Lab3_Bonus Pattern Matching +

105031212 吳紹齊

一、目標

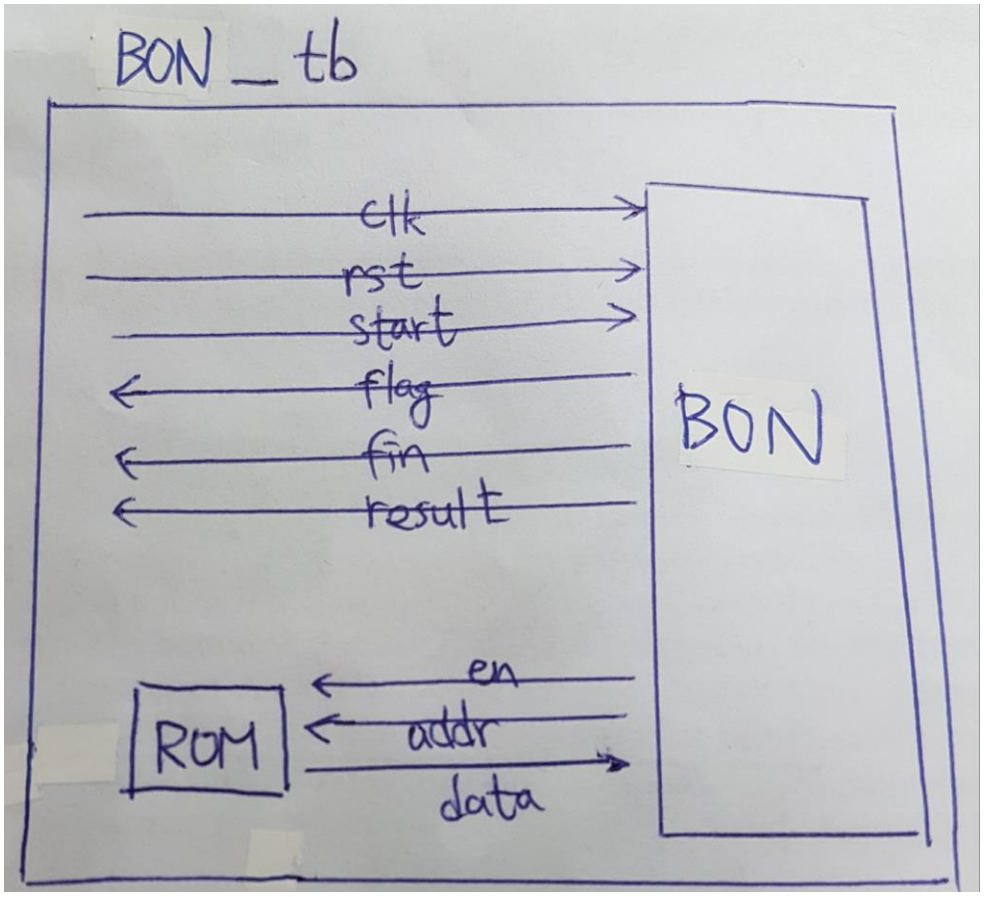
請找符合以下特徵的字串

$$10(100)+(110(100)+)+11$$

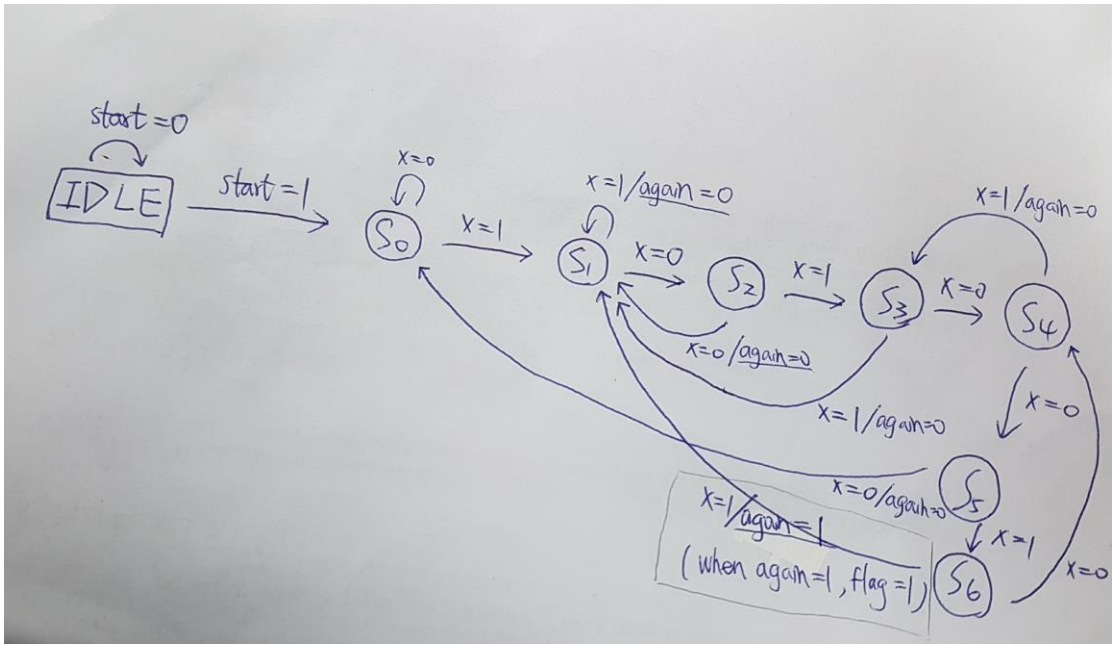
(+ 表示出現次數 >= 1)

