

2020 University/College IC Design Contest

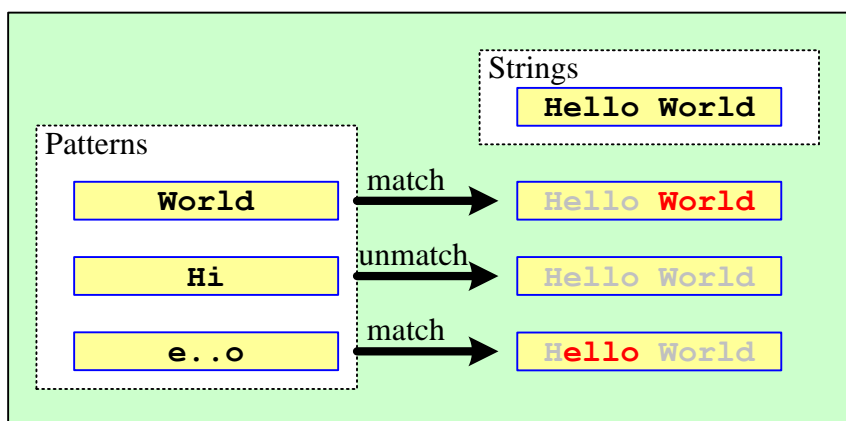
大學部標準元件數位電路設計

String Matching Engine

1.問題描述

請完成一 String Matching Engine (後文以 **SME** 表示)的電路設計，其功能為，題目將依序提供數個字串(後文以 **String** 表示)及關鍵字樣本(後文以 **Pattern** 表示)，SME 電路需比對該 Pattern 是否包含在該 String 中，若有則回應比對成功(match)以及比對到的位置。

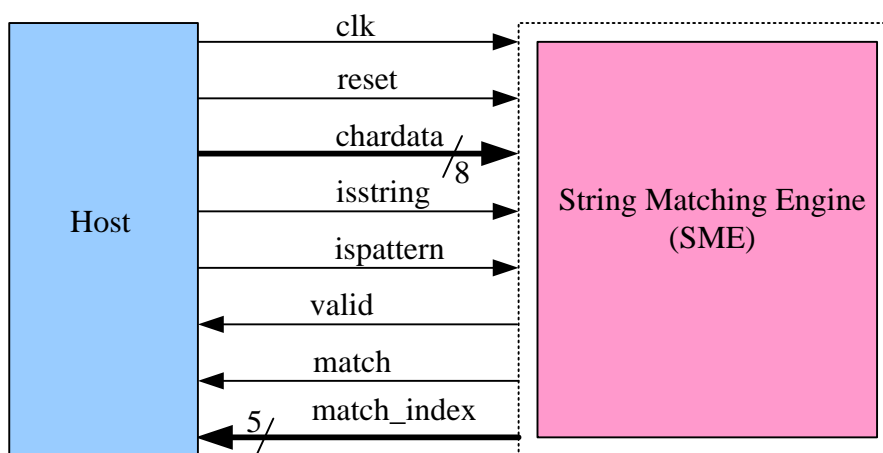
本次 IC 設計競賽比賽時間為上午 08:30 到下午 20:30。當 IC 設計競賽結束後，會根據第三節中的評分標準進行評分。為了評分作業的方便，各參賽隊伍應參考附錄五中所列的要求，附上評分所需要的檔案。



圖一、Pattern Match

2.設計規格

2.1 系統方塊圖



圖二、系統方塊圖

2.2 輸入/輸出介面

表 1 -輸入/輸出訊號

Signal Name	I/O	Width	Simple Description
clk	I	1	本系統為同步於時脈正緣之同步設計。
reset	I	1	高位準”非”同步(active high asynchronous)之系統重置信號。
chardata	I	8	輸入 String 或 Pattern，一次輸入一個 ASCII 字元(8bit)
isstring	I	1	當 isstring 為 high 時，chardata 是屬於 string 資料
ispattern	I	1	當 ispattern 為 high 時，chardata 是屬於 pattern 資料
match	O	1	當 string 和 pattern 比對成功，輸出 match 為 high。
match_index	O	5	當 string 和 pattern 比對成功，輸出 pattern 在 string 中第一個比對成功的位置。
valid	O	1	有效的輸出訊號。當 valid 為 High，表示目前輸出的 match、match_index 資料為有效的輸出，反之，當 valid 為 Low，表示 match、match_index 為無效的輸出。

