ICCAD 2018 CAD Contest

Watermarking-based Intellectual Property Core Protection Scheme

Chip Implementation Center (CIC), NARL

Contents

0. Announcements…….…....P2

1. Introduction ……...…....P3

2. Problem Statement…….....P3

3. Inputs……..P6

4. Requirements……..P8

5. Outputs……P9

6. Grading Criteria……..P9

7. Reference……P10

8. Verilog-xl Guideline……...P11

9. Alpha Test Announce…….P11

10. Beta Test Announce……..P11

10. FAQ…….....P13

0. Announcements

September

- 2018-09-06- Problem D FAQ is updated
- 2018-09-04- Problem D FAQ is updated
- 2018-09-03- Problem D FAQ is updated

August

- 2018-08-31- Problem D FAQ is updated
- 2018-08-21- Problem D FAQ is updated.
- 2018-08-16- Problem D FAQ is updated.
- 2018-08-15- Problem D FAQ is updated.
- 2018-08-06- Problem D Beta Test announced.

July

- 2018-07-31- Problem D FAQ is updated.
- 2018-07-26- Problem D FAQ is updated.
- 2018-07-24- Problem D FAQ is updated.
- 2018-07-18- Problem D FAQ is updated.
- 2018-07-10- Problem D FAQ is updated.
- 2018-07-10- Problem D Alpha Test announced.

June

- 2018-06-20- Problem D FAQ is updated.
- 2018-06-19- Problem D FAQ is updated.
- 2018-06-08- Problem D verilog-xl is announced (P11).
- 2018-06-08- Problem D Testcase is modified.
- 2018-06-07- Problem D Testcase is modified.
- 2018-06-07- Problem D FAQ is updated.
- 2018-06-06- Problem D FAQ is updated.
- 2018-06-01- Problem D FAQ is updated.

May

- 2018-05-03- Problem D FAQ is updated.

April

- 2018-04-27- Problem D Testcase is announced.
- 2018-04-12- FAQ is updated.

February

- 2018-02-06- Problem is released.

Watermarking-based Intellectual Property Core Protection Scheme

Chip Implementation Center (CIC), NARL

1. Introduction

System on a Chip(SOC)因為市場需求配合深次微米技術的進步、設計複雜度增加、設計速度增加與功率降低而逐漸的成為發展趨勢，讓設計的方式也逐漸改變，矽智產(Intellectual Property, IP)重複使用的觀念變成主要的發展要素，許多的 IP 提供者開發優良的的矽智產，讓設計可以快速整合成 SOC，達到增加生產力與快速實現的特性，而 IP 提供者可以透過法律保護、版權宣告等來保護矽智產，另外，還有利用加入浮水印的方式來保護 IP。浮水印嵌入的技術目前有幾種方式，主要是於設計時於不改變設計的規格，如功能等，將浮水印嵌入到電路中，而且於電路的設計流程中都可以加入浮水印，例如利用合成時的限制條件來加入浮水印、設計電路執行順序(scheduling)、電晶體層級、P&R、Layout 上與電路連接拓樸、模擬模型、FPGA 其他資源、DSP 的 IIR 濾波器的 DB 值、與測試電路中加上偵測的電路等等。本題目是利用類似浮水印(watermarking)的概念將矽智產的作者資訊編碼並隱藏到 IP 電路裡，讓使用者在使用 IP 時，其功能不會受到影響，但能讓 IP 的提供者可以利用這個隱含的字串來證明其為 IP 的擁有者，達到 IP 保護的功能。

2. Problem Statement

要解釋 watermark-based 的 IP 保護方法，首先必須將循序電路(sequential circuit)看成是一個 Finite State Machine (FSM)。由過去的研究(如00)可知，watermark-based 的 IP 保護方法可應用到 Completely 或者是 Incompletely Specified FSM (i.e., CSFSM or ISFSM) 上面。若為 ISFSM 的 IP，我們可以利用其 State Transition Graph (STG)裡頭沒有被定義到的 transitions，將要隱藏的字串，配合原先的 STG，embed 到這些新增的 transitions 當中。在另外一方面，若要保護的 IP 為 CSFSM，則沒有可供利用的 transitions，因此，FSM 無法加入浮水印，須輸出”It is CSFSM!!”的訊息，代表此 FSM 為 CSFSM，無法嵌入 watermark。無論如何，這個 watermark 隱含的程序必須要足夠強韌，以避免被破解。

在這個問題當中，我們參照了以往他人的做法，提供了一種簡單的 watermark-based 的 IP 保護方法。我們將用以下的例子來說明如何將數個字串隱含到一個循序電路中：

1. 首先，利用 hash function 將要加入的字串(例如 CAD Contest)轉換成另一個加密編碼的字串，由於 hash function 的不可逆特性，破解者將無法只修改部分電路，達到破解的目的: 例如利用 md5 這個程式將要隱含的字串(CAD Contest、Copyright @ 2018 以及 IP Protection ver. 1)，產生三組 128-bit 的字串，評分單位會替換這三組字串後進行驗證:

md5(CAD Contest) = 0452e2548132d44736a92ca879fff6ca

md5(Copyright @ 2018) = cb952dea72401ffbdea26a32c5856758

md5(IP Protection ver. 1)= 47849f79030673e482467bb2358d3d2c

2. 根據 IP 輸入輸出的數目，定義好每組 128 bits 所對應的輸入與輸出順序。

   例如: 假設 IP 有 2 個輸入，1 個輸出，那麼先將 CAD Contest 這個字串經過 MD5 產生的結果，亦即二進位的字串"000001000101001…"，依據：iioiioiioiio…('i'為 input，'o'為 output)的順序做對應。經對應後，輸入依序為 00, 00, 00, 10, 00, …, 輸出依序為 0, 1, 0, 1, 1, …。換句話說，我們希望能在原始的電路中加入 watermarking 的功能，也就是說對於{00, 00, 00, 10, 00, …}這樣的 input sequence，能產生 output 為{0, 1, 0, 1, 1, …}的 state transition sequence，再來把第二組訊息(Copyright @ 2018)嵌入 FSM 中，以此類推，也可以更改嵌入的順序，或三組訊息同時嵌入。

