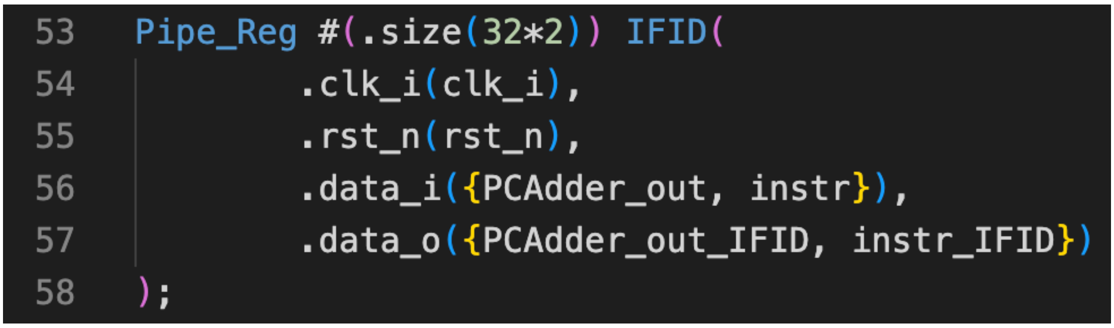
Computer Organization

1. The input fields of each pipeline register:

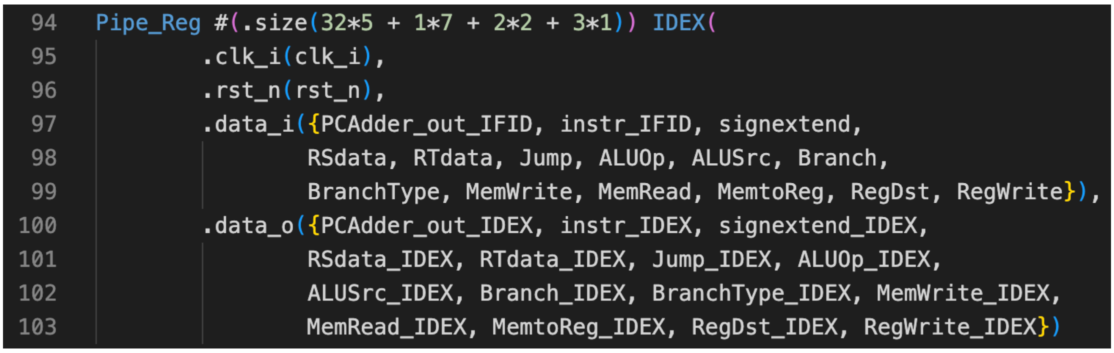
There are 4 pipeline registers: IFID, IDEX, EXMEM, and MEMWB.

IFID:



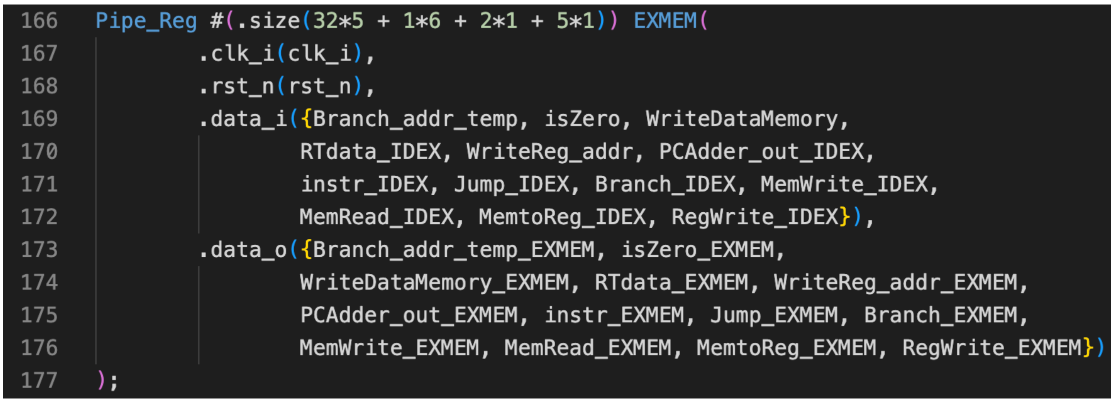
```
Pipe_Reg #(.size(32*2)) IFID(
        .clk_i(clk_i),
        .rst_n(rst_n),
        .data_i({PCAdder_out, instr}),
        .data_o({PCAdder_out_IFID, instr_IFID})
);
```

