CAD/VLSI Circuit Design 期末報告

7109064300　　范文軒　　電機丁組(數位)

(一)、動機

搜尋演算法是近些年科技中很重要的基石，而字串比對演算法(string matching algorithm)也是其中之一。像這樣的比對方式會被使用至 DNA 序列的比對、資料庫的查詢(query)等。而如今資料庫系統中的資料量越來越大。為了某個特徵比對，可能會使用到多次的比對查詢。因此加速這個過程，就顯得重要的。若使用暴力破解的方式，必須一個字元依次比較，而因為沒有辦法加速，時間複雜度會為 O(m*n) (in worst case)。如今不同新的演算法也不斷推出，像是近年最多人使用的演算法之一：KMP 演算法[1]，優化了最原始的方式，使得無意義的比較可以跳過，帶來更多速度上的優勢，而時間複雜度減至 O(m+n) (in worst case)。而本次構想希望參考可平行化運算 KMP 演算法[2]，並且利用硬體亦可平行化的優勢，實做出有平行化的 KMP 演算法的 SME(string matching engine)。

(二)、主要概念與預想[2]

a、　構想概要

(1). KMP Algorithm[1]

KMP 演算法相較於暴力破解，多增加了前處理(preprocessing)的部分，在此演算法即所謂的 Failure Function。增加了一個關於 Pattern 的表格去紀錄若判別失敗該跳去哪一個位置。

(2). 可平行化 KMP Algorithm[2]

而在參考[2]中，使用了平行化技巧，將比對時分切成 4 個特殊大小的字串同時比對，進而減少時間的損耗。

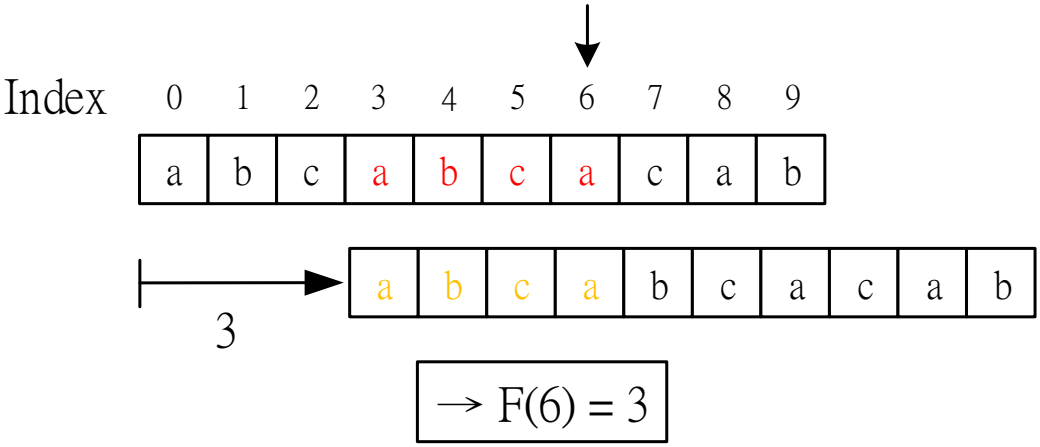
參考演算法概念後，在設計中設計以下分割之作業：Idle、String 與 Pattern 輸入、Failure Function 建立、比對環節以及輸出結果。並打算在比對環節中設計 4 的 PE(Process Engine)同時運作於比對環節中進而實驗平行化。

b、 KMP 演算法

(1). Failure Function

Failure Function 是用來計算 Pattern 比對失敗時可以跳去哪一個 Pattern 位置繼續比較。算法主要概念為「前綴(suffix)要位移幾位可以和目前的字串相同」。

