

Soc Design Lab4-1 Report

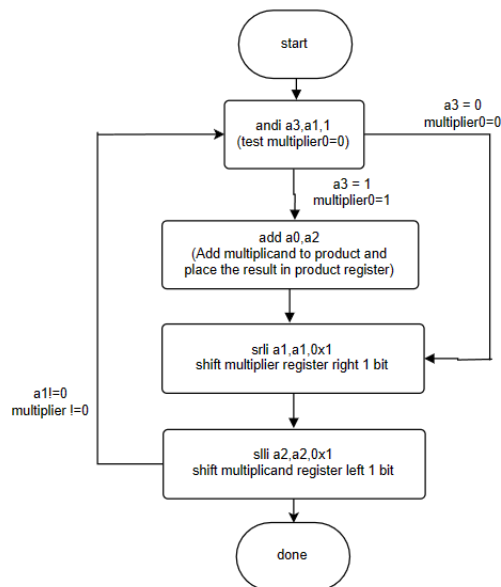
組員：王證皓、丁緒翰、潘金生

1. Explanaion of your firmware code:

A. How does it execute a multiplication in assembly code?

如何執行一個乘法，如下圖流程圖：

1. 第一步驟：測試乘數第 0 位是否為 0，如果是跳到第三步驟，如果不是則繼續第二步驟。
2. 第二步驟：把被乘數的結果相加產生結果放在 product register。
3. 第三步驟：將乘數右移一位
4. 第四步驟：將被乘數左移一位，如果乘數不等於 0，則返回第一步驟，如果乘數等於 0，則結束。



圖一 乘法流程圖

在 assembly code 乘法執行過程，其指令和指令詳細介紹如下：

mv :把 a0 的值給 a2
li :把 0 給 a0
andi :a1 與常數 1 做 AND 運算，將結果寫入 a3 中。
beqz :a3=0 的時候，pc 跳到 38000014 地址
add :將 a0 與 a2 做加法運算寫入 a0 中。
srli :將 a1 邏輯右移一位
slli :將 a2 邏輯左移一位
bnez :a1!=0 的時候，pc 跳到 38000008 地址

```

38000000 <__mulsi3>:
38000000:      00050613          mv     a2,a0
38000004:      00000513          li     a0,0
38000008:      0015f693          andi    a3,a1,1
3800000c:      00068463          beqz   a3,38000014 <__mulsi3+0x14>
38000010:      00c50533          add     a0,a0,a2
38000014:      0015d593          srli    a1,a1,0x1
38000018:      00161613          slli    a2,a2,0x1
3800001c:      fe0596e3          bnez   a1,38000008 <__mulsi3+0x8>
38000020:      00008067          ret

```

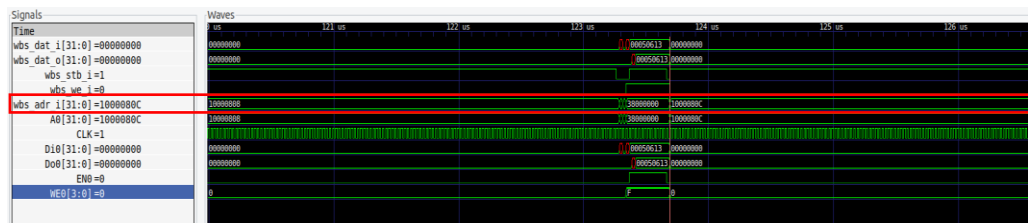
圖二 multiplication in assembly code

B. What address allocate for user project and how many space is required to allocate to firmware code?

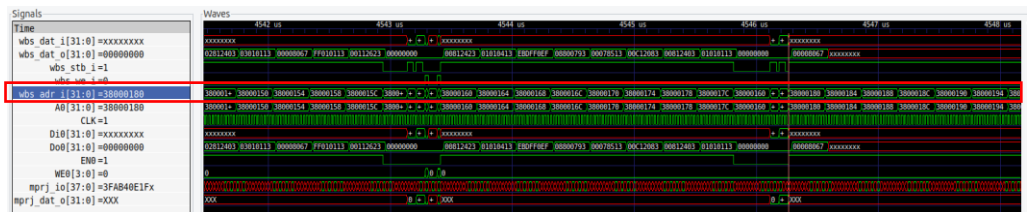
1. address allocate for user project:

起始位置:38000000

終點位置:38000180



圖三 起始位置波形圖



圖四 終點位置波形圖

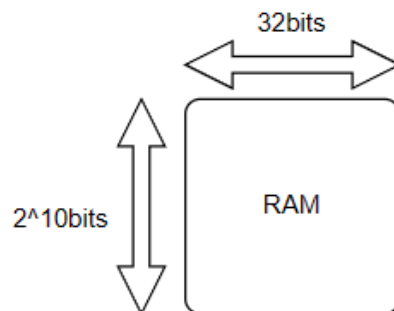
2. Space is required to allocate to firmware code:

我們設置 $N=10$ ，其大小大約等於 4kB。

```

parameter N =10 ;
(* ram_style = "block" *) reg [31:0] RAM[0:2**N-1];

```

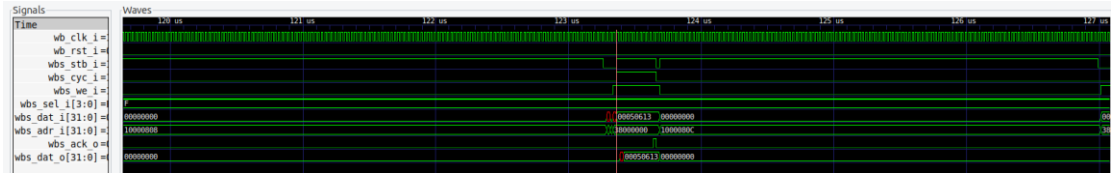


圖五 ram space

C. Interface between BRAM and wishbone:

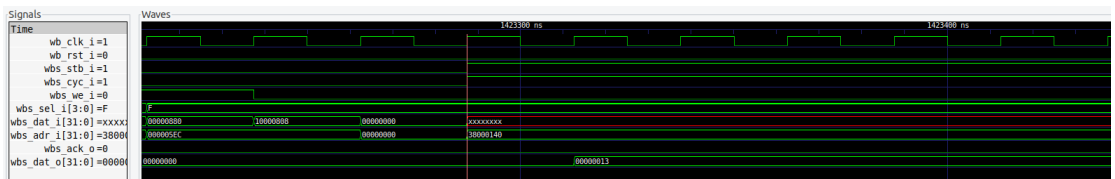
Waveform:

