112061603 李柏叡
112061620 郭邑哲
112061621 貢暐家

SOC Design Lab4-1

- Overview

一般來說，C Code 在 compile 完成後會變成 hex 檔(assembly code)，存在 SPI-FLASH 中，等到需要用到時，再由 CPU 進行提取位在 SPI-FLASH 中的 Firmware code，並透過 interface 對硬體進行控制，而在 Caravel SOC 中，CPU 透過 Wishbone bus 提取位在 SPI-FLASH 的 Firmware code，再透過 Wishbone 或 LA Module(logic analyzer)，對 hardware 進行控制。

但因為將 Firmware code 從 SPI-FLASH 讀取出來，在到 CPU 做執行的時間可能需要很長的 cycle。且當它要對硬體做控制時，已消耗太多時間。以 FIR 為例，當 Data 從 SPI-FLASH 中的 Firmware code 讀取出來給 FIR 會需要花很長的 Cycle，而 FIR 執行完後還要等 Firmware code，因此在怎麼 improve FIR 的效能也沒用。

LAB4-1 中，為了改善此問題，我們在系統一開始時便將 Firmware code 從 SPI-FLASH 搬到 user project memory 中，這樣可以使我們的執行時間較短。

- How does it execute a multiplication in assembly code?

因為 hex 檔放在 mprj ram 中，因此我們從 out 檔中可以發現以下說明:

```
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
```

mv a2,a0//複製 a0 的值給 a2

li a0,0 //0 給 a0

andi a3,a1,1//a3=a1&1(每一位做 and→即保留最後一位) a3=0x0000 or 0x0001(if 32bits)

beqz a3,38000014//a3 等於 0，則跳到 38000014

add a0,a0,a2//a0=a0+a2

srli a1,a1,0x1//a1=a1 右移 1 位

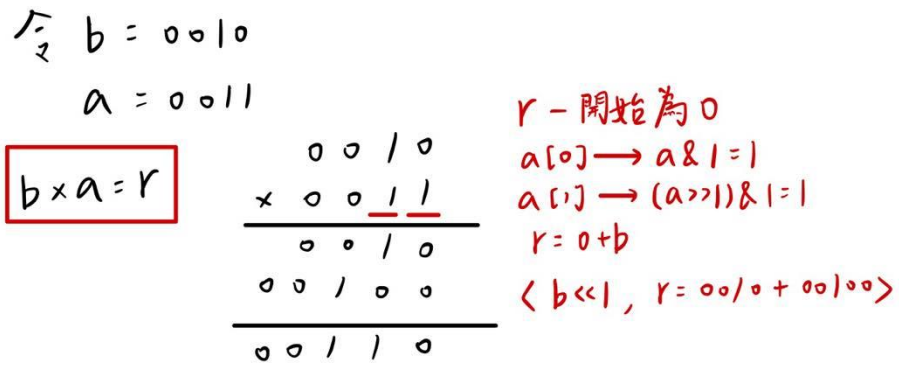
slli a2,a2,0x1//a2=a2 左移 1 位

bnez a1,38000008//a1 不等於 0，則跳到 38000008

以下為其相關演算法:

```
unsigned int
__mulsi3 (unsigned int a, unsigned int b)
{
  unsigned int r = 0;

  while (a)
    {
      if (a & 1)
        r += b;
      a >>= 1;
      b <<= 1;
    }
  return r;
}
```

透過以下推導可以得知此為乘法運算:



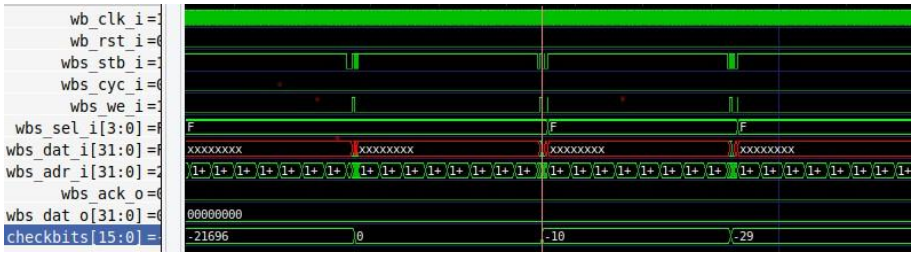
- What address allocate for user project and how many space is required to allocate to firmware code ?

