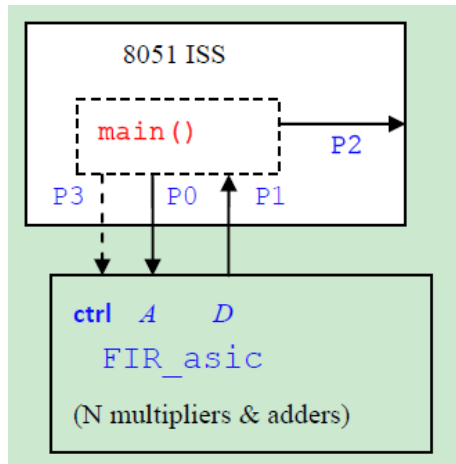


## Hardware/Software Codesign: SystemC Exercise Two

1. Use N multipliers and adders and the result is generated in one cycle.



各接腳訊號：

P0(A) : 8051 送出 random value

P1(D) : FIR\_asic 傳回計算好的 value

P3(B, C) : 為雙向的控制訊號，

(1)B 為 8051 送出的訊號

0 : 8051 尚未準備好 random value

1 : 8051 已經準備好 random value

(2)C 為 FIR\_asic 回傳的訊號

0 : FIR\_asic 尚未計算完畢

1 : FIR\_asic 已經計算完畢

為了讓實驗結果易於觀察，在 FIR 中我使用七個係數{1, 2, 3, 4, 3, 2, 1}，而 8051 的 P0(A)皆送出同一數值 1，所以我們可以預測在 reset 時所有 ports 為 0，T1 時 P0 送出數值 1，T2 時 P3(B)跳為 1 表示 8051 已經將 random 準備好，T3 時 P3(C)跳為 1 表示 FIR\_asic 已經將數值運算完畢，T4 時 P1(D)為運算的結果。

模擬結果：

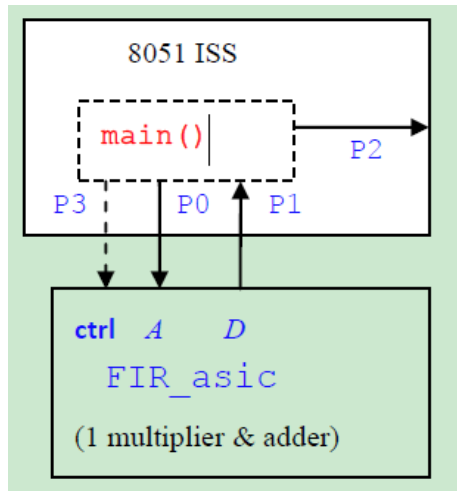
```

C:\Windows\system32\cmd.exe
Time:209, P0=0, P1=0, P2=0, P3=0
Time:210, P0=0, P1=0, P2=0, P3=0
Time:211, P0=1, P1=0, P2=0, P3=0
Time:212, P0=1, P1=1, P2=1, P3=1
Time:213, P0=1, P1=3, P2=1, P3=1
Time:214, P0=1, P1=6, P2=1, P3=1
Time:215, P0=1, P1=10, P2=1, P3=1
Time:216, P0=1, P1=13, P2=1, P3=1
Time:217, P0=1, P1=15, P2=1, P3=1
Time:218, P0=1, P1=16, P2=1, P3=1
Time:219, P0=1, P1=16, P2=1, P3=1
Time:220, P0=1, P1=16, P2=1, P3=1

```

此處的 P2 代表 P3(C)( FIR\_asic 回傳的訊號)，在 Time:212 可以觀察到所有的 ports 都為 1，這裡可能的原因為系統運算速度極快，時間的分割不夠細，所以 FIR\_asic 的數值很快就算好。

