- Explanation of your firmware code
    - How does it execute a multiplication in assembly code

```
122 .L6:
123         .loc 1 13 15
124         lw      a4,-20(s0)
125         lw      a5,-24(s0)
126         sub     a5,a4,a5
127         .loc 1 13 12
128         blt     a5,zero,.L5
129         .loc 1 14 26
130         lui     a5,%hi(outputsignal)
131         addi    a4,a5,%lo(outputsignal)
132         lw      a5,-20(s0)
133         slli    a5,a5,2
134         add     a5,a4,a5
135         lw      s1,0(a5)
136         .loc 1 14 37
137         lui     a5,%hi(taps)
138         addi    a4,a5,%lo(taps)
139         lw      a5,-24(s0)
140         slli    a5,a5,2
141         add     a5,a4,a5
142         lw      a3,0(a5)
143         .loc 1 14 56
144         lw      a4,-20(s0)
145         lw      a5,-24(s0)
146         sub     a5,a4,a5
147         .loc 1 14 54
148         lui     a4,%hi(inputsignal)
149         addi    a4,a4,%lo(inputsignal)
150         slli    a5,a5,2
151         add     a5,a4,a5
152         lw      a5,0(a5)
153         .loc 1 14 41
154         mv      a1,a5
155         mv      a0,a3
156         call    __mulsi3
157         mv      a5,a0
158         .loc 1 14 30
159         add     a4,s1,a5
160         lui     a5,%hi(outputsignal)
161         addi    a3,a5,%lo(outputsignal)
162         lw      a5,-20(s0)
163         slli    a5,a5,2
164         add     a5,a3,a5
165         sw      a4,0(a5)
```

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

```
MEMORY {
        vexriscv_debug : ORIGIN = 0xf00f0000, LENGTH = 0x00000100
        dff : ORIGIN = 0x00000000, LENGTH = 0x00000400
        dff2 : ORIGIN = 0x00000400, LENGTH = 0x00000200
        flash : ORIGIN = 0x10000000, LENGTH = 0x01000000
        mprj : ORIGIN = 0x30000000, LENGTH = 0x00100000
        mprjram : ORIGIN = 0x38000000, LENGTH = 0x00400000
        hk : ORIGIN = 0x26000000, LENGTH = 0x00100000
        csr : ORIGIN = 0xf0000000, LENGTH = 0x00010000
}
```

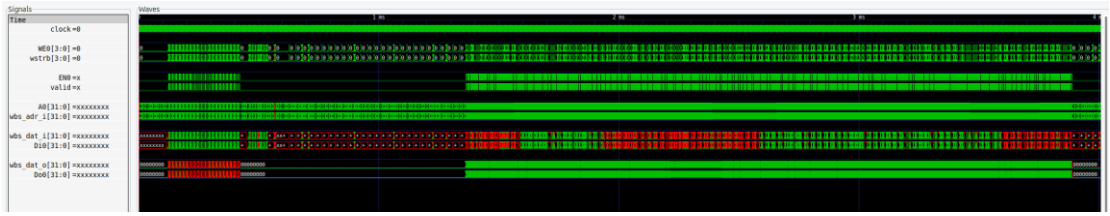
```
module bram(
    CLK,
    WE0,
    EN0,
    Di0,
    Do0,
    A0
);

    input   wire            CLK;
    input   wire    [3:0]   WE0;
    input   wire            EN0;
    input   wire    [31:0]  Di0;
    output  reg     [31:0]  Do0;
    input   wire    [31:0]   A0;

    // 4 kB
    parameter N =10 ;
    (* ram_style = "block" *) reg [31:0] RAM[0:2**N-1];


    always @(posedge CLK)
        if(EN0) begin
            Do0 <= RAM[A0[N-1:0]];
            if(WE0[0]) RAM[A0[N-1:0]][7:0] <= Di0[7:0];
            if(WE0[1]) RAM[A0[N-1:0]][15:8] <= Di0[15:8];
            if(WE0[2]) RAM[A0[N-1:0]][23:16] <= Di0[23:16];
            if(WE0[3]) RAM[A0[N-1:0]][31:24] <= Di0[31:24];
        end
        else
            Do0 <= 32'b0;
endmodule
```