此次 lab 中，user project(MPRJ RAM) starts from address 38000000 to address 380001c0。另外從 hex 檔中的內容可以看出其為 7520Bytes，而由於此次 Bram 為 Word address，因此我們需要將 Bram 的 N 設為 11，即為 2^11-1=2047words，如此便可裝下 1880words 的 hex 檔。

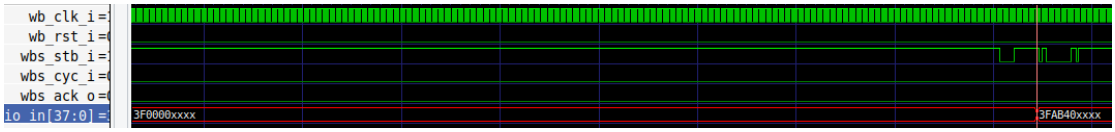
```
38000000 <__mulsi3>:
38000000:       00050613                mv      a2,a0
38000004:       00000513                li      a0,0
38000008:       0015f693                andi    a3,a1,1
3800000c:       00068463                beqz    a3,38000014 <__mulsi3+0x14>
38000010:       00c50533                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

38000024 <initfir>:
38000024:       fe010113                addi    sp,sp,-32
38000028:       00812e23                sw      s0,28(sp)
3800002c:       02010413                addi    s0,sp,32
38000030:       fe042623                sw      zero,-20(s0)
38000034:       0380006f                j       3800006c <initfir+0x48>
38000038:       05c00713                li      a4,92
3800003c:       fec42783                lw      a5,-20(s0)
38000040:       00279793                slli    a5,a5,0x2
38000044:       00f707b3                add     a5,a4,a5
38000048:       0007a023                sw      zero,0(a5)
3800004c:       08800713                li      a4,136
38000050:       fec42783                lw      a5,-20(s0)
38000054:       00279793                slli    a5,a5,0x2
38000058:       00f707b3                add     a5,a4,a5
3800005c:       0007a023                sw      zero,0(a5)
38000060:       fec42783                lw      a5,-20(s0)
38000064:       00178793                addi    a5,a5,1
38000068:       fef42623                sw      a5,-20(s0)
3800006c:       fec42703                lw      a4,-20(s0)
38000070:       00a00793                li      a5,10
38000074:       fce7d2e3                bge     a5,a4,38000038 <initfir+0x14>
38000078:       00000013                nop
3800007c:       00000013                nop
38000080:       01c12403                lw      s0,28(sp)
38000084:       02010113                addi    sp,sp,32
38000088:       00008067                ret

380001a4:       00a00793                li      a5,10
380001a8:       f0e7d0e3                bge     a5,a4,380000a8 <fir+0x1c>
380001ac:       08800793                li      a5,136
380001b0:       00078513                mv      a0,a5
380001b4:       01c12083                lw      ra,28(sp)
380001b8:       01812403                lw      s0,24(sp)
380001bc:       02010113                addi    sp,sp,32
380001c0:       00008067                ret
```

- Waveform :