2. Use one multiplier and adder, we need N cycles to complete the FIR computation.



各接腳訊號：

P0(A) : 8051 送出 random value

P1(D) : FIR\_asic 傳回計算好的 value

P3(B, C) : 為雙向的控制訊號，

(1)B 為 8051 送出的訊號

0 : 8051 尚未準備好 random value

1 : 8051 已經準備好 random value

(2)C 為 FIR\_asic 回傳的訊號

0 : FIR\_asic 尚未計算完畢

1 : FIR\_asic 已經計算完畢

這題與第一題不同的地方在於，FIR\_asic 中的運算有加了 loop，每次 loop 只會執行一次加法與減法，符合題目要求。

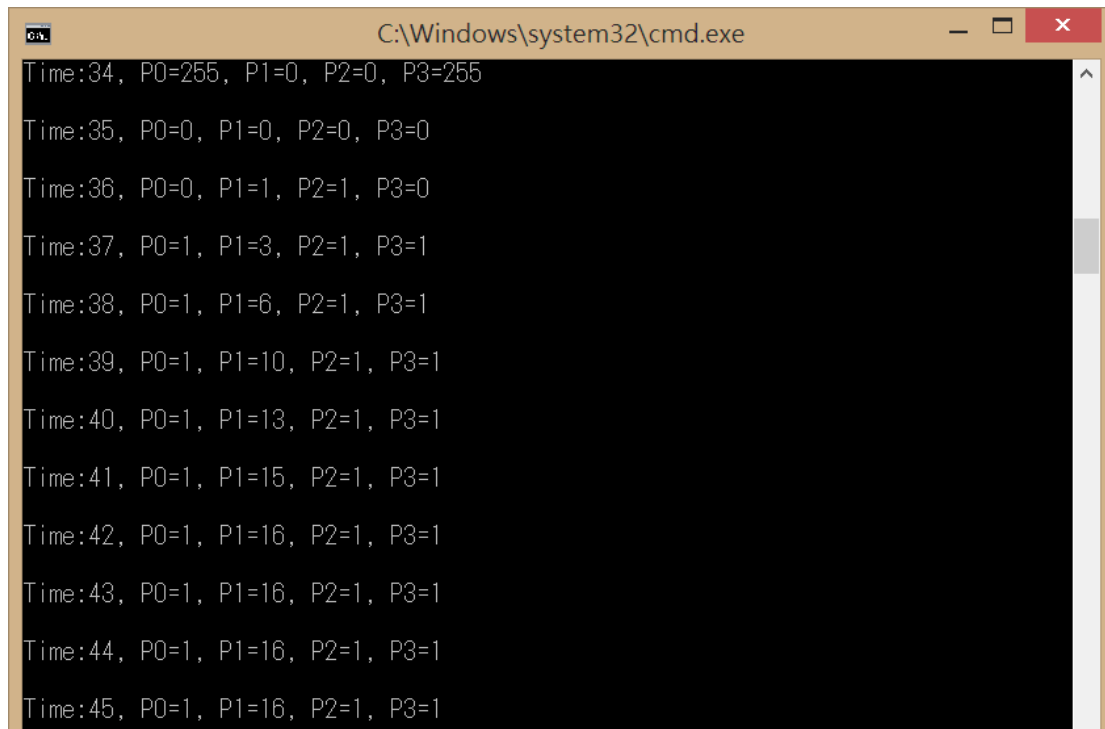
模擬結果：

```
C:\Windows\system32\cmd.exe
Time:209, P0=0, P1=0, P2=0, P3=0
Time:210, P0=0, P1=0, P2=0, P3=0
Time:211, P0=1, P1=0, P2=0, P3=0
Time:212, P0=1, P1=205, P2=1, P3=1
Time:213, P0=1, P1=3, P2=1, P3=1
Time:214, P0=1, P1=6, P2=1, P3=1
Time:215, P0=1, P1=10, P2=1, P3=1
Time:216, P0=1, P1=13, P2=1, P3=1
Time:217, P0=1, P1=15, P2=1, P3=1
Time:218, P0=1, P1=16, P2=1, P3=1
Time:219, P0=1, P1=16, P2=1, P3=1
Time:220, P0=1, P1=16, P2=1, P3=1
```

這裡與第一題結果的差異在於 Time:212 時 P1(D)輸出的結果為 205，而不是預想的 1，這裡想到的可能結論是因為 FIR\_asic 尚未計算完成，所以顯示的

