國立清華大學 電機工程系
112 學年度第一學期

SOC Design Laboratory

Lab #4-2

學校系所：清大電機所、電子所

學號姓名：112061611 陳伯丞

112061524 葉又菘

110063553 張傑閔

指導教授：賴瑾 教授（Prof. Jiin Lai）

中 華 民 國 一 一 二 年 十 一 月

Table of Contents

Block diagram

1.1 Datapath

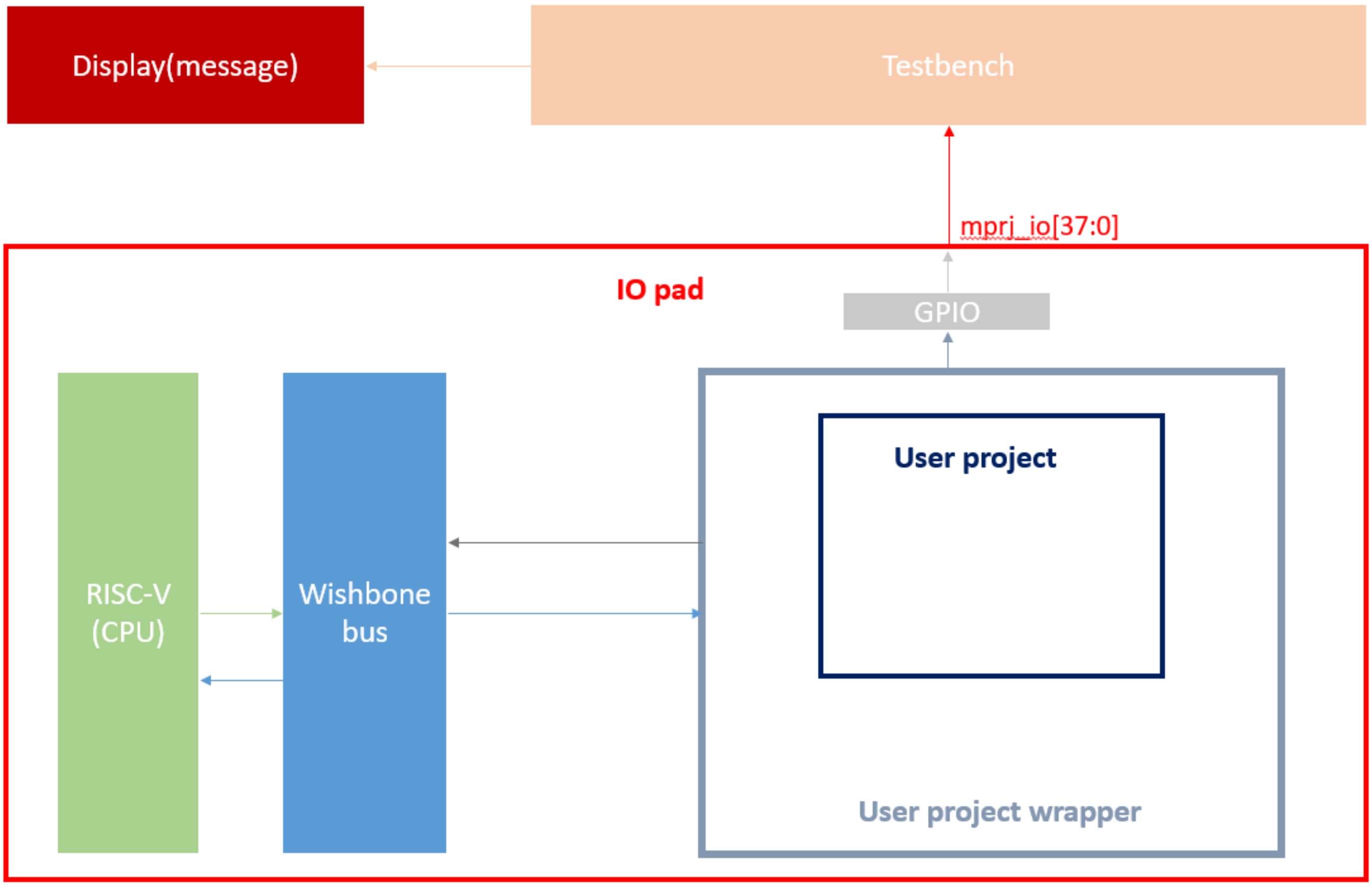

Fig 1. Datapath diagram

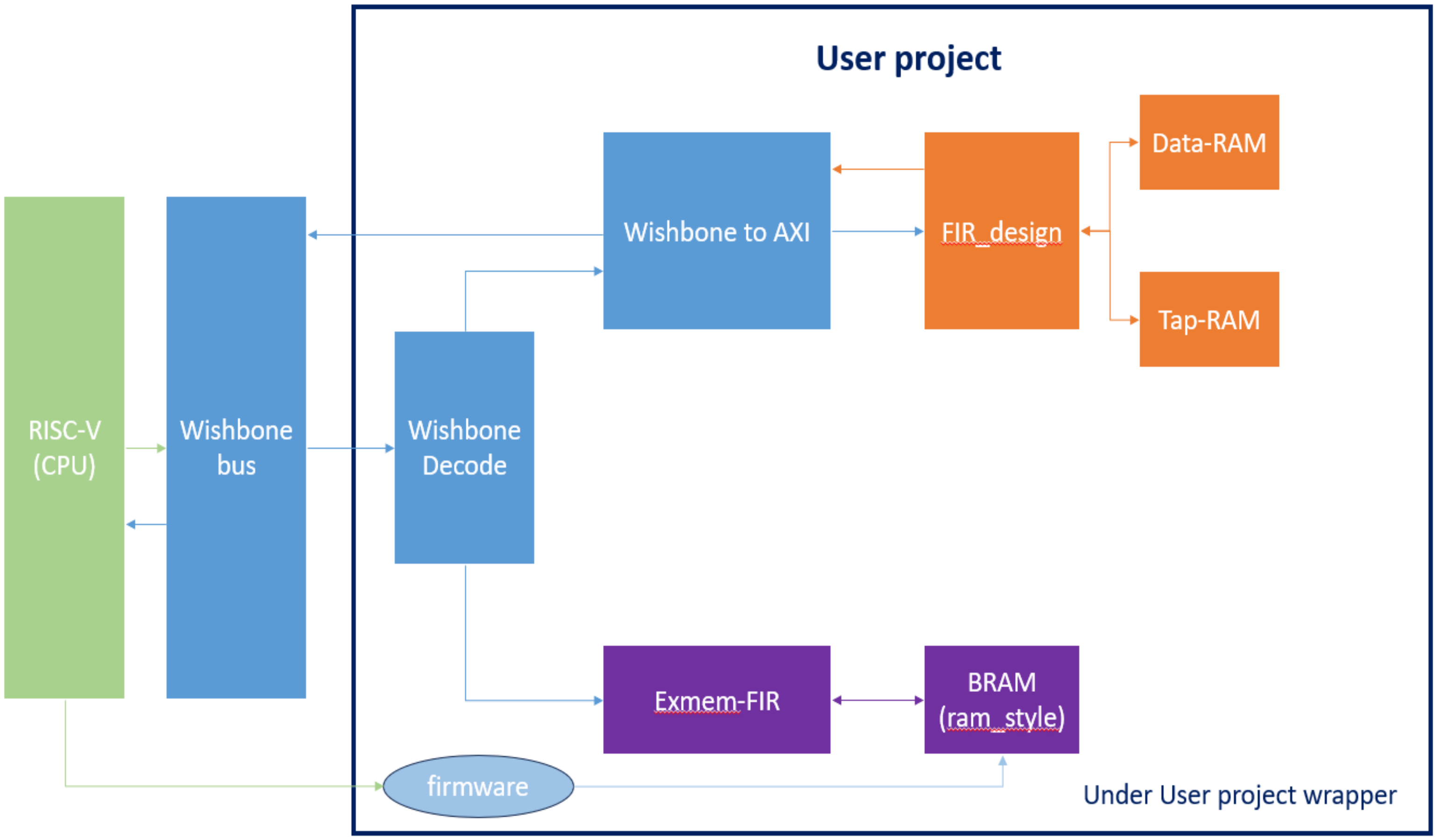

Fig 2. User project block of Datapath diagram

1.2 Controls-path

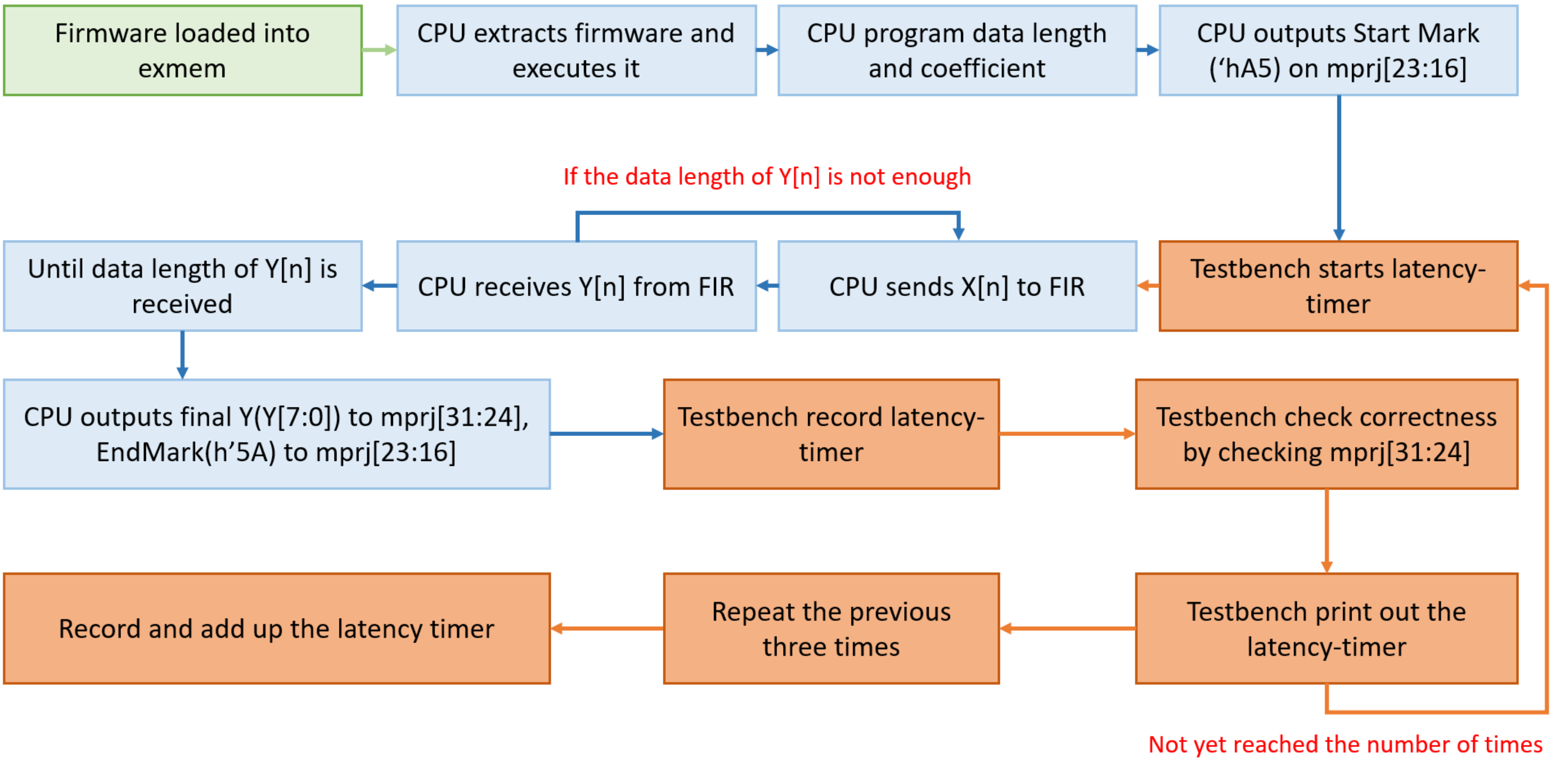

Fig 3. Control-path diagram

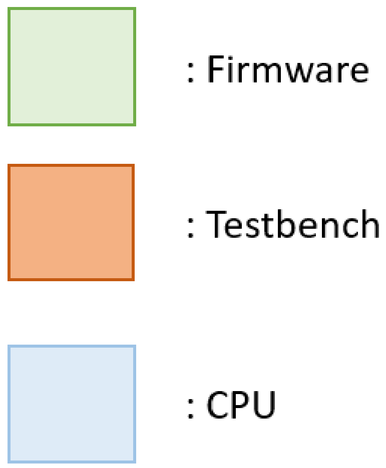

Fig 4. Block representation of Control-path diagram

Description:

First, load the firmware code into BRAM and execute it. Then the CPU will execute the program coefficient and data length according to the firmware code. After the program is completed, mprj[23:16] will output a start mark (‘hA5) to notify Testbench to start latency-timer, then the CPU sends X[n] to FIR calculation, and receives Y[n] after calculation. The CPU will then repeatedly send X[n] and receiving Y[n] until data length of Y[n] is received. when finish, the CPU write final Y[7:0] output to mprj [31:24], and write EndMark('h5A) to mprj[23:16], then record the latency timer, Testbench check correctness by checking mprj[31:24], and print out the latency-timer, and finally repeat three times(The CPU output the start Mark('hA5) on mprj[23:16], sends x[n] to FIR, receives Y[n] from FIR and until the data length of Y[n] is received and write the final Y[n] to mprj[31:24] and output the EndMark('h5A) to mprj[23:16], then check correctness to mprj[31:24]) , and finally record and add up the latency-timer.