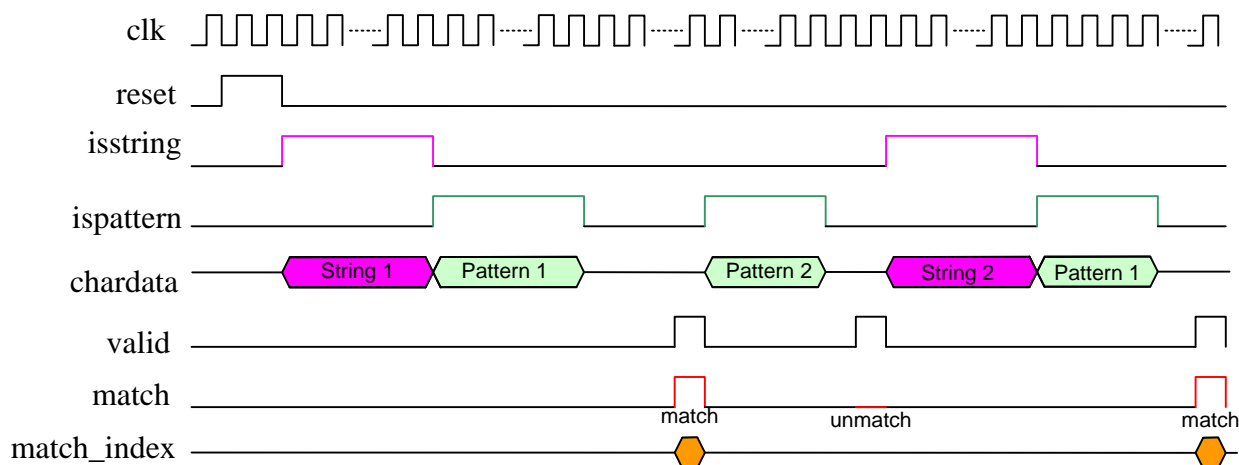
2.3 系統描述

2.3.1 String 及 Pattern 輸入順序

String 及 Pattern 輸入順序採一個 String 搭配數個 Pattern 方式；String 及 Pattern 資料都從 chardata 輸入，一次輸入 1 個 ASCII 字元，當 isstring 為 high 時表示此時 chardata 輸入為 String 資料，當 ispattern 為 high 時表示此時 chardata 輸入為 Pattern 資料；

String 和 Pattern 的長度非固定，但有上限，String 最長 32 字元，Pattern 最長 8 字元。

每完成輸入一個 Pattern 後，測試端便會等待 SME 輸出比對結果，取得結果後立刻再輸入下一個 String 或 Pattern，請見圖三。



圖三、String 及 Pattern 輸入順序

2.3.2 SME 的輸出

當 ispattern 由 high 變為 low 時，表示該 Pattern 輸入完成，SME 可開始比對字串。比對完成後，請將 valid 訊號拉為 High，並在同一個 cycle 內，輸出 match 與 match_index 的比對結果。

若比對結果為不成功(match == 0)，則 match_index 不被參考，可為任意值。

2.3.3 Pattern 特殊符號說明

String 及 Pattern 每一字元以 ASCII 編碼，ASCII 編碼表請見附錄三；Pattern 中可能包含以下四種特殊符號，特殊符號說明如下：

ASCII 字碼	符號	特殊符號說明
5E	^	比對 word 開頭
24	\$	比對 word 結尾
2E	.	比對任意單一字元

註：這三個特殊符號不會在 String 中出現。

上表”word”指的是連續非空白的字元，比如說 String 內容是”**This is a pencil**”，此字串內共含 4 個”word”，分別是”**This**”、”**is**”、”**a**”、”**pencil**”；

則以下 Pattern 皆可比對成功：

Pattern	match 字串	說明
^This	This is a pencil	^ 表示 T 開頭的 word
his\$	This is a pencil	\$ 表示 s 結尾的 word
^is\$	This is a pencil	^ 表示 i 開頭的 word, \$ 表示 s 結尾的 word
^a\$	This is a pencil	這 pattern 表示 word 裏只有一個 a 字元
^a pencil\$	This is a pencil	這 pattern 有兩個 word，最前面的 word 以 a 開頭，最後面的 word 以 l 結尾