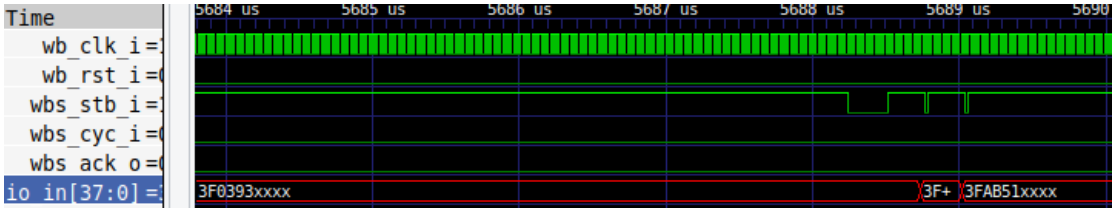
1.透過 checkbits 做比對，確認此次 Lab FIR 的輸出為以下的正確數字。



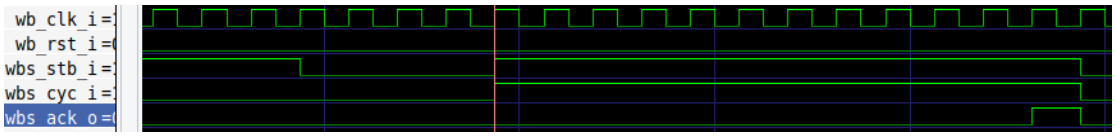
2.Start --> mprj_io[31:16]=16'hAB40



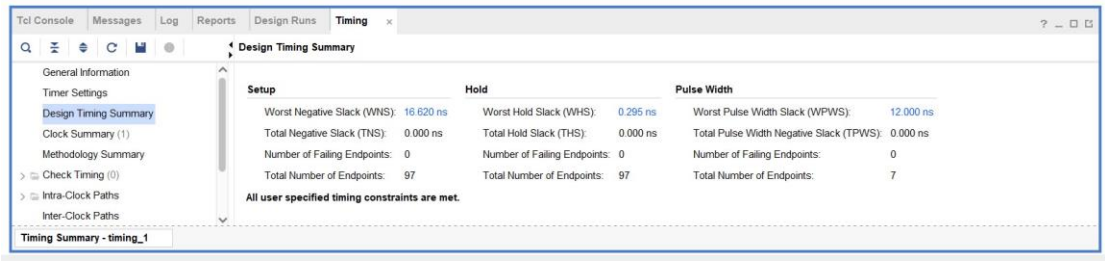
3. End --> mprj_io[31:16]=16'hAB51



4.從 wbs_stb_i 及 wbs_cyc_i 到 wbs_ack_o 共 delay 11clk。



- Synthesis report



LUT and FF:

1. Slice Logic

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	19	0	0	8000	0.24
LUT as Logic	19	0	0	8000	0.24
LUT as Memory	0	0	0	5000	0.00
Slice Registers	4	0	0	16000	0.03
Register as Flip Flop	4	0	0	16000	0.03
Register as Latch	0	0	0	16000	0.00
F7 Muxes	0	0	0	7300	0.00
F8 Muxes	0	0	0	3650	0.00

RTL Component

```
Detailed RTL Component Info :
+---Adders :
           2 Input    4 Bit       Adders := 1
+---Registers :
                       32 Bit    Registers := 1
                        4 Bit    Registers := 1
+---RAMs :
                      16K Bit   (512 X 32 bit)          RAMs := 1
+---Muxes :
           2 Input   32 Bit        Muxes := 6
           2 Input    8 Bit        Muxes := 1
           2 Input    4 Bit        Muxes := 1
           2 Input    1 Bit        Muxes := 4
```

Report Cell Usage:

```
Report Cell Usage:
+------+---------+------+
|      |Cell     |Count |
+------+---------+------+
|1     |BUFG     |     1|
|2     |LUT2     |     4|
|3     |LUT3     |    11|
|4     |LUT4     |     2|
|5     |LUT5     |     2|
|6     |LUT6     |     1|
|7     |RAMB18E1 |     1|
|8     |FDRE     |     4|
|9     |IBUF     |    50|
|10    |OBUF     |    33|
|11    |OBUFT    |   207|
+------+---------+------+
```

github link:https://github.com/816-allen?tab=repositories