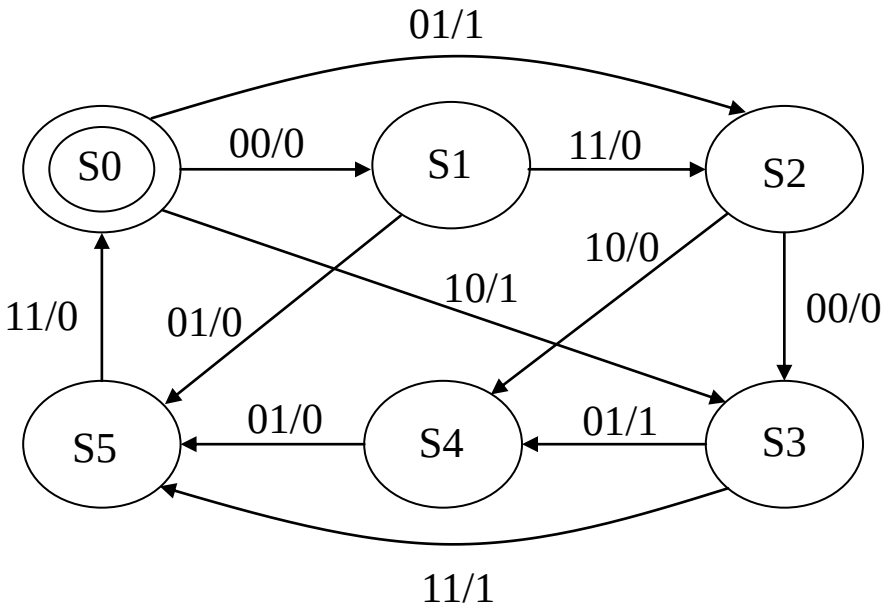
3. 圖一為一個循序電路之 State Transition Graph 的範例，此原始電路並不會產生如步驟 2 所述的 state transition sequence。但是我們注意到此電路是一個 Incompletely Specified FSM，也就是說在它的 STG 中，有一些 state transitions 並沒有被定義，譬如對 S1 這個 state 而言，"10"這個 input pattern 並沒有相對應的 transition，也就是說，它代表了一個"don't-care"的 condition。因此，我們可以利用這些未被定義的 transitions，將步驟 2 中的 watermarking "CAD Contest"這個字串之 MD5 之輸出與所需的輸入與輸出的關係，以及所對應的 state transition sequence，加入到圖一的範例中。要注意的是，原先已經存在的 transitions 與 states 都不能改變，只能利用未定義的 transitions，如果未定義的 transitions 無法完成，則可以在不改變原來功能的情況下，增加 states，達到嵌入 watermarking 的效果。

4. 為了簡化說明起見，我們假設只利用 5 組輸入{00, 00, 00, 10, 00}與 5 組輸出{0, 1, 0, 1, 1}來當做我們目標 embedded 的位元組，起始的狀態為 S0。以下是幾種不同 watermarking 的可能性：

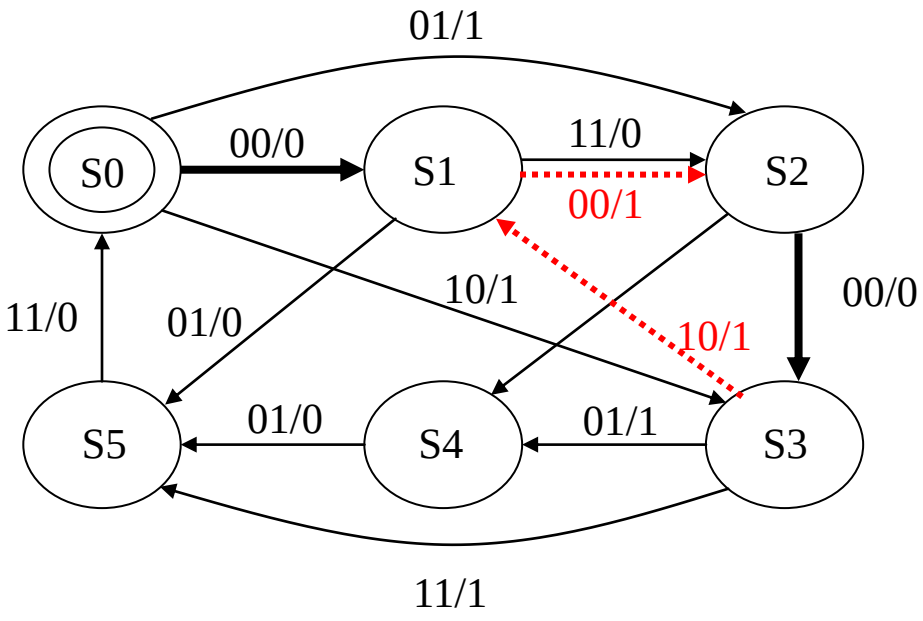
   a. 圖二是加入 watermark 的結果，我們使用了兩個原先的 transitions(粗黑線)，以及兩個新增的 transitions (紅色虛線)即可以達到 watermarking 的效果。因此，這個解答的 Transition Cost 為 2，State Cost 為 0。
   b. 圖三是相同的原始 STG 與相同的 embedded 位元組的另一個解答，這個解答需要新增 2 個 transitions，因此，這個解答的 Transition Cost 為 2，State Cost 為 0，但是起始的狀態為 S2，需要產生 input 為 00, 11 這組 latency sequence。
   c. 圖四是相同的原始 STG 與相同的 embedded 位元組的另一個解答，這個解答需要新增 4 個 transitions 與 4 個 states，因此，這個解答的 Transition Cost 為 4，State Cost 為 4。