二、設計

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

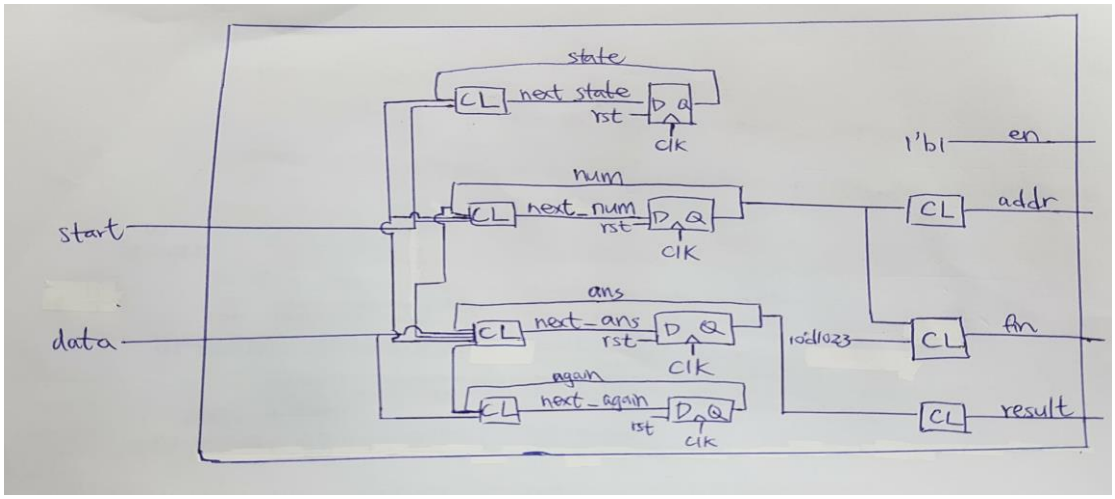


我設計了一個 Mealy Machine 來達成題目的要求，以下是這個 BON 的 State Transition Graph：

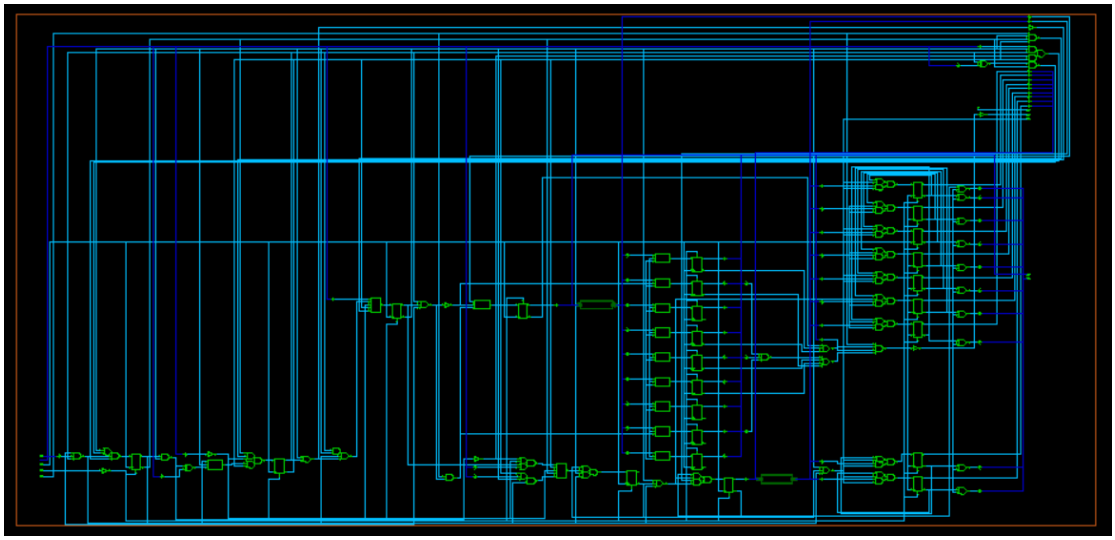


1. 若 Start =1，離開 IDLE 狀態，開始從 address[0] 讀 ROM 的 data。(透過 Counter (num) 讓我能夠從 address[0]往下讀到 address[1022])
2. 判斷 data[0]的值，訂出分別為 1 和 0 時，下一個 state 該如何變化。
3. 與 Lab3 的 PTM 最大的差異在於另一個旗幟 again 的引入，當達成第一次 10(100)+11 時，again 升起(=1)，此後除了(100)+循環以外，state 都只能往前推進，否則 pattern 就錯誤了，因此在回頭的情況下 again 下降(=0)，偵測從頭開始。
4. 抵達 S6 時，若輸入 data[0]為 1，again 為 1，則完整 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(=8)。

ncverilog 模擬結果：

(make sim)

```
[dld0026@ic26 ~/Lab_3b]$ make sim
ncverilog header.v BON.v BON_tb.v +access+r
ncverilog: 14.10-s005: (c) Copyright 1995-2014 Cadence Design Systems, Inc.
file: header.v
file: BON.v
        module worklib.BON:v
                errors: 0, warnings: 0
file: BON_tb.v
        module worklib.BON_tb:v
                errors: 0, warnings: 0
                Caching library 'worklib' ....... Done
        Elaborating the design hierarchy:
        Building instance overlay tables: ..................... Done
        Generating native compiled code:
                worklib.BON:v <0x07a45e8c>