剛開始，當位址等於 38000000，把 firmware code 存到 memory 裡。



圖六 位址等於 38000000

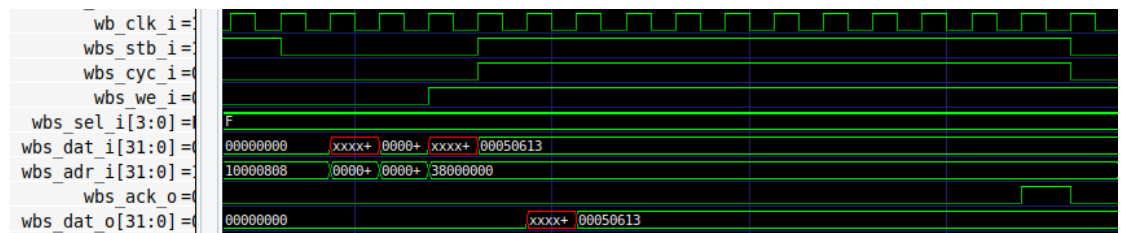
當位址等於 38000140，把 firmware code 拿出來用。



圖七 位址等於 38000140

1. Wishbone Waveform

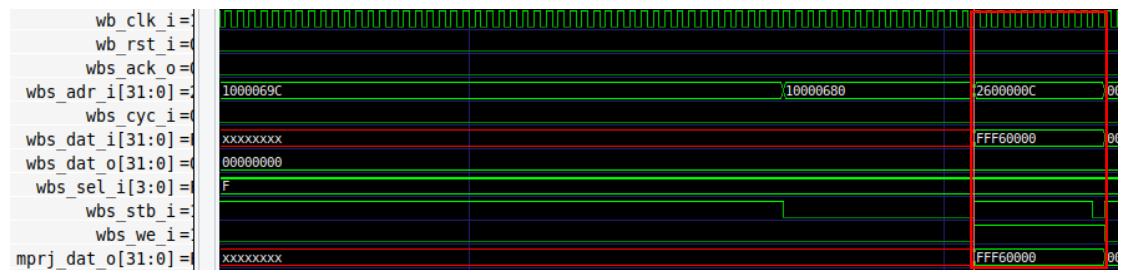
從 wbs_cyc_i 到 wbs_ack_o 會有 10 的 delay



圖八 wbs_cyc_i 到 wbs_ack_o

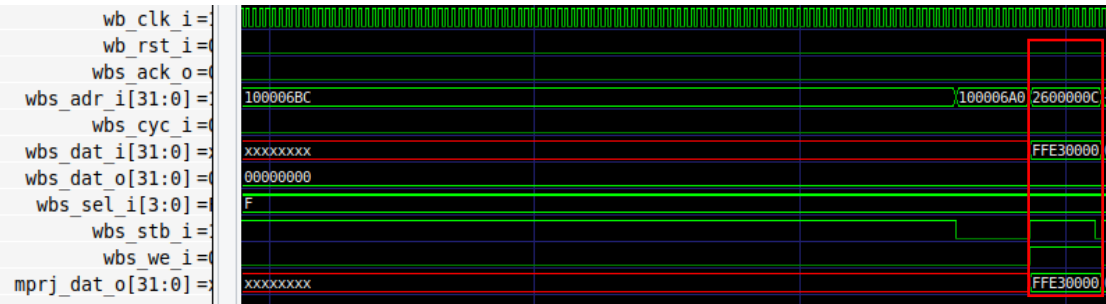
2. Output Waveform

第二筆計算結果的前 16 的 bit 為 16'hFFF6=-10



圖九 第二筆計算結果

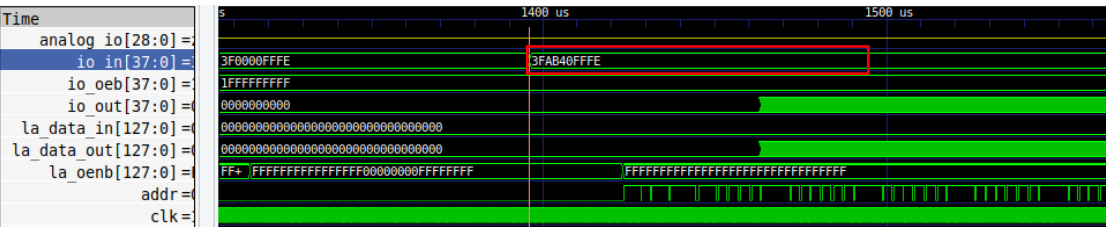
第三筆計算結果的前 16 的 bit 為 16'hFFE3=-29



圖十 第三筆計算結果

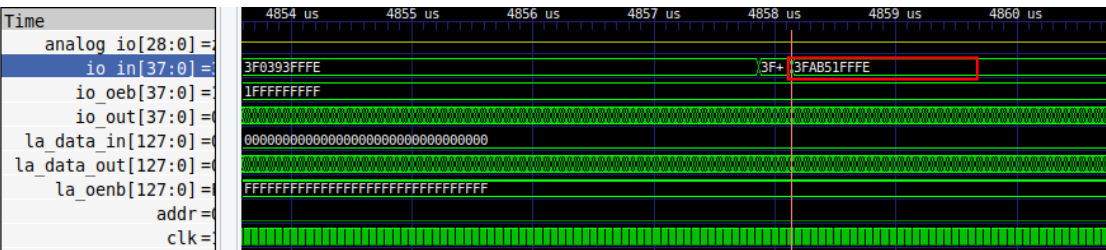
3. Start & End Mark Waveform

Start Mark=16'hAB40



圖十一 Start Mark=16'hAB40

End Mark=16'hAB51



圖十二 End Mark=16'hAB51

4. Synthesis report:

本次 Lab4-1 的合成結果 WHS:5.840ns，WHS:0.207ns。



圖十三 slack 結果

Detailed RTL components:

下圖為合成結果的 RTL components

```
Detailed RTL Component Info :
+---Adders :
      2 Input   32 Bit      Adders := 1
      2 Input   16 Bit      Adders := 1
+---Registers :
      32 Bit     Registers := 2
      16 Bit     Registers := 1
      1 Bit      Registers := 1
+---RAMs :
      32K Bit    (1024 X 32 bit)    RAMs := 1
+---Muxes :
      2 Input   32 Bit      Muxes := 7
      2 Input   16 Bit      Muxes := 1
      2 Input    8 Bit      Muxes := 1
      2 Input    1 Bit      Muxes := 2
```

圖十四 RTL components

Cell Usage:

附圖為本次合成的 Cell Usage

```
Report Cell Usage:
+-----+-----+
|      | Cell      | Count |
+-----+-----+
| 1     | BUFG      | 1     |
| 2     | CARRY4    | 12    |
| 3     | LUT2      | 20    |
| 4     | LUT3      | 10    |
| 5     | LUT4      | 24    |
