncelab: *W,CUVWSP (./PTM_syn.v,88|20): 1 output port was not connected:
ncelab: (/theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v,18500): QN

  DFFSX1 \num_reg[1]  ( .D(next_num[1]), .CK(clk), .SN(n54), .Q(addr[1]) );
                    |
ncelab: *W,CUVWSP (./PTM_syn.v,89|20): 1 output port was not connected:
ncelab: (/theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v,18500): QN

  DFFSX1 \num_reg[0]  ( .D(next_num[0]), .CK(clk), .SN(n54), .Q(addr[0]) );
                    |
ncelab: *W,CUVWSP (./PTM_syn.v,94|20): 1 output port was not connected:
ncelab: (/theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v,18500): QN

        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 = 363  Annotated = 100.00% -- No. of Tchecks = 207  Annotated = 100.00%
                                        Total      Annotated      Percentage
                 Path Delays              363             363          100.00
                      $width               69              69          100.00
                   $setuphold             138             138          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.DFFSX1:v <0x1e0f713d>
                        streams:   2, words:   106
                tsmc13.XOR2X1:v <0x7c0db446>
                        streams:   4, words:   284
                worklib.PTM:v <0x75e0bf38>
                        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:              103      25
                UDPs:                  23       1
                Primitives:           247       7
                Timing outputs:       138      14
                Registers:             31      11
                Scalar wires:         166       -
                Expanded wires:        10       1
                Always blocks:          1       1
                Initial blocks:         3       3
                Cont. assignments:      1       1
                Pseudo assignments:     2       2
                Timing checks:        207      29
                Interconnect:         270       -
                Delayed tcheck signals:  69      25
                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* : 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 30810 NS + 0
./PTM_tb.v:97           $finish;
ncsim> exit
[dld0026@ic26 ~/Lab_3]$
```

心得與討論：

1. 這次遇到比較大的問題是一開始不太懂 ROM 裡面的 data 是怎麼存放的，研究了一下才理解 address 和 data 分別代表的意義可以想成很多抽屜(address)與裡面放的東西(data)。

2. 一開始設計 State Transition Graph 的時候，動不動就把 next_state 連回 S0，後來發現其實在某些情況下可以直接連到 S1，讓電路效率提升。

3. 嘗試 State Reg.、Next state logic、Output logic 的 FSM coding style，我覺得這種表示法在把 PTM 改寫成 BON (bonus)的時候比較不容易刪到不該刪的 code，因為 Next state logic 不改變的形況下，根本不用處理那個 block，只要專心處理 output 就好。

4. 上次因為 environment variable "–X" 設定錯誤的關係，沒有使用到 dv，這次第一次用，覺得 dv 拉近了我與真實電路的距離。
   另外，雖然上次作業因為 dv 打不開，分數應該搞砸了，不過如果因為我的關係不會有人再犯這個有點蠢的錯誤，似乎就不那麼難過了 >_<

» 附件
1. nwave.PNG (40 KB)

(吳紹齊, wsc861029@gmail.com, 2018-04-21 20:57)
我路過回一下，你是不是 "-X" 打成小寫了(?)

(樊浩賢, fanhouin@gmail.com, 2018-04-21 21:00)
喔!原來是這樣，我可以開啟了，謝謝你。

回應 關閉