Interface protocol

firmware

```c
for(int i=0; i<3; i++){
    reg_mprj_datal = 0x00A50000;
    int* tmp = fir();
    reg_mprj_datal = (*tmp & 0xFF) << 24 | (0x5A << 16);
}
```

Testbench

```verilog
initial begin
    // analysis timer
    // 1 time
    wait(checkbits[7:0] == 8'hA5);
    $display("1st: MPRJ-Logic WB started, time at ", $realtime, " ns");

    wait(checkbits[7:0] == 8'h5A);
    $display("1st: Mega-Project WB (RTL) passed, time at ", $realtime, " ns");

    // 2 time
    wait(checkbits[7:0] == 8'hA5);
    $display("2nd: MPRJ-Logic WB started, time at ", $realtime, " ns");

    wait(checkbits[7:0] == 8'h5A);
    $display("2nd: Mega-Project WB (RTL) passed, time at ", $realtime, " ns");

    // 3 time
    wait(checkbits[7:0] == 8'hA5);
    $display("3rd: MPRJ-Logic WB started, time at ", $realtime, " ns");

    wait(checkbits[7:0] == 8'h5A);
    $display("3rd: Mega-Project WB (RTL) passed, time at ", $realtime, " ns");

    #10000;
    $finish;
end
```

Fig 5. Interface between firmware and Testbench

Fig 6. Interface between firmware and User_project

Waveform

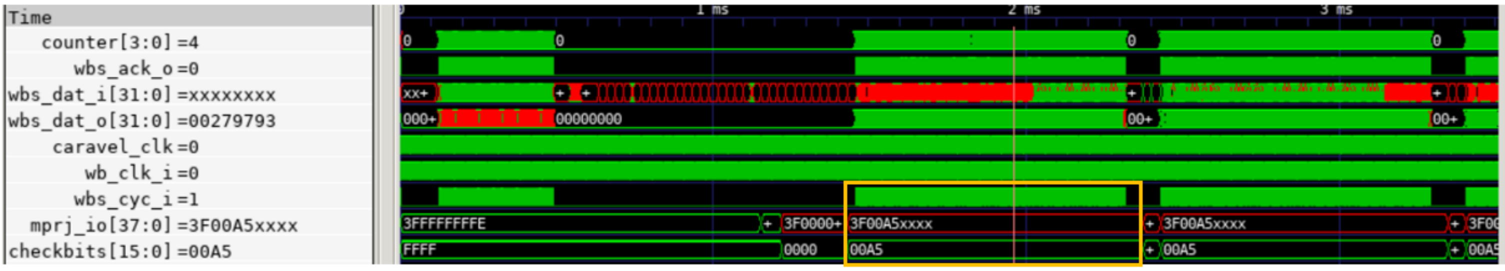

Fig 7. CPU interaction with Wishbone and mprj_io

Description:

From the yellow box in Fig 7, you can see that when the CPU sends a request to the user project through wishbone, the wbs_cyc_i will be pulled to 1, and then wb_decode in the user project will determine whether wbs_cyc_i is to be given to exmem or FIR, and then the hardware that receives the signal will start executing the behavior. Before executing the behavior, the CPU will first outputs a start mark('hA5) on mprj[23:16] to tell testbench to perform, so you can see the Start mark ('hA5) output by mprj[23:16] from Fig 7, and then the check bit is used by testbench to check whether the mprj output is correct.

Questions

What is the FIR engine theoretical throughput?

Description:

It takes 11 cycles to complete a FIR operation, and a total of 64 data are continuously sent to the FIR operation, so it is 11x1+63 = 74 cycles.

Throughput = $\frac{1}{74}$

What is latency for firmware to feed data?

```
1st: MPRJ-Logic WB started, time at 1429912.500 ns
1st: Mega-Project WB (RTL) passed, time at 2373012.500 ns
2nd: MPRJ-Logic WB started, time at 2430912.500 ns
2nd: Mega-Project WB (RTL) passed, time at 3347512.500 ns
3rd: MPRJ-Logic WB started, time at 3405412.500 ns
3rd: Mega-Project WB (RTL) passed, time at 4322012.500 ns
```

Fig 8. Show message in display

Description:

Latency = 2373012.500 − 1429912.500 = 943100(ns)

∵ 1ns per cycle, so it is 943100 cycles.

What techniques used to improve the throughput?

- Does bram12 give better performance? In what way?

  Description:

  Yes, we can put data in one more location to reduce the number of reads and writes, but it will increase the area.

- Can you suggest other method to improve the performance?

  Description:

  Use circuit techniques such as unfolding, pipelines to increase throughput and performance.

GitHub

https://github.com/ken01235/SOC_Design/tree/master/course-lab_4-2%20report