| 6     | LUT5      | 38    |
| 7     | LUT6      | 5     |
| 8     | RAMB36E1  | 1     |
| 9     | FDRE      | 49    |
| 10    | IBUF      | 77    |
| 11    | OBUF      | 240   |
+-----+-----+
```

圖十五 cell usage

Timing Report Max Path Delay:

本次合成的 Data Path Delay 為 4.156ns

Max Delay Paths

```
-----  
Slack (MET) :          5.840ns (required time - arrival time)  
Source:              wbs_sel_i[2]  
                     (input port clocked by wb_clk_i {rise@0.000ns fall@5.000ns period=10.000ns})  
Destination:         count_reg[0]/D  
                     (rising edge-triggered cell FDRE clocked by wb_clk_i {rise@0.000ns fall@5.000ns period=10.000ns})  
Path Group:          wb_clk_i  
Path Type:           Setup (Max at Slow Process Corner)  
Requirement:         10.000ns (wb_clk_i rise@10.000ns - wb_clk_i rise@0.000ns)  
Data Path Delay:      4.156ns (logic 1.344ns (32.328%) route 2.813ns (67.672%))  
Logic Levels:        4 (IBUF=1 LUT4=1 LUT5=1 LUT6=1)  
Input Delay:         3.000ns  
Clock Path Skew:     2.988ns (DCD - SCD + CPR)  
  Destination Clock Delay (DCD):  2.988ns = ( 12.988 - 10.000 )  
  Source Clock Delay (SCD):       0.000ns  
  Clock Pessimism Removal (CPR):  0.000ns  
Clock Uncertainty:    0.035ns ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE  
  Total System Jitter (TSJ):      0.071ns  
  Total Input Jitter (TIJ):       0.000ns  
  Discrete Jitter (DJ):          0.000ns  
  Phase Error (PE):              0.000ns
```

圖十六 Max Path Delay

5. Github Link:

<https://github.com/TINGxTING/Soc-design-Lab4-1/tree/main>