- Interface between BRAM and wishbone
  - Waveform from xsim



```
 97     // WB MI A
 98     assign valid = wbs_cyc_i && wbs_stb_i && decoded;
 99     assign wstrb = wbs_sel_i & {4{wbs_we_i}};
100     assign wbs_dat_o = rdata;
101     assign wdata = wbs_dat_i;
102     assign wbs_ack_o = ready;
```

```
155     bram user_bram (
156         .CLK(clk),
157         .WE0(wstrb),
158         .EN0(valid),
159         .Di0(wbs_dat_i),
160         .Do0(rdata),
161         .A0(wbs_adr_i)
162     );
```

```
121     always @(posedge clk) begin
122         if (rst) begin
123             ready <= 1'b0;
124             delayed_count <= 16'b0;
125         end else begin
126             ready <= 1'b0;
127             if ( valid && !ready ) begin
128                 if ( delayed_count == DELAYS ) begin
129                     delayed_count <= 16'b0;
130                     ready <= 1'b1;
131                 end else begin
132                     delayed_count <= delayed_count + 1;
133                 end
134             end
135         end
136     end
```

由上圖描述將相對應 wire 接起來，再加上 delay 10 個 cycle 讓 output 可以正常輸出。

- Synthesis report

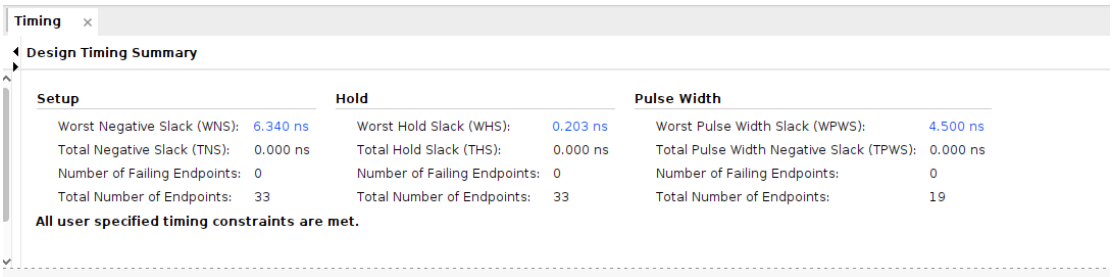
1. Slice Logic

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	26	0	0	53200	0.05
LUT as Logic	26	0	0	53200	0.05
LUT as Memory	0	0	0	17400	0.00
Slice Registers	17	0	0	106400	0.02
Register as Flip Flop	17	0	0	106400	0.02
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	1	0	0	140	0.71
RAMB36/FIFO*	1	0	0	140	0.71
RAMB36E1 only	1				
RAMB18	0	0	0	280	0.00

- Timing Report
  - Slack



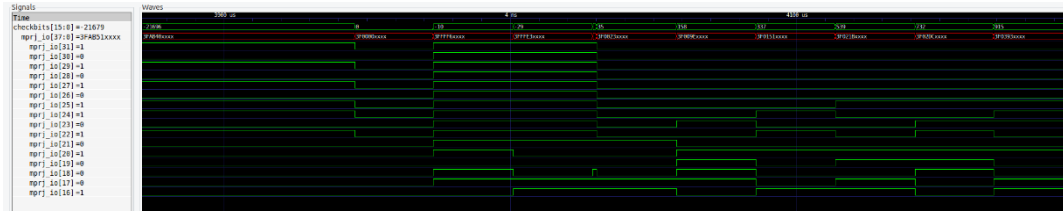
  - Max delay

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             6.340ns  (required time - arrival time)
  Source:                 delayed_count_reg[4]/C
                            (rising edge-triggered cell FDRE clocked by wb_clk_i  {rise@0.000ns fall@5.000ns period=10.000ns})
  Destination:            delayed_count_reg[12]/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:        3.557ns  (logic 1.936ns (54.428%)  route 1.621ns (45.572%))
  Logic Levels:           4  (CARRY4=3 LUT2=1)
  Clock Path Skew:        -0.145ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    2.137ns = ( 12.137 - 10.000 )
    Source Clock Delay      (SCD):    2.479ns
    Clock Pessimism Removal (CPR):    0.198ns
  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
```