（舉例）



圖一、Failure Function 概念舉例

但通常如果用此定義算通常會計算非常久，所以後來大多都使用 Dynamic Programming 的方式加速 Failure Function 的算法。

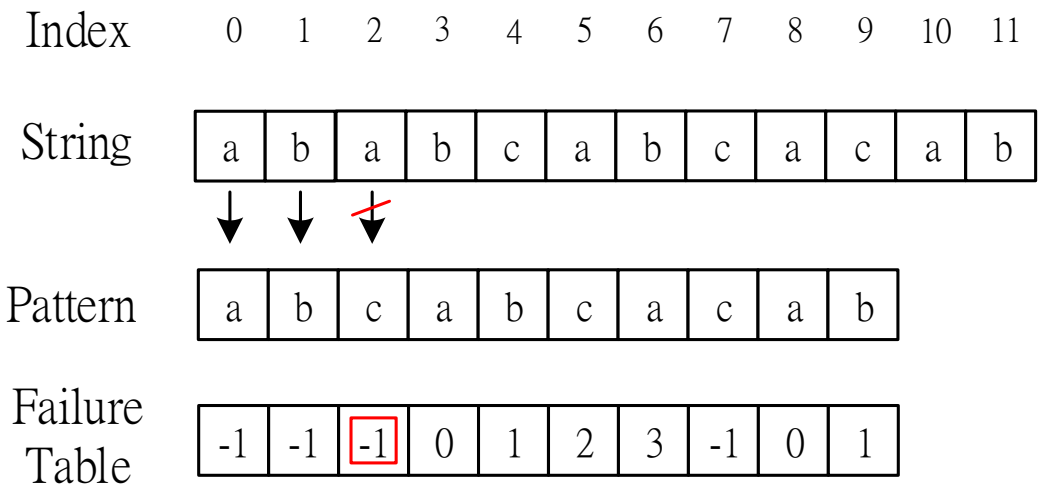
$$f(j) = \begin{cases} -1 & \text{if } j = 0 \\ f^m(j-1)+1 & \text{where } m \text{ is the least integer} \\ & k \text{ for which } p_{f^k(j-1)+1} = p_j \\ -1 & \text{if there is no } k \text{ satisfying the above} \end{cases}$$

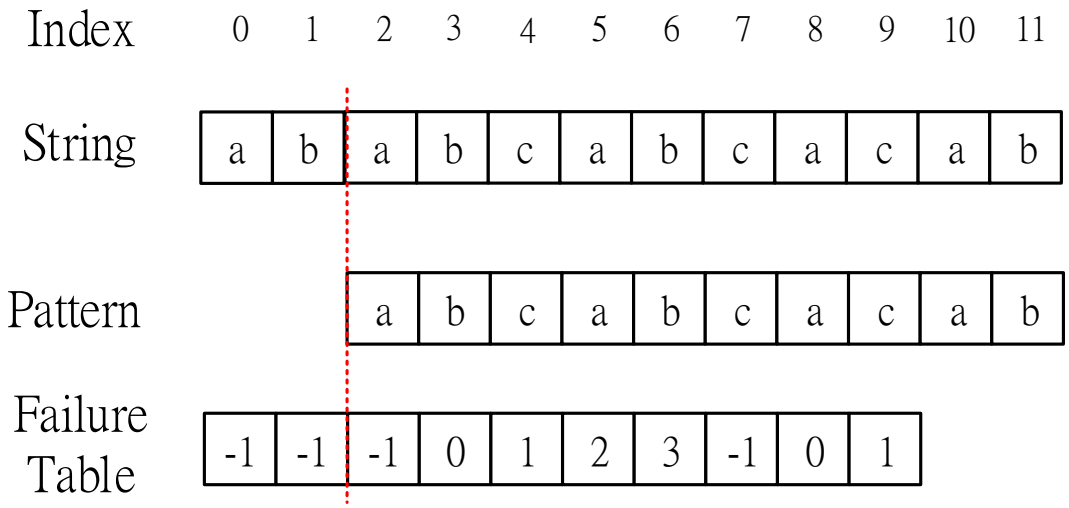
$$f^1(j) = f(j) \text{ and } f^m(j) = f(f^{m-1}(j))$$