IDEX:



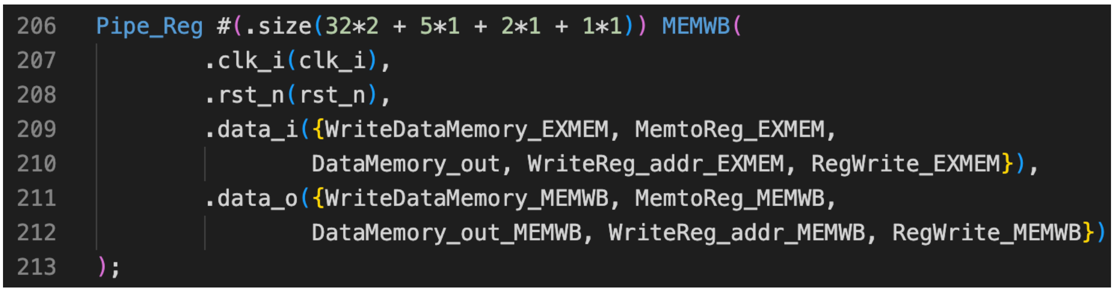
```
Pipe_Reg #(.size(32*5 + 1*7 + 2*2 + 3*1)) IDEX(
        .clk_i(clk_i),
        .rst_n(rst_n),
        .data_i({PCAdder_out_IFID, instr_IFID, signextend,
                RSdata, RTdata, Jump, ALUOp, ALUSrc, Branch,
                BranchType, MemWrite, MemRead, MemtoReg, RegDst, RegWrite}),
        .data_o({PCAdder_out_IDEX, instr_IDEX, signextend_IDEX,
                RSdata_IDEX, RTdata_IDEX, Jump_IDEX, ALUOp_IDEX,
                ALUSrc_IDEX, Branch_IDEX, BranchType_IDEX, MemWrite_IDEX,
                MemRead_IDEX, MemtoReg_IDEX, RegDst_IDEX, RegWrite_IDEX})
```

EXMEM:



```
Pipe_Reg #(.size(32*5 + 1*6 + 2*1 + 5*1)) EXMEM(
        .clk_i(clk_i),
        .rst_n(rst_n),
        .data_i({Branch_addr_temp, isZero, WriteDataMemory,
                RTdata_IDEX, WriteReg_addr, PCAdder_out_IDEX,
                instr_IDEX, Jump_IDEX, Branch_IDEX, MemWrite_IDEX,
                MemRead_IDEX, MemtoReg_IDEX, RegWrite_IDEX}),
        .data_o({Branch_addr_temp_EXMEM, isZero_EXMEM,
                WriteDataMemory_EXMEM, RTdata_EXMEM, WriteReg_addr_EXMEM,
                PCAdder_out_EXMEM, instr_EXMEM, Jump_EXMEM, Branch_EXMEM,
                MemWrite_EXMEM, MemRead_EXMEM, MemtoReg_EXMEM, RegWrite_EXMEM})
);
```

MEMWB:



```
Pipe_Reg #(.size(32*2 + 5*1 + 2*1 + 1*1)) MEMWB(
        .clk_i(clk_i),
        .rst_n(rst_n),
        .data_i({WriteDataMemory_EXMEM, MemtoReg_EXMEM,
                DataMemory_out, WriteReg_addr_EXMEM, RegWrite_EXMEM}),
        .data_o({WriteDataMemory_MEMWB, MemtoReg_MEMWB,
                DataMemory_out_MEMWB, WriteReg_addr_MEMWB, RegWrite_MEMWB})
);
```

2. Compared with lab4, the extra modules:

I added an additional module called ‘Pipe_Reg.v’ for my pipeline registers.