                        streams:    9, words:  5230
                worklib.BON_tb:v <0x3874d57b>
                        streams:    6, words:  8351
        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:          5       5
                Initial blocks:         3       3
                Cont. assignments:      3       4
                Pseudo assignments:     2       2
                Simulation timescale:  100ps
        Writing initial simulation snapshot: worklib.BON_tb:v
Loading snapshot worklib.BON_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 'BON.fsdb'
*Verdi3* : Begin traversing the scopes, layer (0).
*Verdi3* : End of traversing.

GET ! addr =    18 , your_flag = 1 , ans_flag = 1
GET ! addr =    49 , your_flag = 1 , ans_flag = 1
GET ! addr =    71 , your_flag = 1 , ans_flag = 1
GET ! addr =    89 , your_flag = 1 , ans_flag = 1
GET ! addr =    95 , your_flag = 1 , ans_flag = 1
GET ! addr =   123 , your_flag = 1 , ans_flag = 1
GET ! addr =   151 , your_flag = 1 , ans_flag = 1
GET ! addr =   197 , your_flag = 1 , ans_flag = 1

Result =     8 , Answer =     8
!!!!! ACCEPTED !!!!!

Simulation complete via $finish(1) at time 30810 NS + 0
./BON_tb.v:97           $finish;
ncsim> exit
```

