SOC Lab4-1 report

Explanation of your firmware code：

這次 firmware code 是用 C 完成的。一開始先將 inputbuffer 和 outputsignal 初始化，接著將 inputbuffer shift 一位，input 存入 inputbuffer[0]中，最後再 for 迴圈相乘累加。

How does it execute a multiplication in assembly code：

組合語言其實沒有乘法，它是由 mv、add 和一些奇怪的指令所組成的。

```
Disassembly of section .mprjram:

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
```

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

```
Disassembly of section .mprjram:

38000000 <__mulsi3>:
38000000: 00050613           mv  a2,a0
38000004: 00000513           li  a0,0
38000214: 01412483           lw  s1,20(sp)
38000218: 02010113           addi  sp,sp,32
3800021c: 00008067           ret
```

佔 0x21c byte = 540 byte

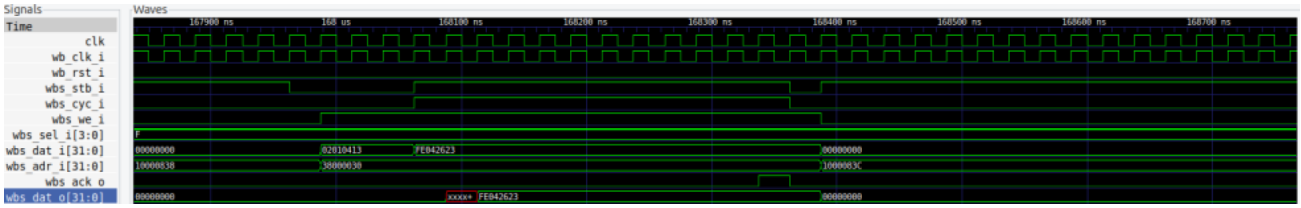
Interface between BRAM and wishbone（沒有用 FSM 實現）：

接口：

```
bram user_bram (
        .CLK(wb_clk_i),
        .WE0(we),
        .EN0(valid),
        .Di0(wbs_dat_i),
        .Do0(rdata),
        .A0(wbs_adr_i)
    );
```

```
assign valid = (wbs_stb_i == 1) && (wbs_cyc_i == 1) && (wbs_adr_i[31:24] == 8'h38);
assign we    = {4{wbs_we_i & valid}};
```

waveform：



Synthesis report：

1. Slice Logic

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	53	0	0	53200	0.10
LUT as Logic	53	0	0	53200	0.10
LUT as Memory	0	0	0	17400	0.00
Slice Registers	33	0	0	106400	0.03
Register as Flip Flop	33	0	0	106400	0.03
Register as Latch	0	0	0	106400	0.00
F7 Muxes	0	0	0	26600	0.00
F8 Muxes	0	0	0	13300	0.00

2. Memory

Site Type	Used	Fixed	Prohibited	Available	Util%
Block RAM Tile	2	0	0	140	1.43
RAMB36/FIFO*	2	0	0	140	1.43
RAMB36E1 only	2				
RAMB18	0	0	0	280	0.00