且以下 Pattern 皆比對不成功：

Pattern	match 字串	說明
^his	This is a pencil	找不到 h 開頭的 word
pen\$	This is a pencil	找不到 n 結尾的 word
hi\$	This is a pencil	找不到 i 結尾的 word

註：為簡化狀況，**^**特殊符號只會出現在 Pattern 的最前面，**\$**特殊符號只會出現在 Pattern 最後面，且這兩個特殊符號不會和空白相連。

點號.可代表任意單一字元，以上面 String 例子，以下 Pattern 可比對成功：

Pattern	match 字串	說明
h.s	This is a pencil	. 代表 i 字元
p.n.il	This is a pencil	兩個.分別代表 e 和 c 字元
is.a.pe..il	This is a pencil	四個.分別代表 2 個空白和 nc 兩字元

註：一個 Pattern 可能有多個點號.特殊符號

2.3.4 比對成功位置(match_index)說明

String 及 Pattern 資料都從 chardata 輸入，一次輸入 1 個 ASCII 字元，皆由第 0 字元開始依序輸入，如底下 String，是先輸入 **T** 字元，最後輸入 **l** 字元

String 內容：This is a pencil

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
字元	T	h	i	s		i	s		a		p	e	n	c	i	l

當比對成功時，match_index 輸出 Pattern 在 String 中 match 的第一個位置，

若比對結果為 unmatched，則 match_index 不被參考，可為任意值。請見底下例子：

編號	Pattern	是否 match	match 部位(螢光標記部分)	match_index
1	pencil	match	This is a pencil	10
2	is	match	This is a pencil	2
3	^is	match	This is a pencil	5
4	h.s	match	This is a pencil	1
5	_pen	match	This is a pencil	9
6	pen\$	unmatch	This is a pencil	--
7	^an	unmatch	This is a pencil	--

註：編號 2 Pattern 在 String 中有多種 match 可能，取 match_index 最小的結果。

註：編號 5 Pattern 最前方有一空白字元，因此 match_index 為第 9 字元。

3. 評分標準

評分方式會依設計完成程度，分成 A、B、C 三種等級，排名順序為 A>B>C。評分項目有兩個，分別為總模擬週期數(cycle)及分數(score)，**模擬結束時會自動顯示這兩個數值。本題不限制 clock 週期時間，參賽者可自行調整 clock 週期時間。**

分數(score)公式為：

若該 pattern 正確結果為 match，SME 輸出 match 且 match_index 正確，加 3 分

若該 pattern 正確結果為 match，SME 輸出 match 但 match_index 錯誤，加 1 分

若該 pattern 正確結果為 match，SME 輸出 unmatched，不計分

若該 pattern 正確結果為 unmatched，SME 輸出 unmatched，加 1 分

若該 pattern 正確結果為 unmatched，SME 輸出 match，不計分

設計完成程度分二種等級，如下：

✧ **等級 A：** 等級 A 條件：

a、 功能正確，RTL 模擬與正確解答比對完全正確，**滿分 score 為 100。**

b、 完成 Synthesis，且 Gate-Level 模擬結果正確。

等級 A 之評分方法：

依繳交時間由早到晚排序。

✧ **等級 B：** 等級 B 條件：

a、 RTL 模擬與正確解答比對尚有部分錯誤

b、 或是 RTL 雖完全正確，但 Gate-Level 模擬尚有部分錯誤

等級 B 之評分方法：

以 RTL 模擬分數(score)排名，score 越高者，同級名次越好。

若多組 score 相同，依總模擬週期數(cycle)排序，cycle 數越少者，名次越佳。

就算沒有把全部功能完成，也請盡可能將最後結果上傳，有上傳才有晉級機會。

附錄

附錄一為 ASCII 字碼表；附錄二為設計檔案說明；附錄三為模擬訊息說明說明；附錄四為評分用檔案，亦即參賽者必須繳交的檔案資料；附錄五則為設計檔案壓縮整理步驟說明；

附錄一: ASCII 字碼表

ASCII 碼		字元
十進位	十六進位	
32	20	
33	21	!
34	22	"
35	23	#
36	24	\$
37	25	%
38	26	&
39	27	'
40	28	(
41	29)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7

ASCII 碼		字元
十進位	十六進位	
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O

ASCII 碼		字元
十進位	十六進位	
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D]
94	5E	^
95	5F	_
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g

