

# Computer Organization (ET3502701)

## Homework I

Assign Date: 2020/11/2

B10702239 吳威廷

1. 請將程式碼之費波那契(Fibonacci)數列程式以一個完整的組語程式實現(檔名 fibo.s)，假設參數 n 設為 7，整數輸入值(n 值)存取於暫存器 a0，答案(數列於 n=7 時數值)存取於暫存器 a1，將實作完的程式於 spike 上執行並在 debug 模式使用 reg 0 指令產生暫存器報告。(此題須附上程式的組合語言 fibo.s 檔、spike 執行結果的截圖、解說該組合語言程式如何運作)(40%)

```
10      addi a1, zero, 0    //set fib = 0
11      addi t2, zero, 2    //set i = 2
12      addi t0, zero, 0    //set t0 = 0
13      addi t1, zero, 1    //set t1 = 1
14      addi a0, zero, 7    //set n = 7
15      addi t3, zero, 1    //t3 is a flag to judge if n < i
16      BEQ a0, zero, EXIT  //judge if n == 0
17 JUMP: add a1, t0, t1      //if t3 == 0 then it means n < i
18      add t0, t1, zero     //set t0 = t1
19      add t1, a1, zero     //set t1 = a1
20      addi t2, t2, 1       //i = i + 1
21      SLT t3, a0, t2       //judge if n < i
22      BEQ t3, zero, JUMP  //judge if t3 == 0 it means n < i then jump
23 EXIT:                    //if n == 0 then it will jump to this line
```

### 程式碼

a1, t2, t0, t1, a0, t3 對應到 C 語言的程式碼分別是 fib, i, t0, t1, n, t3 則是一個去判斷迴圈要不要再執行一次的 flag。

在程式的一開始先做各個佔存器初始值的設定，接者在第 16 行去判斷我的輸入，也就是 n 的數值是否為 0，如果 n 是 0 的話那就直接 branch 到第 23 行結束程式的執行，如果 n 的數值不是 0 的話那就執行對應到 C 語言 for loop 裡面的動作，(對應到組語是第 17-19 行)，之後把 i 的數值做加一的動作接著繼續判斷。因為要使用我們這個 single cycle cpu 來跑然後有些指令沒有做的關係只能用有做的指令來執行我們的組語所以在這個判斷式我是用  $n < i$  的方式來判斷，如果  $n < i$  不成立的時候 t3 的數值會變成 0 (因為  $n = 7$ ) 在第 22 行判斷到  $t3 = 0$  時就會再跑一次迴圈直到  $n < i$  這個判斷成立的時候 t3 會等於 1 接著就會跳出迴圈代表程式結束，還有因為如果初始值  $i = 1$  的話迴圈會多跑一次，所以把 i 設成從 2 開始。

執行結果示意圖：

```
: reg 00
zero: 0x0000000000000000 ra : 0x0000000000000000 sp : 0x0000000000000000 gp : 0x0000000000000000
tp : 0x0000000000000000 t0 : 0x0000000000000001 t1 : 0xffffffff80000050 t2 : 0x0000000000000007
s0 : 0x0000000000000000 s1 : 0x0000000000000000 a0 : 0x0000000000000007 a1 : 0x000000000000000d
a2 : 0x0000000000000000 a3 : 0x0000000000000000 a4 : 0x0000000000000000 a5 : 0x0000000000000000
a6 : 0x0000000000000000 a7 : 0x0000000000000000 s2 : 0x0000000000000000 s3 : 0x0000000000000000
s4 : 0x0000000000000000 s5 : 0x0000000000000000 s6 : 0x0000000000000000 s7 : 0x0000000000000000
s8 : 0x0000000000000000 s9 : 0x0000000000000000 s10: 0x0000000000000000 s11: 0x0000000000000000
t3 : 0x0000000000000000 t4 : 0x0000000000000001 t5 : 0x0000000000000000 t6 : 0x0000000000000000
```

- 附上 ALU、Single-Cycle CPU 的程式碼(30%)、CPU 的 Synthesis 電路報告圖以及檔案(\*.area, \*.timing 檔案)(10%)、在 CPU 使用 TB1 測試下 RTL Simulation 波型圖(5%)，以及結果 Congratulation. You pass TA's pattern 的截圖(5%)