(make syn)

```
[dld0026@ic26 ~/Lab_3b]$ make syn
ncverilog header.v BON_syn.v BON_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: BON_syn.v
        module worklib.BON_DW01_inc_0:v
                errors: 0, warnings: 0
        module worklib.BON_DW01_inc_1:v
                errors: 0, warnings: 0
        module worklib.BON:v
                errors: 0, warnings: 0
file: BON_tb.v
        module worklib.BON_tb:v
                errors: 0, warnings: 0
file: /theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v
        module tsmc13.INVX3:v
                errors: 0, warnings: 0
        module tsmc13.AOI211X1:v
                errors: 0, warnings: 0
        module tsmc13.AOI221XL: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.NAND2BX1: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.OAI211X1:v
                errors: 0, warnings: 0
        module tsmc13.OAI22XL:v
                errors: 0, warnings: 0
        module tsmc13.OAI221XL:v
                errors: 0, warnings: 0
        module tsmc13.OAI31XL: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[8]  ( .D(next_num[8]), .CK(clk), .SN(n131), .Q(addr[8]) );
                      |
ncelab: *W,CUVWSP (./BON_syn.v,71|20): 1 output port was not connected:
ncelab: (/theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v,18500): QN

  DFFRX1 again_reg ( .D(n52), .CK(clk), .RN(n131), .QN(n40) );
                   |
ncelab: *W,CUVWSP (./BON_syn.v,72|17): 1 output port was not connected:
ncelab: (/theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v,18240): Q
```

```
  DFFRX1 again_reg ( .D(n52), .CK(clk), .RN(n131), .QN(n40) );
                   |
ncelab: *W,CUVWSP (./BON_syn.v,72|17): 1 output port was not connected:
ncelab: (/theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v,18240): Q

  DFFSX1 \num_reg[1]  ( .D(next_num[1]), .CK(clk), .SN(n131), .Q(addr[1]) );
                      |
ncelab: *W,CUVWSP (./BON_syn.v,90|20): 1 output port was not connected:
ncelab: (/theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v,18500): QN

  DFFSX1 \num_reg[3]  ( .D(next_num[3]), .CK(clk), .SN(n131), .Q(addr[3]) );
                      |
ncelab: *W,CUVWSP (./BON_syn.v,91|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(n131), .Q(addr[0]) );
                      |
ncelab: *W,CUVWSP (./BON_syn.v,94|20): 1 output port was not connected:
ncelab: (/theda21_2/CBDK_IC_Contest/cur/Verilog/tsmc13.v,18500): QN

ncelab: *W,CUSFNF: The SDF file "./BON.sdf" not found..
                       $sdf_annotate(`SDFFILE, bon);
                                     |
ncelab: *W,CUSSTI (./BON_tb.v,43|15): This SDF System Task will be Ignored..
        Building instance overlay tables: ..................... Done
        Generating native compiled code:
                tsmc13.ADDHXL:v <0x0b9e90ed>
                        streams:   4, words:   284
                tsmc13.DFFRX1:v <0x4d4d4aa3>
                        streams:   2, words:   106
                tsmc13.DFFSX1:v <0x09870b6c>
                        streams:   2, words:   106
                tsmc13.XOR2X1:v <0x7c0db446>
                        streams:   4, words:   284
                worklib.BON:v <0x280260a5>
                        streams:   1, words:   115
                worklib.BON_tb:v <0x031d64f6>
                        streams:   6, words:  8850
        Building instance specific data structures.
        Loading native compiled code:     ..................... Done
        Design hierarchy summary:
                                Instances  Unique
                Modules:              110      29
                UDPs:                  24       1
                Primitives:           265       7
                Timing outputs:       146      27
                Registers:             32      11
                Scalar wires:         175       -
                Expanded wires:        10       1
                Always blocks:          1       1
                Initial blocks:         3       3
                Cont. assignments:      1       1
                Pseudo assignments:     2       2
                Timing checks:        216      28
                Simulation timescale:  1ps
        Writing initial simulation snapshot: worklib.BON_tb:v
Loading snapshot worklib.BON_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 'BON_syn.fsdb'
*Verdi3* : Begin traversing the scopes, layer (0).
*Verdi3* : End of traversing.

GET ! addr =   18 , your_flag = 1 , ans_flag = 1
GET ! addr =   49 , your_flag = 1 , ans_flag = 1
GET ! addr =   71 , your_flag = 1 , ans_flag = 1
GET ! addr =   89 , your_flag = 1 , ans_flag = 1
GET ! addr =   95 , your_flag = 1 , ans_flag = 1
GET ! addr =  123 , your_flag = 1 , ans_flag = 1
GET ! addr =  151 , your_flag = 1 , ans_flag = 1
GET ! addr =  197 , your_flag = 1 , ans_flag = 1

Result =    8 , Answer =    8
!!!!! ACCEPTED !!!!!

Simulation complete via $finish(1) at time 30810 NS + 0
```

心得與討論：

1. BON 是改良版的 PTM，其實如果曾經花大量時間在研究 FSM 的 code 的話，要想到怎麼寫不難，可是一開始還是有卡關，因為我以為題意是「找出各個不能再延長的 pattern」，這樣的話可能還要預先讀下一段 data，似乎難度就高很多，後來重新讀題目後，才理解原本的題意。不過也想跟助教討論一下如果是前者有沒有可能做出來？

2. 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 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。