```
    Location             Delay type                Incr(ns)  Path(ns)    Netlist Resource(s)
  -------------------------------------------------------------------    -------------------
                         (clock wb_clk_i rise edge)
                                                    0.000     0.000 r
                                                    0.000     0.000 r  wb_clk_i (IN)
                         net (fo=0)                 0.000     0.000    wb_clk_i
                                                                     r  wb_clk_i_IBUF_inst/I
                         IBUF (Prop_ibuf_I_O)       0.972     0.972 r  wb_clk_i_IBUF_inst/O
                         net (fo=1, unplaced)       0.800     1.771    user_bram/wb_clk_i_IBUF
                                                                     r  user_bram/RAM_reg_i_1/I0
                         LUT3 (Prop_lut3_I0_O)      0.124     1.895 r  user_bram/RAM_reg_i_1/O
                         net (fo=19, unplaced)      0.584     2.479    clk
                         FDRE                                        r  delayed_count_reg[4]/C
  -------------------------------------------------------------------    -------------------
                         FDRE (Prop_fdre_C_Q)       0.518     2.997 r  delayed_count_reg[4]/Q
                         net (fo=2, unplaced)       0.994     3.991    delayed_count_reg_n_0_[4]
                                                                     r  delayed_count_reg[4]_i_2/S[3]
                         CARRY4 (Prop_carry4_S[3]_CO[3])
                                                    0.671     4.662 r  delayed_count_reg[4]_i_2/CO[3]
                         net (fo=1, unplaced)       0.009     4.671    delayed_count_reg[4]_i_2_n_0
                                                                     r  delayed_count_reg[8]_i_2/CI
                         CARRY4 (Prop_carry4_CI_CO[3])
                                                    0.117     4.788 r  delayed_count_reg[8]_i_2/CO[3]
                         net (fo=1, unplaced)       0.000     4.788    delayed_count_reg[8]_i_2_n_0
                                                                     r  delayed_count_reg[12]_i_2/CI
                         CARRY4 (Prop_carry4_CI_O[3])
                                                    0.331     5.119 r  delayed_count_reg[12]_i_2/O[3]
                         net (fo=1, unplaced)       0.618     5.737    data0[12]
                                                                     r  delayed_count[12]_i_1/I0
                         LUT2 (Prop_lut2_I0_O)      0.299     6.036 r  delayed_count[12]_i_1/O
                         net (fo=1, unplaced)       0.000     6.036    delayed_count[12]
                         FDRE                                        r  delayed_count_reg[12]/D
  -------------------------------------------------------------------    -------------------
```

```
                         (clock wb_clk_i rise edge)
                                                    10.000    10.000 r
                                                     0.000    10.000 r  wb_clk_i (IN)
                         net (fo=0)                  0.000    10.000    wb_clk_i
                                                                      r  wb_clk_i_IBUF_inst/I
                         IBUF (Prop_ibuf_I_O)        0.838    10.838 r  wb_clk_i_IBUF_inst/O
                         net (fo=1, unplaced)        0.760    11.598    user_bram/wb_clk_i_IBUF
                                                                      r  user_bram/RAM_reg_i_1/I0
                         LUT3 (Prop_lut3_I0_O)       0.100    11.698 r  user_bram/RAM_reg_i_1/O
                         net (fo=19, unplaced)       0.439    12.137    clk
                         FDRE                                         r  delayed_count_reg[12]/C
                         clock pessimism             0.198    12.334
                         clock uncertainty          -0.035    12.299
                         FDRE (Setup_fdre_C_D)       0.077    12.376    delayed_count_reg[12]
-------------------------------------------------------------------
                         required time                        12.376
                         arrival time                         -6.036
-------------------------------------------------------------------
                         slack                                 6.340
```

- Mprj_io & checkbits
  - Waveform
    - Fir outputs



上圖所示，可以從 checkbits 看出 fir 的輸出結果，

因為一開始有將 mprj_io[31:16]宣告為 output。

    - Spi_flash to bram

上圖所示，從 spi flash 讀出 firmware 送到 bram。

  - Simulation log

```
ubuntu@ubuntu2004:~/Desktop/nthu_SoC/lab4/lab4-1/testbench/counter_la_fir$ source run_clean
ubuntu@ubuntu2004:~/Desktop/nthu_SoC/lab4/lab4-1/testbench/counter_la_fir$ source run_sim
Reading counter_la_fir.hex
counter_la_fir.hex loaded into memory
Memory 5 bytes = 0x6f 0x00 0x00 0x0b 0x13
VCD info: dumpfile counter_la_fir.vcd opened for output.
LA Test 1 started
LA Test 2 passed
ubuntu@ubuntu2004:~/Desktop/nthu_SoC/lab4/lab4-1/testbench/counter_la_fir$
```