Timing 圖：

Point	Incr	Path
clock clk_i (rise edge)	0.00	0.00
clock network delay (ideal)	0.50	0.50
PC/count_0_reg[2]/Q (DFFRX1)	0.00	0.50 r
PC/count_0_reg[2]/Q (DFFRX1)	0.73	1.23 f
PC/count_0[2] (ProgramCounter)	0.00	1.23 f
adder_PC_plusfour/src1_i[2] (adder_2)	0.00	1.23 f
adder_PC_plusfour/add_17/A[2] (adder_2_DW01_add_0_DW01_add_4)	0.00	1.23 f
adder_PC_plusfour/add_17/U1_2/CO (ADDFHX4)	0.39	1.62 f
adder_PC_plusfour/add_17/U5/Y (NAND2X4)	0.07	1.69 r
adder_PC_plusfour/add_17/U1/Y (NAND3X2)	0.13	1.82 f
adder_PC_plusfour/add_17/U1_4/CO (ADDFHX4)	0.22	2.03 f
adder_PC_plusfour/add_17/U1_5/CO (ADDFHX2)	0.24	2.27 f
adder_PC_plusfour/add_17/U1_6/CO (ADDFHX4)	0.21	2.48 f
adder_PC_plusfour/add_17/U1_7/CO (ADDFHX2)	0.24	2.71 f
adder_PC_plusfour/add_17/U1_8/CO (ADDFHX4)	0.21	2.92 f
adder_PC_plusfour/add_17/U1_9/CO (ADDFHX2)	0.24	3.16 f
adder_PC_plusfour/add_17/U1_10/CO (ADDFHX4)	0.21	3.36 f
adder_PC_plusfour/add_17/U1_11/CO (ADDFHX2)	0.24	3.60 f
adder_PC_plusfour/add_17/U1_12/CO (ADDFHX4)	0.21	3.81 f
adder_PC_plusfour/add_17/U1_13/CO (ADDFHX2)	0.24	4.04 f
adder_PC_plusfour/add_17/U1_14/CO (ADDFHX4)	0.20	4.25 f
adder_PC_plusfour/add_17/U1_15/CO (ADDFHX2)	0.36	4.61 f
adder_PC_plusfour/add_17/U1_16/CO (ADDFHX4)	0.23	4.84 f
adder_PC_plusfour/add_17/U1_17/CO (ADDFHX4)	0.23	5.07 f
adder_PC_plusfour/add_17/U2/Y (NAND2X1)	0.19	5.26 r
adder_PC_plusfour/add_17/U2/Y (NAND3X2)	0.15	5.42 f
adder_PC_plusfour/add_17/U1_19/CO (ADDFHX4)	0.22	5.64 f
adder_PC_plusfour/add_17/U1_20/CO (ADDFHX2)	0.30	5.94 f
adder_PC_plusfour/add_17/U5/Y (NAND2X4)	0.11	6.05 r
adder_PC_plusfour/add_17/U14/Y (NAND2X2)	0.22	6.27 r
adder_PC_plusfour/add_17/U4/Y (NAND2X5)	0.10	6.37 f
adder_PC_plusfour/add_17/U3/Y (NAND2X4)	0.08	6.46 r
adder_PC_plusfour/add_17/U6/Y (NAND2X1)	0.19	6.65 f
adder_PC_plusfour/add_17/U1_23/CO (ADDFHX4)	0.29	6.94 f
adder_PC_plusfour/add_17/U37/Y (NAND2X1)	0.18	7.12 r
adder_PC_plusfour/add_17/U11/Y (NAND3X1)	0.19	7.31 f
adder_PC_plusfour/add_17/U1_25/CO (ADDFHX2)	0.35	7.65 f
adder_PC_plusfour/add_17/U7/Y (NAND2X4)	0.12	7.77 r
adder_PC_plusfour/add_17/U26/Y (NAND3X2)	0.13	7.89 f

Area 圖：

```

*****
Report : area
Design : Simple_Single_CPU
Version: Q-2019.12
Date   : Wed Nov 11 00:13:44 2020
*****

Library(s) Used:

    slow (File: /cad/CBDK/CBDK_IC_Contest_v2.1/SynopsysDC/db/slow.db)

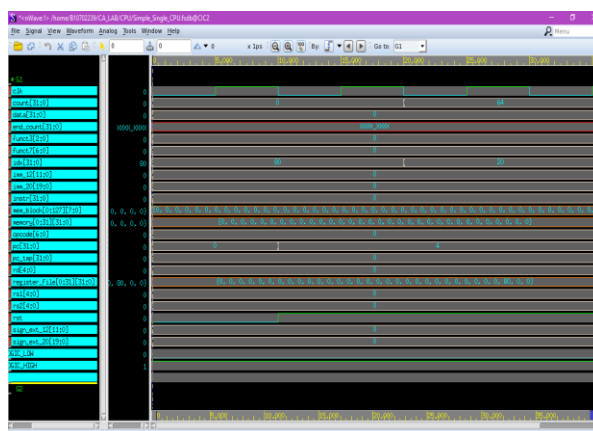
Number of ports:      1861
Number of nets:       16461
Number of cells:      14563
Number of combinational cells: 12386
Number of sequential cells: 2080
Number of macros/black boxes: 0
Number of buf/inv:    1523
Number of references: 17

Combinational area:      109787.833321
Buf/Inv area:            9125.222258
Noncombinational area:   60126.999500
Macro/Black Box area:    0.000000
Net Interconnect area:   2239404.597839

Total cell area:         169914.832821
Total area:              2409319.430660