5. 將所有的 Watermark 字串(128 bits)嵌入到原來的循序電路，字串可以個別嵌入，順序也可以隨意，也可以一次考慮到所有要嵌入的字串。驗證時每組會分開驗證，由

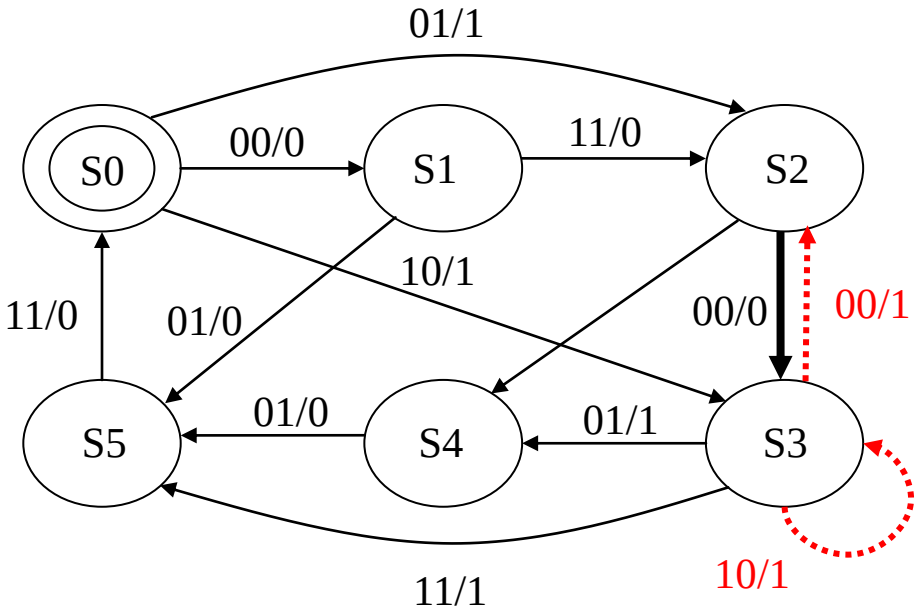
起始狀態加上 latency sequence 後，開始驗證。



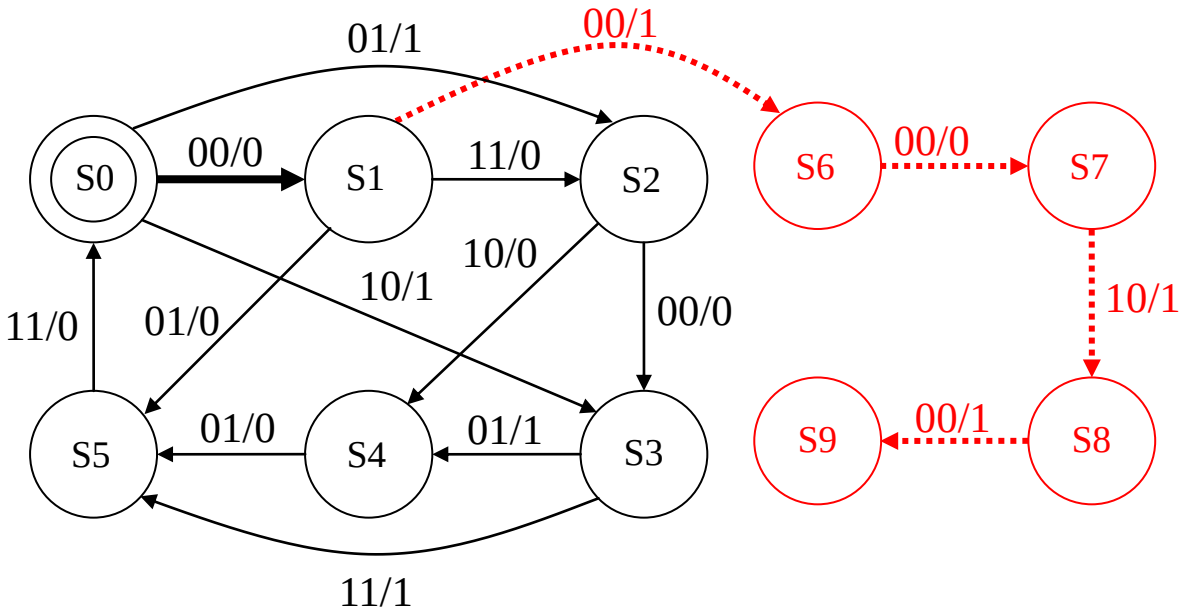
圖一: 原始的循序電路



圖二: 加入 watermark 後的循序電路(範例 a)



圖三: 加入 watermark 後的循序電路(範例 b)



圖四: 加入 watermark 後的循序電路(範例 c)

3. Inputs

(1) 欲加入做 watermark128 位元字串，組數固定為三組。

(2) Sequential circuit in Berkeley KISS (keep it Simple Stupid) format (a .kiss file)

我們將限制我們的測試電路之(輸入數目+輸出數目)小於 128，因此，需要一個以上的 transitions 才可以將此兩組 128 bits 的字串隱含進去。

KISS 的格式如下:

```
.i <num-inputs>
.o <num-outputs>
[.p <num-terms>]
[.s <num-states>]
[.r <reset-state>]
<input> <current-state> <next-state> <output>
...
（更多的狀態表欄位 state transitions)
...
<input> <current-state> <next-state> <output>
.e
```

其中 ---

.i: 輸入數目(number of inputs)

.o: 輸出數目(number of outputs)

.s: 狀態數目(number of states)

.p: 狀態表的項數(terms in state table)

.r: 起始狀態

.e: 檔案結束

狀態表欄位依序分別為：(i) input pattern, (ii) present state, (iii) next state, (iv) output response。輸入的 pattern 是由一串列的{0, 1, -}所組成，而輸出的結果也是由一串列的{0,1,-}所組成，”-“代表 don’t care，故 input pattern 代表 cube cover 的狀態，例如 1-0 代表 100 與 110 的組合，所以於狀態表欄位輸入如果出現”-“，代表可以展開成數個項，例如 “ 1-- S0 S1 11 ” 可以展開成 4 個狀態表的項:

100 S0 S1 11
101 S0 S1 11
110 S0 S1 11

111 S0 S1 11

最後結果因為計算 Transition Cost 是以項次計算，故請盡可能合併，不要展開以減少 Transition Cost。

上圖一的 STG 其對應的 KISS 格式如下:

```
.i 2
.o 1
.s 6
.p 11
.r s0
00 s0 s1 0
01 s0 s2 1
10 s0 s3 1
11 s1 s2 0
01 s1 s5 0
10 s2 s4 0
00 s2 s3 0
01 s3 s4 1
11 s3 s5 1
01 s4 s5 0
11 s5 s0 0
.e
```

4. Requirements

將題目給予的三組 128 位元字串，從 MSB 開始，依據電路的輸入與輸出數目，以及 Problem Statement 中所描述的方法，切割成為數組依序的輸入與輸出 patterns，在 128 除以"電路的輸入加輸出數目"不能整除的部分位元補 0，並將這組輸入與輸出的 sequence 加入到循序電路中，在不改變原有電路功能，而且在新增最少的狀態(states)的情況下，找出增加最少的未定義的 transitions 下的最佳 watermarking 實現方式，其中第一組與第二組的順序可以調換，注意：最佳的實現方式不一定將 watermark sequence 從起始狀態(reset state)開始，換句話說，可以執行一段 latency sequence 之後，將電路帶到某一個 state，然後再實現此隱含的 watermark sequence，此外，程式需要偵測電路是否是 Completely FSM，如果是則輸出無法加入 watermark。編譯好的程式需放在提供的目錄最上層，於提示符號下執行 "**time your_program_name –i ifsm.kiss –o ofsm.kiss –m md5_1.dat –m md5_2.dat –m md5_3.dat"**，其中 ifsm.kiss2 是預定要加入 watermark 的

原始電路，md5_1.dat, md5_2.dat, md5_3.dat 是分別存放三組 128 位元字串的文字內容的檔案，而 ofsm.kiss2 則是已將三組 watermark“一起”加入的電路，電路中含有三組一起嵌入的 watermark，但是驗證時是一組一組分開檢驗，每次只驗證一組，每一組都是從 reset state，加上 latency sequence(如果有)，再開始驗證是否嵌入正確的 watermark。

5. Outputs

(1) 隱含 watermark 字串之 KISS 格式的循序電路(ofsm.kiss)。

(2) 產生每組 watermark output response 的 input sequence (if necessary, 包含前面的 latency sequence)。共產生六個檔案: md5_1.env, md5_2.env, md5_3.env, md5_1.ini, md5_2.ini, md5_3.ini，其中 md5_?.env 是以 16 進位表示之 32 位元數字，代表需要多少組的 input sequence，md5_?.ini 則是對應的 md_?.dat 的 latency sequence，各組間可以使用”_”區隔，範例如下(假設輸入為 2 個位元):

md5_1.env : 00000000

md5_2.env: 00000003

md5_3.env: 0000000e

md5_1.ini:

md5_2.ini: 00_00_01

md5_3.int: 00_00_01_10_10_01_01_00_01_00_00_01_11_10

附註: md5_1.ini 為空的檔案，因為從 reset 開始就嵌入 watermark，故不需要 latency sequence。

6. Grading Criteria

- Completeness: able to demonstrate the watermark sequence without changing the circuit functionality. We will apply verification on your revised circuit to check the functional equivalence.
- The program should be finished within 1 hour for one testcase on the specified machine.
- 所有 testcase 裡(包含給定與隱藏的 testcases)，功能正確數目最多的獲勝，如果功能正確的數目相同時，則比 Hardware cost= number of increased states + number of increased transitions，最低者獲勝，如果 Hardware cost 相同時，則比執行時間，越少者獲勝。

7. Reference

[1] A.L. Oliveria, "*Robust Techniques for Watermarking Sequential Circuit Designs"*, Proc. IEEE/ACM Design Automation Conference, June 1999, pp. 837-842.

[2] I. Torunoglu and E. Charbon, *"Watermarking-Based Copyright Protection of Sequential Functions"*, IEEE Journal of Solid-State Circuits, Vol. 35, No. 3, Feb. 2000, pp. 434-440.

8. verilog-xl 教學

請先執行 source /cad/cadence/CIC/incisiv.cshrc

CAD Tool:

Software:innovus

source /cad/cadence/CIC/innovus.cshrc

source /cad/cadence/CIC/license.csh

Software:VCS、

version: N-2017.12

source /cad/synopsys/CIC/vcs-mx.cshrc

version: vF-2011.12-SP1

source /cad/cadence/CIC/vcs.cshrc

Software: VERILOG-XL

version: 08.20.001-d

source /cad/cadence/CIC/incisiv.cshrc

9. Alpha Test Announce

test1	test2	test3	test4	test5	test6	Total	Time	Function Work	Rank
0	51	22	16	20	38	147	2117.675	6	1
0	50	11	5	-53	0	13	391.916	5	2
0	55	0	17	22	52	146	0.135	5	3
0	54	14839	11927	16296	0	43116	0.121	5	4
0	51	22	16	0	0	89	0.039	4	5

10. Beta Test Announce

test1	test2	test3	test4	test5	test6	Totoal	Time	Function Work	Rank
0	50	21	5	-9	29	96	3799.712	6	1
0	50	21	5	-9	29	96	6037.134	6	2
0	54	22	16	20	33	145	0.056	6	3
0	112	212	315	403	1368	2410	0.204	6	4
0	54	24	3	7	0	88	11.239	5	5

10. FAQ

Q1."編譯好的程式需放在提供的目錄最上層，於提示符號下執行 “time your_program_name -i ifsm.kiss -o ofsm.kiss -m md5_1.dat -m md5_2.dat -m md5_3.dat”"，請問是先把 c++檔案先編譯成 binary 檔後，把 binary 檔和 kiss 檔放在同一個 folder 下，並且在 folder 路徑下執行"time ... " 的指令嗎?