ASCII 碼		字元
十進位	十六進位	
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	
125	7D	}
126	7E	~

附錄二 設計檔案說明

1. 下表為主辦單位所提供各參賽者的設計檔

表 3、設計檔案說明

檔名	說明
SME.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告。
tb.sv	Test Bench 檔案。
tb_term.sv	Test Bench 檔案，和 tb.sv 相同，但在 terminal 下可用顏色標示結果，方便觀看。
Etestdata.txt	Test Pattern 資料
.synopsys_dc.setup synopsys_dc.setup	使用 Design Compiler 做合成之初始化設定檔。參賽者請依 Library 實際擺放位置，自行修改 Search Path 的設定。注意：合成時請使用 worst case library。
SME.sdc	Design Compiler 作合成之 Constraint 檔案。參賽者可依需求自行修改 cycle 的設定，其餘環境參數請勿更改。
report.000	report 檔格式，見附錄 4。
dc_syn.tcl	dc 合成參考指令
ncvlog.f	ncverilog 模擬參考參數檔案。
vcs.cmd	vcs 模擬參考指令

請使用 **SME.v**，進行 SME 電路之設計。其模組名稱、輸出/入埠宣告如下所示：

```
module SME ( clk, reset, chardata, isstring, ispattern, valid, match, match_index);  
    input          clk;  
    input          reset;  
    input [7:0]    chardata;  
    input          isstring;  
    input          ispattern;  
    output         valid;  
    output         match;  
    output [4:0]   match_index;  
endmodule
```

2. 本題所提供的 Test Bench 檔案，有多增加幾行特別用途的敘述如下：

```
`define End_CYCLE 1000000  
`define SDFFILE    "./SME_syn.sdf"  
`ifdef SDF  
    initial $sdf_annotate(`SDFFILE, u_SME);  
`endif
```

註：

1. `tb.sv` 及 `tb_term.sv` 都是 testbench，兩者皆可使用，若在 terminal 下模擬建議使用 `tb_term.sv`，會以顏色標示錯誤位置，但若不是在 terminal 下模擬，這些顏色標記可能會變成亂碼，此時請改用 `tb.sv`。
2. Testbench(`tb.sv/tb_term.sv`)以 system verilog 格式撰寫，使用 ncverilog 模擬時請加入 `-sv` 參數；使用 vcs 模擬時請加入 `-sverilog` 參數。
3. End_CYCLE 預設 100 萬個 Cycles，其目的可以防止參賽者因電路有錯，模擬陷入無窮回圈之境，參賽者可視需要請自行加大此 Cycle 數。
4. SDF 檔案，請自行修改 SDF 實際檔名及路徑後再模擬。
5. 在 Test Bench 中，主辦單位提供 `ifdef SDF 的描述，其目的是讓本 Test Bench 可以作為 RTL 模擬與合成後模擬皆可使用。注意：當參賽者在合成後模擬，請務必多加一個參數”`+define+SDF`”，方可順利模擬。

例如：

當合成後，使用 NC-Verilog 模擬，在 UNIX terminal 下執行下面指令

```
➤ ncverilog -sv tb_term.sv SME_syn.v \  
+ncmaxdelays +define+SDF +access+r
```

當合成後，使用 VCS 模擬，在 UNIX terminal 下執行下面指令

```
➤ vcs -R -full64 -sverilog tb_term.sv SME_syn.v \  
+maxdelays +define+SDF +access+r +vcs+fsdbon
```

6. 請盡可能直接在 linux 環境將設計檔解開，避免在 window 環境解壓縮後才 ftp 傳至 linux 環境，以免因 ftp 在兩系統間置換換行符號不正確造成 testbench 無法模擬的問題。

附錄三 模擬訊息說明

在模擬過程，terminal 會列出目前 String 內容及 Pattern 內容，當 SME 送出結果時，則會分別顯示(1)在第幾 cycle 接收到結果、(2)預期正確內容、(3)接收到的內容、(4)比較結果，如下圖

```
== String 1  "abcdijk lmnop rstuv"
-- Pattern 1  "1234"
      cycle 24, expect(0,00) , get(1,00) >> Fail
-- Pattern 2  "abcd"
      cycle 31, expect(1,00) , get(1,00) >> Pass
-- Pattern 3  "dijk"
      cycle 3d, expect(1,03) , get(1,02) >> Wrong index
```