圖二、Failure Function with Dynamic Programming

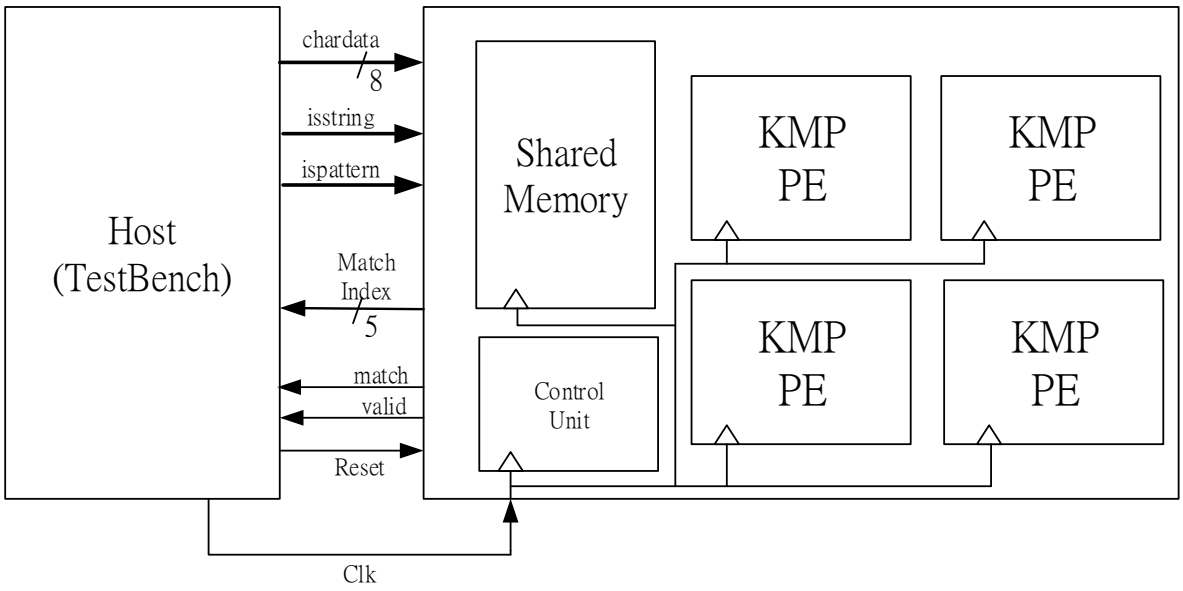
(2). KMP Matching Operation

由上述的 Failure Function 完成計算後，生成的 Failure Table 可以紀錄比對失敗時 Pattern 該從 Failure Function 的值加一的位置，與 String 比對失敗的地方繼續比較下去，依此類推。





(三)、主要架構



圖一、SME 系統架構圖

c、 SME 概要

此次設計的 SME 使用多模組的方式建構 (可參考上圖)，主要有 Shared Memory、Control Unit 以及 KMP Compare PE(Process Engine)。而 Shared Memory 內部為了要計算 KMP 演算法中的前處理部分，多設計了專門計算 Failure Function 的模組 — Failure Function Calculator 一併放在 Shared Memory 內部，來使用特殊方式將 Shared Memory 實作出來(將在三、c 部分說明)。

資料將從外部傳入 Shared Memory，Shared Memory 將接收並且儲存在內部。等待資料全部載入完成，將會啟動 Failure Function Calculator 計算該 Pattern 的 Failure Function 後一併將輸入的資料分享到每一個 PE 中，而 Control Unit 將會把需要做的長度告訴各個 PE，而各個 PE 會依照 KMP 演算法的方式計算出各分段結果並傳給 Control Unit，最後，Control Unit 會將結果輸出。

b、Shared Memory

因字串的長度需要相對應大小的暫存器，因此利用 Shared Memory 來增大暫存器的使用率，同時間也減少不必要 PE 裡的暫存器。

c、Failure Function Calculator

Failure Function 部分會將 Shared Memory 的 Pattern 資料傳入，進而計算。而此次設計計算 Failure Function 的方式是利用，Dynamic Programming 的演算法實作，利用上一筆的結果來繼續計算下一筆的，使得比起重新計算更加有效率。

d、KMP Compare Process Engine(PE)