A1.沒錯，your_program_name 就是編譯好的 binary 執行檔檔名，而且其他的輸入檔案則須放在與執行檔同一個目錄下才能讀取的到，而這個目錄要在繳交資料目錄結構的最上層，方便評審作業。

Q2.請問有沒有可能有一些 state 是從 s0 到達不了的?
如果有些是 s0 到達不了的 state，那麼就算整個 graph 不是 complete，他仍不能加 watermark。

A2.所有的 state 都能從 start state (S0)走到。

Q3.輸出”It is CSFSM!!”的訊息格式相關規定是否可以再敘述一些呢?

A3.沒有規範，只要輸出 “It is a complete FSM” 或是”It is CSFSM!!” 之類的即可。

Q4.state 的名稱會如同範例:s0,s1,s2 嗎?

A4.會如範例，以 S 開頭加上一個數字。

Q5.請問有沒有一些 state 無法從 start state 到達?

A5:沒有，所有的 state 都能走到。

Q6.請問 state 的命名方式和範例給的 s0, s1, s2 等，會一樣嗎?

A6:會如範例，以 S 開頭加上一個數字。

Q7.請問 input 檔案的 transition terms，會從 state 小的按照順序排列到 state 大的嗎?

A7: 有些順序會不一樣，不一定由小到大，也可能有跳號的情形。

Q8.如果偵測到是 complete FSM 時，該如何印出訊息? 題目的敘述當中並沒有仔細規範。

A8:沒有規範，只要印出 It is a complete FSM 之類的即可。

Q9.請問"輸入數目"、"輸出數目"、"狀態數目"和"狀態表的項數"的最大值分別為多少?

A9.基本上程式應該考量不同設計的 FSM，不應該有上述數值的限制，請參賽者依據測試資料自行評估。

Q10.請問 md5.ini 檔案中紀錄的 latency sequence，一定要用底線"_"做區隔嗎?

A10.因為這個檔案須以 verilog 格式，讓 testbench 讀取，故需要以底線當區隔，沒有底線也可以讀取，所以不加底線也可以。

Q11.

狀態表欄位依序分別為：(i) input pattern, (ii) present state, (iii) next state, (iv) output response。輸入的 pattern 是由一串列的{0, 1, -}所組成，而輸出的結果也是由一串列的{0,1,-}所組成，"-"代表 don't care，故 input pattern 代表 cube cover 的狀態，例如 1-0 代表 100 與 110 的組合，所以於狀態表欄位輸入如果出現"-"，代表可以展開成數個項，例如 " 1-- S0 S1 11 " 可以展開成 4 個狀態表的項:

而輸出的結果也是由一串列的 {0,1,-}所組成

這個敘述中，輸出也會有 don't care 的出現，假如有一組 state transition 是

"100 S0 S1 1-"

那這組 transition 會在同一個輸入一次產生兩個不同的輸出嗎?

A11.輸出為 don't care 時，代表此時輸出為 0 或為 1 都可以， 但是只能選擇一個輸出值，為了簡化題目，目前輸出沒有 don't care 的情形，請同學不用考量這個狀況。

Q12. 我想請問此題可以使用線上開源的 SAT solver 作為程式碼的一部份嗎？

A12. 由於審查者無法一一檢查程式碼與判斷是否使用開放原始碼，加上先針對執行檔進行驗證，故將不限制使用開源軟體。"

Q13. 想請問 Problem D 在最後 testbench 驗證的時候

它顯示 Error，後面的 in / expect result / out 分別代表什麼?如下

Time: 80 Error in WaterMark ==> in: 0 expect result: 04 out =01

驗證正確通過時會顯示什麼呢? -2018/06/06

A13. in: FSM 的輸入，

out: 則是 FSM 的輸出，

expect result: 是指有嵌入字串後預期的輸出，

當輸出與預期的輸出不同時會有錯誤訊息，所有的 pattern 都正確則不會有任何錯誤訊息。

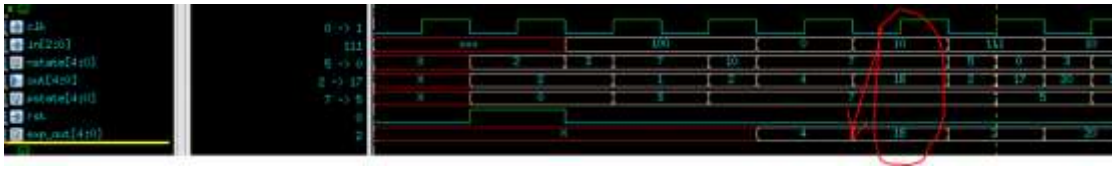
另外，關於環境部分，近期將會安裝。

Q14. ，如果 FSM 的 input 是 9 bits ，output 是 10 bits(就像測資 t3.kiss 裡給的那樣)，以 iiiiiiiiiooooooooo 的順序做對應，但 input+output 的長度是 19，而 watermarking 的長度是 128，128 沒有辦法被 19 整除，請問這樣要怎麼處理?(順帶一題題目上給的 iio 長度 3 也沒有辦法整除 128) -2018/06/06

A14. 如果沒有辦法整除，剩下的輸入/輸出都需要補 0，

最後一組 pattern, 如果剩下 output ，則沒有設定值的 output 補 0， 如果包含 input，則沒有設定值的 input 與 output 都補 0。

Q15. 我們在使用主辦單位提供的 tesbench.v 時，發現了以下問題：
testbench 會取樣 input 出現之前的 output，如圖：
在紅色圈圈的地方，010 的 input，對應的 exp_output 以及我們產生的 output，是相同的。但是 testbech 卻會取樣到前一個 output。 -2018/06/06