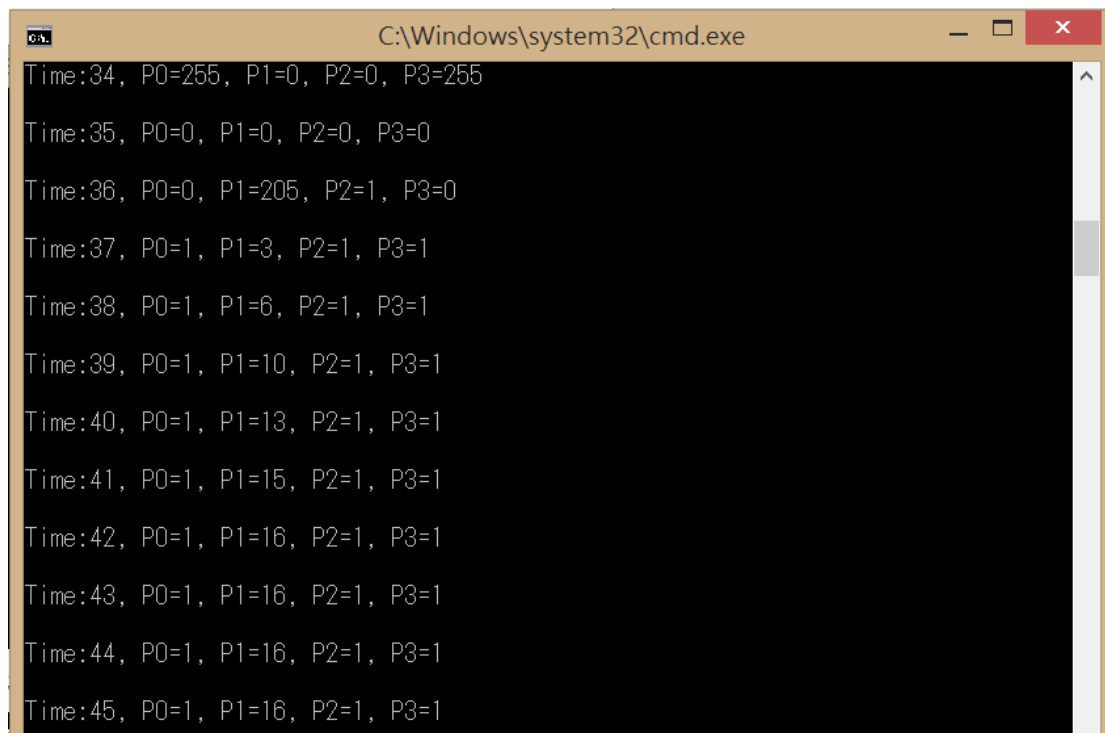
結果才會有所出入。

3. Use N multipliers and adders, run 8051 at 800MHz and FIR at 100MHz.



```
C:\Windows\system32\cmd.exe
Time:34, P0=255, P1=0, P2=0, P3=255
Time:35, P0=0, P1=0, P2=0, P3=0
Time:36, P0=0, P1=1, P2=1, P3=0
Time:37, P0=1, P1=3, P2=1, P3=1
Time:38, P0=1, P1=6, P2=1, P3=1
Time:39, P0=1, P1=10, P2=1, P3=1
Time:40, P0=1, P1=13, P2=1, P3=1
Time:41, P0=1, P1=15, P2=1, P3=1
Time:42, P0=1, P1=16, P2=1, P3=1
Time:43, P0=1, P1=16, P2=1, P3=1
Time:44, P0=1, P1=16, P2=1, P3=1
Time:45, P0=1, P1=16, P2=1, P3=1
```

4. Use one multiplier and adder, run 8051 at 800MHz and FIR at 100MHz.



```
C:\Windows\system32\cmd.exe
Time:34, P0=255, P1=0, P2=0, P3=255
Time:35, P0=0, P1=0, P2=0, P3=0
Time:36, P0=0, P1=205, P2=1, P3=0
Time:37, P0=1, P1=3, P2=1, P3=1
Time:38, P0=1, P1=6, P2=1, P3=1
Time:39, P0=1, P1=10, P2=1, P3=1
Time:40, P0=1, P1=13, P2=1, P3=1
Time:41, P0=1, P1=15, P2=1, P3=1
Time:42, P0=1, P1=16, P2=1, P3=1
Time:43, P0=1, P1=16, P2=1, P3=1
Time:44, P0=1, P1=16, P2=1, P3=1
Time:45, P0=1, P1=16, P2=1, P3=1
```

3 跟 4 中因為 8051 的 clock 從 100MHz 變成 800MHz，速度提高了八倍，所以在前面初始化的過程時間也縮短了快八倍之多，Time = 211 -> Time = 35。其他的過程都各自與 1&2 結果相同。

#### **結論：**

分析 1&2 題的模擬結果，在 Timing 的顯示結果幾乎都相同，並沒有預想中只用了單一乘法器與加法器在時間的消耗上會比用 N 個來的多，原因很可能是因為前面提到的，系統顯示對於 time 的切割單位不夠小到可以觀察到整個過程。分析 1,2 的差異，運用 N 個乘法與加法器讓整個運算能個在 one cycle 完成，可以預想到系統的面積一定比較大，但是能夠快速得到結果是它的好處，至於耗能上就必須要實際測試才會知道，因為以單位時間耗能來看 1 一定耗費較多，但是以完成同一個工作的總耗能來看，1 不一定會比 2 來的多。