```

使用 TB1 測試的 RTLsimulation 波形圖：



TA' pass 圖：

```

Congratulation. You pass TA's pattern

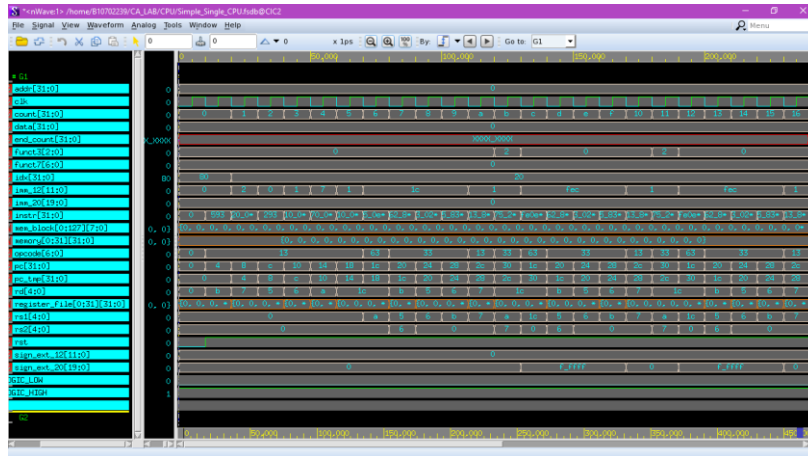
Register:
r0= 0, r1= 0, r2= 0, r3= 0,
r4= 0, r5= 0, r6= 0, r7= 0,
r8= 0, r9= 0, r10= 0, r11= 0,
r12= 0, r13= 0, r14= 0, r15= 0,
r16= 0, r17= 0, r18= 0, r19= 0,
r20= 0, r21= 0, r22= 0, r23= 0,
r24= 0, r25= 0, r26= 0, r27= 0,
r28= 0, r29= 128, r30= 0, r31= 0,

Memory:
m0= 0, m1= 0, m2= 0, m3= 0,
m4= 0, m5= 0, m6= 0, m7= 0,
m8= 0, m9= 0, m10= 0, m11= 0,
m12= 0, m13= 0, m14= 0, m15= 0,
m16= 0, m17= 0, m18= 0, m19= 0,
m20= 0, m21= 0, m22= 0, m23= 0,
m24= 0, m25= 0, m26= 0, m27= 0,
m28= 0, m29= 0, m30= 0, m31= 0,

```

- 將題1經 riscv-toolchain 轉出之 binary file 作為測資 TB2(檔名 fibo.bin) ,  
將 CPU 在 RTL Simulation 波型圖(5%) 與結果 Congratulation. You pass  
TA's pattern 的截圖附上(5%)

波形示意圖：



You pass TA's pattern 截圖：

```
FSDB Dumper for IUS, Release Verdi_Q-2020.03, Linux, 02/09/2020
(C) 1996 - 2020 by Synopsys, Inc.
*VerdiA : Create FSDB file 'Simple_Single_CPU.fsdb'
*VerdiA : Begin traversing the scopes, layer (0).
*VerdiA : End of traversing.
*VerdiA : Begin traversing the MDAs, layer (0).
*VerdiA : Enable +mda and +packedmda dumping.
*VerdiA : End of traversing the MDAs.

=====
Congratulation. You pass TA's pattern
=====
Register=====
r0=      0, r1=      0, r2=      0, r3=      0,
r4=      0, r5=      8, r6=     13, r7=      8,
r8=      0, r9=      0, r10=     7, r11=    13,
r12=     0, r13=     0, r14=     0, r15=     0,
r16=     0, r17=     0, r18=     0, r19=     0,
r20=     0, r21=     0, r22=     0, r23=     0,
r24=     0, r25=     0, r26=     0, r27=     0,
r28=     1, r29=    128, r30=     0, r31=     0,
Memory=====
m0=      0, m1=      0, m2=      0, m3=      0,
m4=      0, m5=      0, m6=      0, m7=      0,
m8=      0, m9=      0, m10=     0, m11=     0,
m12=     0, m13=     0, m14=     0, m15=     0,
m16=     0, m17=     0, m18=     0, m19=     0,
m20=     0, m21=     0, m22=     0, m23=     0,
m24=     0, m25=     0, m26=     0, m27=     0,
m28=     0, m29=     0, m30=     0, m31=     0,
Simulation stopped via $stop(1) at time 460 NS + 0
./Test_Bench.v:529      $stop;
```