A15. testbench 在 clk 負緣時會設定下一個週期的輸入與檢查目前這個週期的輸出與預期的輸出，所以您提到取樣的問題，可能是這個緣故。這組輸入需要搭配下一組的輸出。
A15’ 回答修正如下： -2018/06/07
這的確是 testbench 的 bug，已經修正，麻煩下載新版的驗證程式並幫忙確認是否修正

Q16. 在 Linux 執行會用什麼標準來判定執行時間為 1 小時？ -2018/06/07
A16. 驗證者會使用 time 這個指令紀錄所執行的時間，如果執行時間超過一個小時，驗證者會進行強制中斷動作，並檢查 time 的 CPU 執行時間是否超過一個小時，來確認。

Q17. .kiss 中給的 state 數量如 “.s 7” state 的名字一定會是 S0～S6 嗎？
-2018/06/07
A17. 為了簡化問題，所有的狀態名字都是以"S"開頭，後面加一串數字，數字可能不連續。

Q18. 在驗證時"testbench"裡面的 input/output 的 bits 數是寫死的到時候 alpha test 驗證的時候評審會主動改嗎? 還是我們要在"README"裡面寫上這個步驟呢? -2018/06/07

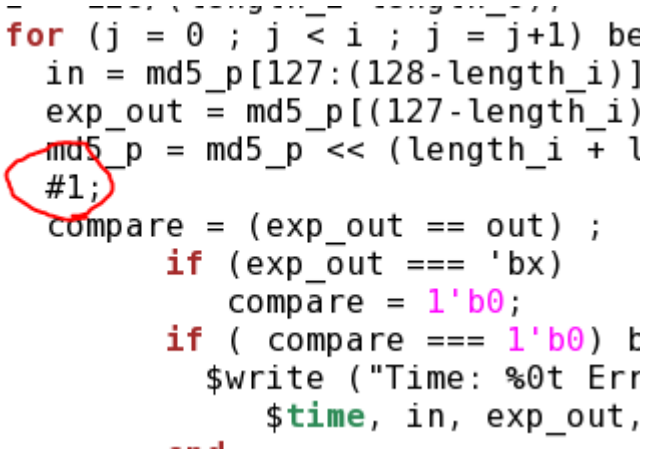
```
module test;

parameter length_i = 3,
          length_o = 5,
          max_ini = 1024;

reg clk, reset;
reg [length_i-1:0] in;
wire [length_o-1:0] out;
reg [length_o-1:0] exp_out;
reg [31:0] cnt[1:0];
```

A18.評審會修改此兩個參數。

Q19. 我們也遇到了和 Q15.相同的問題，請問這個問題是"testbench"有誤嗎?
如果是有誤，會提供新的"testbench"提供驗證測試嗎?
我們自行在"testbench"中
這個位置加入了 1 個時間單位的延遲後再進行取樣比較
這樣就解決了"output 取樣到上一組"的問題
請問這樣的做法是可行的嗎?
如果可行，也是要寫在"README"裡面嗎?還是未來會提供新的"testbench"呢?
-2018/06/07



A19. 原來的驗證程式有錯誤，請下載修正過的版本驗證。
Link: http://iccad-contest.org/2018/tw/problems.html

Q20.當中的 latency sequence，請問這些過渡的 latency 是否有規定 output 必須要是原始電路的 output？題目似乎沒特別定義。
A20.latency sequence 輸出沒有規定，一般都是原始電路的輸出，主要目的是進入開始嵌入浮水印的狀態。

Q21. 1.關於問題當中的 latency sequence，請問這些過渡的 latency 是否有規定 output 必須要是原始電路的 output？題目似乎沒特別定義。
A21. 1. latency sequence 輸出沒有規定，一般都是原始電路的輸出，主要目的是進入開始嵌入浮水印的狀態。

Q22.在使用 CIC machine 的 verilog-xl 時，題目上要我們先執行 source /cad/cadence/CIC/incisiv.cshrc 的指令，但我們執行後都得到 bash: /cad/cadence/CIC/incisiv.cshrc: line 17: syntax error near unexpected token `(' bash: /cad/cadence/CIC/incisiv.cshrc: line 17: `foreach bin_pre (${bin_prefix})'
的問題，無法確實執行
A22. 請先執行 /bin/tcsh

再 source incisiv.cshrc
系統登入的 shell 將修改成 tcsh ， 修改後就不需要執行 /bin/tcsh

A23. 請問 don't care 的問題是不管項，還是用此項展開成數項？
題目稿有講到數項，
在 t3.kiss 的時候，有發現在
state S2 之後，有出現 input 有重複的
1-10----- S2 S2 0100000000
1-1-0---- S2 S2 0100000000
展開後有 32 項是一樣的。
之後有在 state S3, S6, S7, S11 有發現一樣的問題
在 S11：
1---0--1- S11 S0 0000000000
1--0---1- S11 S0 0000000000
展開後有 32 項是一樣的。
我個人認為，如果是不管項，應該是成立
如果是展開項，會有重複，這樣會違反 complete 的問題
A23. Don't care 代表邏輯 0 或邏輯 1 都成立，故可以展開，以兩個位元為例，
0- S2 S2 10
-0 S2 S2 10

Next state 和輸出都相同，故可以展開成 三項:
01 S2 S2 10
00 S2 S2 10
10 S2 S2 10

寫三項與兩項的功能相同，但是最佳的表示方式是使用兩項的表示方式。
而是不是 complete 則是看有無列出所有可能的項次，以上面的例子來說，如果還有加上
11 S2 S2 10 則是 complete 的 FSM， 如果沒有，則為 Incompletely Specified FSM。

Q24. 請問關於 Output 的 env 檔是只要 latency 的數目就好，還是 latency+(128/(input bits + output bits))?
題目的 inputs 裡有提到「最後結果因為計算 Transition Cost 是以項次計算，故請盡可能合併，不要展開以減

少 Transition Cost。」，這句話的意思是最後測試是直接取我們輸出的 kiss 裡的.p 後面數字來做 transition cost 的結算，所以要盡可能合併成由"-"組成的狀態列嗎?