利用 Sharing Memory 的輸出得到資料、Failure Function 進而比對，與先前的 KMP 演算法無異，唯有不同之處為可以彈性調整字串比較的長短來做比較，也是為了做到平行化的前提。

e、資料與 Pattern

圖一為此次設計的架構，左方的 Host 為 Testbench 的部分，會將需要比對的資料傳送至右方的設計中（SME)，為了模擬 DNA 的比對情況，將會傳送內容為 DNA 的序列（ A 、 T 、 C 、 G ）的字串序列。

e、彈性化

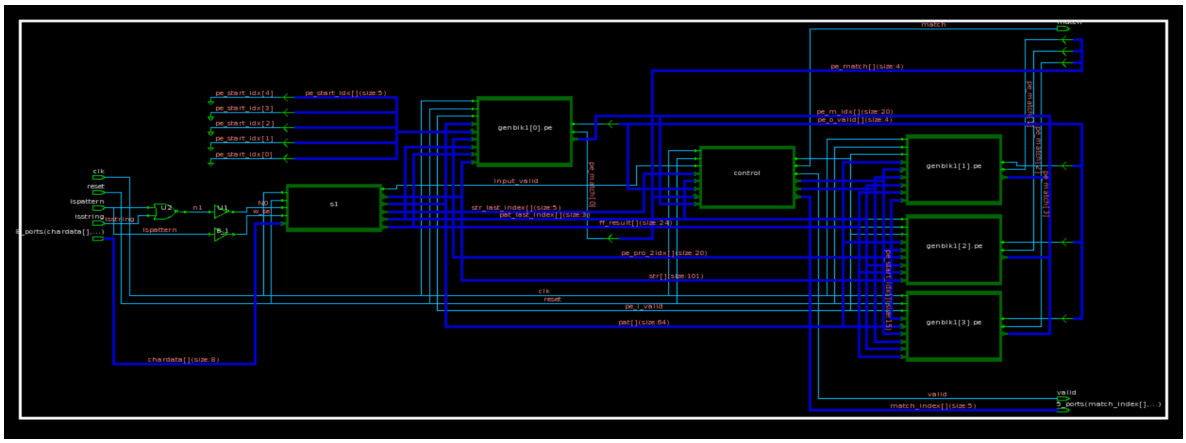
因為為了使此 SME 更加有彈性，設置了多個參數可供調整，可自行決定 String 與 Pattern 的最大長度，同時也可以自由擴大 PE 數目。

(四)、合成與結果

```
== String 6  "ACACGGAAGCTTTGTAACT"
-- Pattern 1  "TTCAC"
     cycle 7c2, expect(0,--) , get(0,--) >> Pass
-- Pattern 2  "ATGCACA"
     cycle 7eb, expect(0,--) , get(0,--) >> Pass
-- Pattern 3  "ACG"
     cycle 80f, expect(1,02) , get(1,02) >> Pass
-- Pattern 4  "G"
     cycle 82c, expect(1,04) , get(1,04) >> Pass
-- Pattern 5  "CG"
     cycle 84c, expect(1,03) , get(1,03) >> Pass
-- Pattern 6  "GG"
     cycle 86c, expect(1,04) , get(1,04) >> Pass
-- Pattern 7  "C"
     cycle 888, expect(1,01) , get(1,01) >> Pass
-- Pattern 8  "TCCCAT"
     cycle 8ae, expect(0,--) , get(0,--) >> Pass
-- Pattern 9  "TTCATG"
     cycle 8d4, expect(0,--) , get(0,--) >> Pass
-- Pattern a  "TGT"
     cycle 8f8, expect(1,0c) , get(1,0c) >> Pass
---------------------------------
- Simulation finish, ALL PASS  --
- cycle =2297 , Score =122       --
---------------------------------
finish called at time : 22970 ns : File "C:/Users/Alan/Desktop/git/SME_parallel/test_bench/sme_tb.sv" Line 201
```