上面: **cycle 3d, expect(1,03) , get(1,02) >> Wrong index**

cycle 3d 表示在 reset 後第 3d (16 進位)個 cycle 接收到結果，

expect(1,03) 第一個數字(1)表示 golden 結果是 match，第二個數字(03)表示 golden 的 match_index

get(1,02) 第一個數字(1)表示得到 SME 回應是 match，第二個數字(02)是回應的 match_index

最後的 **Wrong index**，表示 golden 和回應都是 match，但回應的 match_index 不對，

如果這裏寫的是 **Fail**，表示 match/unmatch 結果回應錯誤(該 match 卻回應 unmatch，或是該 unmatch 卻回應 match)。

模擬結束後，會顯示總模擬週期數(cycle)及分數(score)，如下圖，此即為評分依據

```
-----
-- Simulation finish      --
-- cycle =473 , Score =31 --
-----
```

附錄四 評分用檔案

評分所須檔案可以下幾個部份：(1)RTL design，即各參賽隊伍對該次競賽設計的 RTL code，若設計採模組化而有多個設計檔，請務必將合成所要用的各 module 檔放進來，以免評審進行評分時，無法進行模擬；(2)Gate-Level design，即由合成軟體所產生的 gate-level netlist，以及對應的 SDF 檔；(3)report file，參賽隊伍必須依照自己的設計內容，撰寫 report.000 檔，以方便主辦單位進行評分，report.000 的格式如下圖所示。(report 檔以後三碼序號表示版本，若繳交檔案更新版本，則新版的 report 檔的檔名為 report.001，依此類推)

表 4

RTL category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	N/A	Design Report Form
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
Gate-Level category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
Pre-layout Gate-level Simulation	*_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	*_syn.sdf	Pre-layout gate-level sdf

report 檔

FTP account: B20xxx, FTP 帳號

Total simulation cycle (cycle) : 9000, 總模擬週期數,見附錄 3

Simulation score (score) : 100, 模擬完得到的分數,見附錄 3

--- RTL category---

HDL simulator : ncverilog/vcs, 使用之 HDL 模擬器名稱

RTL filename : SME.v, RTL 檔案名稱以及使用到的子模組檔案...

--- Pre-layout gate-level ---

gate_level filename: SME_syn.v, gate-level 檔案名稱

gate-level sdf filename: SME_syn.sdf, sdf 檔案名稱

-----(annotation)-----

(其餘注意事項依各參賽隊伍的需求填寫)

附錄五 檔案壓縮整理步驟

當所有的文件準備齊全如表 4 所列，均需要提交至 TSRI。請按照以下的步驟指令，提交相關設計檔案，將所有檔案複製至同一個資料夾下壓縮，步驟如下：

1. 建立一個 result_xxx 資料夾。其中“xxx”表示繳交版本。例如“000”表示為第一次上傳；“001”表示為第二度上傳；002 表示為第三度上傳，以此類推...。
 > **mkdir result_000**
2. 將附錄四要求的檔案複製到 result_xxx 這個目錄。例如：
 > **cp SME.v result_000**
 > **cp SME_syn.v result_000**
 > **cp report.000 result_000**

3. 執行 tar 指令將 result_xxx 資料夾包裝起來，tar 的指令範例如下：
 > **tar cvf result_000.tar result_000**
 執行完後應該會得到 result_000.tar 的檔案
4. 使用 ftp 將 result_xxx.tar 上傳至 TSRI 提供的 ftp server，評審將以最後上傳的設計檔編號進行評分作業。

請注意!!上傳之設計檔僅可使用 tar 或 zip 壓縮格式,使用 rar 或其他格式者一律不予計分。
上傳之 FTP 需切換為二進制模式(binary mode)，且傳輸埠均設為 21(port:21)。

ftp 的帳號和密碼在賽前已用 email 寄給各參賽者。若有任何問題，請聯絡 TSRI

FTP site1 (新竹半導體中心)：iccftp.tsri.org.tw (140.126.24.18)

FTP site2 (南區半導體中心)：iccftp2.tsri.org.tw(140.110.117.9)

EdaCloud內請見開啟terminal時訊息

5. 若需要繳交更新版本，請重覆以上步驟，並記得修改 tar 檔的版本編號，因為您無法修改或刪除或覆蓋之前上傳的資料。