A24. 一般的 FSM 輸入輸出位元數沒有最大的範圍，這個競賽建議可以用 1024-bits 當最大輸入位元數，
超過可以顯示程式無法處理。當然，最好的方式是設計成沒有限制最好。

Q25. 想請問 alpha_test 中，ProblemD 有包含哪些測試?
已知有做 "讀進電路並執行程式" + "計算 cost"，
但之後有做 "用 verilog 檢驗輸出電路的正確性" 了嗎?

A25. 執行後，會將 kiss 檔轉成 verilog 檔，執行是否嵌入正確的浮水印，以及驗證原先的 FSM 是否被破壞。

Q26. 請問讀進的 kiss 檔會有 "故意把多個可以合併的 edge 拆為多項"，
而要我們自己檢查且合併，以降低 cost 的情形嗎?
ex: 1- S0 S1 11 會不會故意被寫為 10 S0 S1 11、11 S0 S1 11 等兩 edge

A26. 會有可以合併的 Transition 項， 故請檢查是否有其可以合併的項，並進行合併。

Q27. .env 檔裡面的數字是包含什麼?
以及加入浮水印後的狀態列都要盡可能合成 don't care 的形式嗎?

A27.
1. .env 的數值代表需要多少組的 input sequence(也就是代表.ini 的組數)，才能由 Reset 的狀態進入到嵌入浮水印的狀態。
2. transition 的數目代表硬體成本，而合成成 don't care 的形式，有可能會減少 transition 的數目。

Q28.想請問 Alpha test 中，rank2 的隊伍對於 test5 的答案為何是負數嗎?

A28. 因為原來的 FSM 沒有將 transition 數目做最佳化，故如果有做最佳化，則 transition 數目會減少，減少的數目比因為浮水印增加
的 transition 數目多時，就會成為負值。

Q29. 我們繳交的 kiss 檔中允許對原題目的 kiss 檔(t2 t3 t4.kiss)在不改變原本功能的情形下做修改嗎?

A29. 化簡或展開 transition 時，不能修改原先的功能， 故在不修改原先 FSM 功能下，能進行修改。

Q30. 原題目的 kiss 電路中，有可能會部分功能錯誤嗎?

例如以下兩組 kiss 電路:
00 S0 S1 10
00 S0 S2 11
A30. 原題目的 FSM 除非筆誤，應該不會有所列情形。

Q31. 請問以下兩組電路可以同時存在嗎?
00 S0 S3 10
00 S0 S3 11
A31. 不能同時存在， 因為這樣就無法確定輸出到底是 10 還是 11。

Q32. 想請問 alpha_test 中，ProblemD 有包含哪些測試?
已知有做 "讀進電路並執行程式" + "計算 cost"，
但之後有做 "用 verilog 檢驗輸出電路的正確性" 了嗎?
A32. 執行後，會將 kiss 檔轉成 verilog 檔，執行是否嵌入正確的浮水印，以及驗證原先的 FSM 是否被破壞。

Q33. 請問讀進的 kiss 檔會有 "故意把多個可以合併的 edge 拆為多項"，
而要我們自己檢查且合併，以降低 cost 的情形嗎?
A33. 會有可以合併的 Transition 項， 故請檢查是否有其可以合併的項，並進行合併。

Q34. 承上，不過關於 Q33 的回覆，我認為這次的題目是"Watermarking-based IP Core Protection"，合理上是要比如何最有效率得嵌入隱藏碼。但若還要做合併...就會牽扯到 FSM Optimization，這顯然是個相當龐大、複雜而且背離原題目的問題，最後也可能導致大家都在比優化的最佳化程度(誰能去除掉最多 state、transition)...而這應該不是此題最初的競賽主軸了吧。
當然我能理解，主辦單位不可能逐一檢查是否有組別做優化電路，但我認為至少要能保證"做測試的電路須為最簡化 FSM"，以確保大家是公平得競爭於"誰能最有效率得完成 Watermarking-based IP Core Protection"。
A34. 這個題目設計時本來就是基於 FSM 邏輯最佳化的問題再加上嵌入隱藏碼的困難度來設計，
因為邏輯化簡早已發展成熟，但是因為加上嵌入浮水印而增加複雜度，讓這樣較複雜的邏輯化簡與最佳化問題成為題目的方向，
單純只嵌入浮水印基本上沒有甚麼難度，但是有效的嵌入和邏輯最佳化也有關聯，因此優化的過程本來就是這個題目的目標，
至於是否只針對增加浮水印電路與周邊電路做最佳化還是對全部電路做最佳化，出題者認為應該要全部電路做比較好，
因此測試題目才有沒有最佳化的題型。

Q35. 請問如果 hardware cost 發生負值為何種意思？
A35. 因為原來的 FSM 沒有將 transition 數目做最佳化，故如果有做最佳化，則 transition 數目會減少，減少的數目比因為浮水印增加的 transition 數目多時，就會成為負值。

Q36. 有關公告的第 24 個回答，當中提到可以假設最大輸入長度為 1024 bits，請問是否想表達最長是 10 位元(十進位制 0~1023)呢？md5 的檔案中會有 128 位元的資料，還是是要在超過 128 位元的部分都補上 0 的意思呢？因為 1024 和 128 位元有一些落差故有此疑問。
A36. 1024bits 是指 FSM 的輸入之最大位元數， 而嵌入訊息的位元數是 128bits， 兩者不同。如果 FSM 輸入與輸出位元和大於 128，或是無法整除 128 時，則嵌入的訊息要補上 0，成為一組完整的輸入與輸出。