圖二、Simulation 結果(Vivado)

(Max String Length : 32, Max Pattern Length : 8, no.PE : 4)



圖三、合成結果(Design Vision)

(Process : TSMC 90nm ， Clock Constrain : 200Mhz, no.PE : 4)

```
" -- Pattern 2  "ATGCACA
      cycle 7eb, expect(0,--) , get(0,--) >> Pass
" -- Pattern 3  "ACG
      cycle 80f, expect(1,02) , get(1,02) >> Pass
" -- Pattern 4  "G
      cycle 82c, expect(1,04) , get(1,04) >> Pass
" -- Pattern 5  "CG
      cycle 84c, expect(1,03) , get(1,03) >> Pass
" -- Pattern 6  "GG
      cycle 86c, expect(1,04) , get(1,04) >> Pass
" -- Pattern 7  "C
      cycle 888, expect(1,01) , get(1,01) >> Pass
" -- Pattern 8  "TCCCAT
      cycle 8ae, expect(0,--) , get(0,--) >> Pass
" -- Pattern 9  "TTCATG
      cycle 8d4, expect(0,--) , get(0,--) >> Pass
" -- Pattern a  "TGT
      cycle 8f8, expect(1,0c) , get(1,0c) >> Pass
-------------------------------------
-- Simulation finish, ALL PASS  --
-- cycle =2297 , Score =122       --
-------------------------------------
Simulation complete via $finish(1) at time 11485 NS + 0
./sme_tb.sv:203                    $finish;
```

圖四、Pre-Simulation(Ncverilog)

(Run Clock Rate : 200Mhz , Max String Length : 32)

(Max Pattern Length : 8, no.PE : 4)

```
****************************************
Report : area
Design : SME
Version: K-2015.06-SP1
Date   : Tue Dec 29 05:24:10 2020
****************************************

Library(s) Used:

    slow (File: /usr/cad/DesignKit/CBDK_TSMC90GUTM_Arm_f1.0/CIC/SynopsysDC/db/slow.db)

Number of ports:                         2034
Number of nets:                          5617
Number of cells:                         3551
Number of combinational cells:           3025
Number of sequential cells:               519
Number of macros/black boxes:               0
Number of buf/inv:                        855
Number of references:                       8

Combinational area:              26857.252930
Buf/Inv area:                     8329.607940
Noncombinational area:           10964.318250
Macro/Black Box area:                0.000000
Net Interconnect area:         2239778.279861

Total cell area:                 37821.571180
Total area:                    2277599.851042
1
```

圖五、Area Report(Design Vision)

```
Operating Conditions: slow   Library: slow
Wire Load Model Mode: top

Design        Wire Load Model            Library
------------------------------------------------
SME                   tsmc090_w150      slow


Global Operating Voltage = 0.9
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =   2.2494 mW   (66%)
  Net Switching Power  =   1.1377 mW   (34%)
                         ---------
Total Dynamic Power    =   3.3871 mW  (100%)

Cell Leakage Power     = 242.9314 uW

                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    ) Attrs
-----------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000              0.0000             0.0000  (   0.00%)
memory             0.0000           0.0000              0.0000             0.0000  (   0.00%)
black_box          0.0000           0.0000              0.0000             0.0000  (   0.00%)
clock_network      0.0000           0.0000              0.0000             0.0000  (   0.00%)
register           2.0995        9.0558e-02          5.4301e+07            2.2443  (  61.83%)
sequential     1.4887e-04        5.0023e-04          5.1239e+05        1.1615e-03  (   0.03%)
combinational      0.1498           1.0466          1.8812e+08            1.3845  (  38.14%)
-----------------------------------------------------------------------------------------------
Total              2.2494 mW        1.1377 mW       2.4293e+08 pW         3.6300 mW
1
```

圖六、Power Report(Design Vision)

```
Operating Conditions: slow   Library: slow
Wire Load Model Mode: top

  Startpoint: reset (input port clocked by clk)
  Endpoint: s1/dp1/o_fail_func_reg[9]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  SME                tsmc090_w150          slow

  Point                                                   Incr       Path
  ---------------------------------------------------------------------------
  clock clk (fall edge)                                   2.50       2.50
  clock network delay (ideal)                             0.50       3.00
  input external delay                                    0.00       3.00 f
  reset (in)                                              0.05       3.05 f
  s1/reset (shared_memory)                                0.00       3.05 f
  s1/U456/Y (BUFX14)                                      0.08       3.13 f
  s1/U482/Y (BUFX20)                                      0.09       3.22 f
  s1/U572/Y (CLKBUFX40)                                   0.14       3.36 f
  s1/dp1/reset (DP_FailFunc)                              0.00       3.36 f
  s1/dp1/U205/Y (CLKBUFX40)                               0.15       3.51 f
  s1/dp1/U143/Y (BUFX20)                                  0.10       3.61 f
  s1/dp1/U108/Y (CLKINVX40)                               0.09       3.70 r
  s1/dp1/U114/Y (AND3X8)                                  0.22       3.92 r
  s1/dp1/U74/Y (NAND2X6)                                  0.17       4.09 f
  s1/dp1/U76/Y (INVX20)                                   0.17       4.26 r
  s1/dp1/U35/Y (NAND2X8)                                  0.22       4.48 f
  s1/dp1/U91/Y (AOI2BB1X4)                                0.24       4.73 f
  s1/dp1/U4/Y (BUFX12)                                    0.25       4.97 f
  s1/dp1/U21/Y (OA12BB2X4)                                0.31       5.28 r
  s1/dp1/o_fail_func_reg[9]/D (DFFHQX8)                   0.00       5.28 r
  data arrival time                                                  5.28

  clock clk (rise edge)                                   5.00       5.00
  clock network delay (ideal)                             0.50       5.50
  clock uncertainty                                      -0.10       5.40
  s1/dp1/o_fail_func_reg[9]/CK (DFFHQX8)                  0.00       5.40 r
  library setup time                                     -0.12       5.28
  data required time                                                 5.28
  ---------------------------------------------------------------------------
  data required time                                                 5.28
  data arrival time                                                 -5.28
  ---------------------------------------------------------------------------
  slack (MET)                                                        0.00
```

圖七、Timing Report(Design Vision)

(四)、規格

Hardware	
Synthesis Process	TSMC 90 nm
CLK Cycle	200Mhz
Throughput	5.2 M Match /Sec

[備註]：此 Throughput 之情況為原始設計目標最大 String 長度 32Bytes，最大 Pattern 長度 8 Bytes，SME 中共有 4 個比對 PE，於 200Mhz 下筆對了 60 筆得到 2297 Cycle(即 11485 ns)，由此推估 Throughput 可達 5.2M Match/Sec。

(五)、參考

[1] D. E. Knuth, J. H. Morris and V. R. Pratt, "Fast Pattern Matching in Strings," *SIAM J. COMPUT*, Vol.6, No.2, June *1977*.

[2] U. S. Alzoabi, N. M. Alosaimi, A. S. Bedaiwi and A. M. Alabdullatif, "Parallelization of KMP String Matching Algorithm," *2013 World Congress on Computer and Information Technology (WCCIT)*, pp. 1-3, 2013.

[備註]所有的 Code 皆放在我個人的 github 上面：

https://github.com/vanwanTaiwan/SME_parallel