Q37. 請問 FSM 除了縮減 transition 數量外，會有需要縮減 state 數量嗎?
A37. 在不改變 FSM 原始的功能下，如果 state 也有機會減少，也可以減少 state，降低 cost。

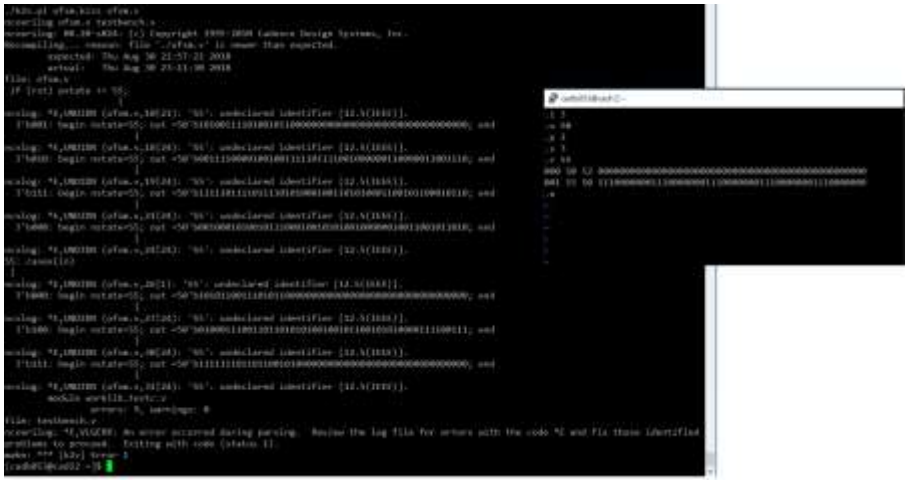
Q38. 請問一下 problem D 的執行時間問題，
如果執行時間已經達到一個小時，你們強制中斷後，程式是有產生結果的，
而且結果最後驗證是通過的，這樣你們最後判斷的時候會判定這個為 fail 還是成功的呢?
A38. 如果強制中斷後，仍有結果產生，則會繼續接下來的驗證，
如果驗證通過，則結果會以功能正常判定。

Q39. 請問 State 的命名有沒有可能是 S01 這種由 0 開頭的 STATE? 如果有的話，S000 跟 S0 是不是代表同一個 STATE ？
A39. 依據 kiss 檔案格式，state 的名稱是字串表示，也就說 S01 和 S1 代表不同的 state，但題目為了簡化，改成狀態名字都是以"S"開頭，後面加一串數字，因此測資不會出現數字重複情形，例如 S01 和 S1 或是 S000 和 S0 等這類情形。

Q40. 我的建議是，在計算 total cost 的時候，不應該把錯誤的 test case 的 cost 也列入計算。
像是此次 beta test，我的 test case 6 function not work，卻將 case 6 的 cost 也加入了我的 total cost，
而 rank 5 的參賽者因為他的 case 6 沒有跑出結果，所以 cost 為 0，
我比較之後發現我 case 1~5 的 total cost 是比他低的，卻因為我的 case 6 有輸出結果，所以最後 cost 比他高。

我想這是比較不合理的，既然這個 case not work，那麼就不應該再把它的 cost 加入最後的結果了。
希望出題老師能夠在 final test 將我的建議列入考量。
A40. 我們同意參賽者的建議， final test 時，會將 function 不正常的 cost 移除，謝謝參賽者的建議。

Q41. 您在 Q7 說過， State 可能會有跳號的情況，是指說如果 FSM 只有 3 個 STATE 的話，可以取名為 S0、S2 與 S5 嗎? 因為，我們在 cad 環境上測了很多種情況發現如果有跳號的命名方式，會出 error，如附圖。



A41. State 可能有跳號情形，是指輸入的 FSM state 可能有跳號情形，但輸出時可以重新指定 State 名稱避免跳號情形，
k2v.pl 對於跳號會有宣告的 bug，故驗驗時，如果有跳號情形，會在宣告 parameter 時，手動增加 沒有定義的狀態編碼，
來繼續驗證。
也就說如果發生如附檔的錯，則要手動修改 ofsm.v 內
parameter S0=2'd0, S1=2'd1, S2=2'd2, S5=2'd3;
應該就可以執行了。

Q42. 續上一個問題，那請問我們輸出的 ofsm.kiss 如果有跳號的情況，大會那邊的 k2v 驗證會通過嗎?
您的回答面提到跳過的號碼大會那邊會手動增加沒有定義的 STATE 編碼，那那些增加的 STATE 編碼也會算在 cost 裡面嗎?
還有一個問題，請問最後的測資會有超過 2^64 次方筆的 transition 嗎?因為您之前說過，input 會到 1024bit，如果沒有使用 don't care 的話 transiton 數是可以到很大量的。
A42.
(1) 如果 FSM 有跳號，驗證者會手動加入跳號的宣告進行評量，而增加的 state 不會計算到 cost 裡，如果沒有功能的錯誤。則會通過。

(2) 測資最大數目的 transition 是 1123 ，但是實際應用可能會比這個數目大，所以不應該以測資來設計。
(3) 的確如果沒有使用 don’t care 的話，transition 的數量會很大，這部分需要靠程式設計與說明該程式能處理最大的數量來解決。

Q43. 請問關於 Q36 的回答說「如果 FSM 輸入與輸出位元和大於 128，或是無法整除 128 時，則嵌入的訊息要補上 0，成為一組完整的輸入與輸出。」，但是題目裡有提到「我們將限制我們的測試電路之(輸入數目+輸出數目)小於 128，因此，需要一個以上的 transitions 才可以將此兩組 128 bits 的字串隱含進去。」
請問兩個說法是否衝突?測資真的會出現輸入與輸出位元和大於 128bits 的狀況嗎?
A43. 請以題目描述為準， 因為簡化題目，故測資的測試電路之輸入與輸出之和將小於 128，問題回覆時沒有注意到已將問題簡化。