3. Explain your control signals in sixth cycle (both test patterns CO_P5_test_data1 and CO_P5_test_data2 are needed):

Control Signals:

CO_P5_test_data1	CO_P5_test_data2
MemtoReg: 0 RegWrite: 1 MemRead: 0 MemWrite: 0 RegDst: 1 PCSrc: 0 AluOP: 010 ALUSrc: 0 Branch: 0	MemtoReg: 0 RegWrite: 1 MemRead: 0 MemWrite: 0 RegDst: 0 PCSrc: 0 AluOP: 011 ALUSrc: 1 Branch: 0

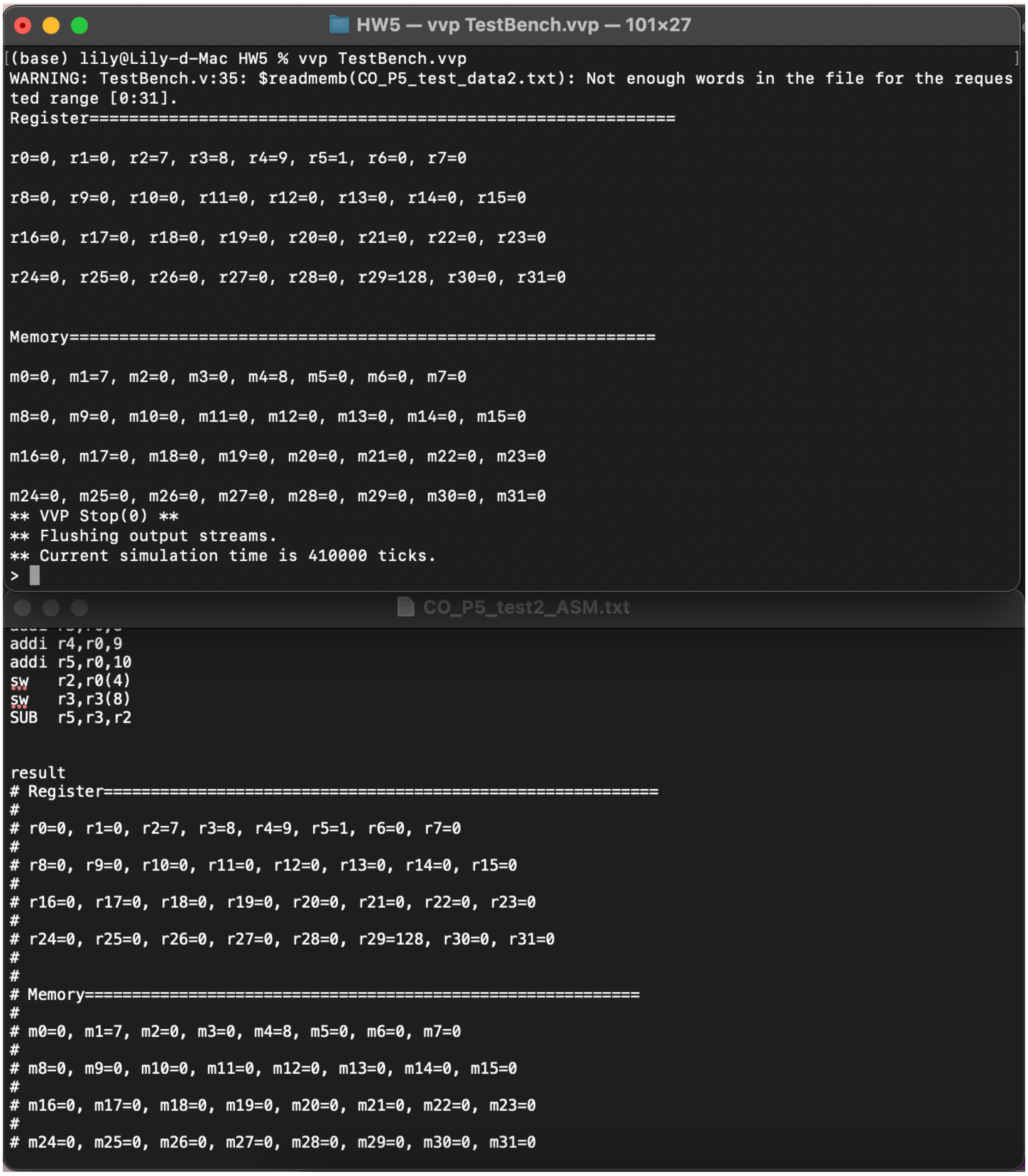
4. Problems you met and solutions:

Connecting all the wires were really difficult, especially keeping track of all the different wires and adding up its size. The Write Back stage also got me debugging for a while. Moreover, I did not code the Decoder correctly last time (I forgot to do something with sw and lw) so I had to fix it this time and it took a really long time for me to notice that the problem happened in Decoder instead of inside the Pipeline CPU.

5. Summary:

In this project, I learned how to turn a single cycle CPU into a pipeline CPU and I also learned how to make a simple register that stores data. Additionally, I passed both of TAs tests as shown below.

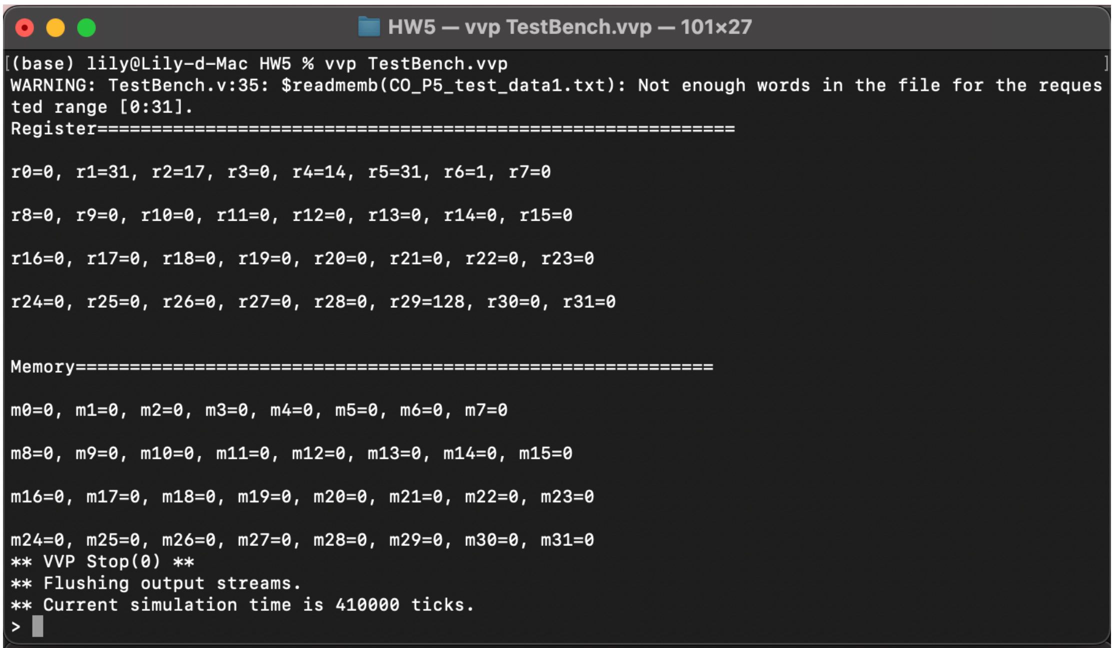
Test 1:



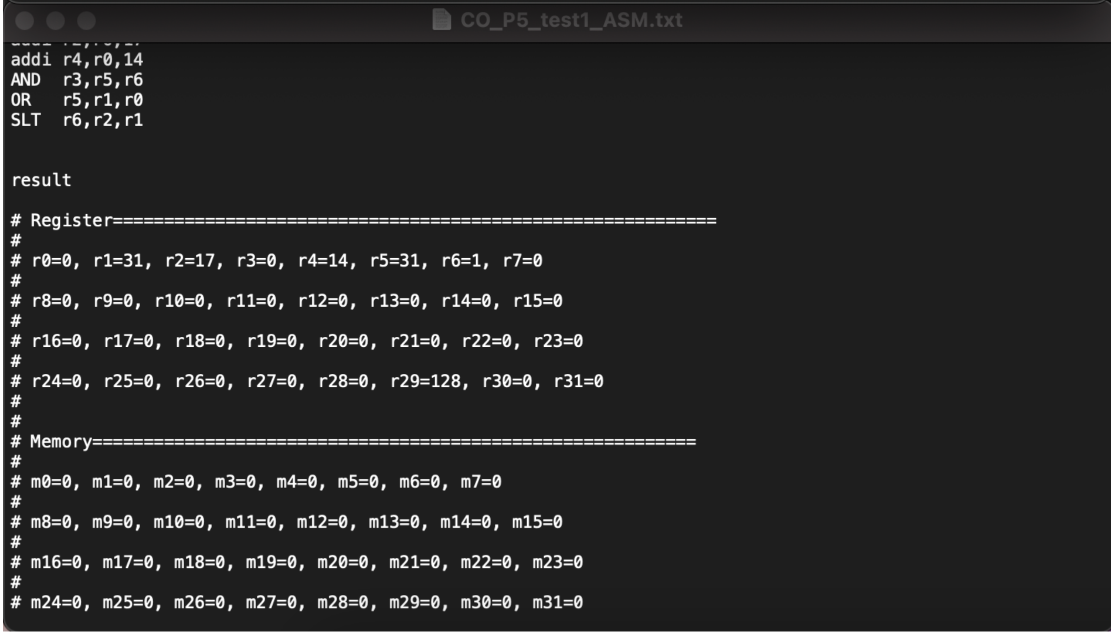
```
[(base) lily@Lily-d-Mac HW5 % vvp TestBench.vvp
WARNING: TestBench.v:35: $readmemb(CO_P5_test_data2.txt): Not enough words in the file for the reques
ted range [0:31].
Register===========================================================

r0=0, r1=0, r2=7, r3=8, r4=9, r5=1, r6=0, r7=0

r8=0, r9=0, r10=0, r11=0, r12=0, r13=0, r14=0, r15=0

r16=0, r17=0, r18=0, r19=0, r20=0, r21=0, r22=0, r23=0

r24=0, r25=0, r26=0, r27=0, r28=0, r29=128, r30=0, r31=0


Memory=============================================================

m0=0, m1=7, m2=0, m3=0, m4=8, m5=0, m6=0, m7=0

m8=0, m9=0, m10=0, m11=0, m12=0, m13=0, m14=0, m15=0

m16=0, m17=0, m18=0, m19=0, m20=0, m21=0, m22=0, m23=0

m24=0, m25=0, m26=0, m27=0, m28=0, m29=0, m30=0, m31=0
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 410000 ticks.
> 
```

```
addi r4,r0,9
addi r5,r0,10
sw   r2,r0(4)
sw   r3,r3(8)
SUB  r5,r3,r2


result
# Register============================================================
#
# r0=0, r1=0, r2=7, r3=8, r4=9, r5=1, r6=0, r7=0
#
# r8=0, r9=0, r10=0, r11=0, r12=0, r13=0, r14=0, r15=0
#
# r16=0, r17=0, r18=0, r19=0, r20=0, r21=0, r22=0, r23=0
#
# r24=0, r25=0, r26=0, r27=0, r28=0, r29=128, r30=0, r31=0
#
#
# Memory============================================================
#
# m0=0, m1=7, m2=0, m3=0, m4=8, m5=0, m6=0, m7=0
#
# m8=0, m9=0, m10=0, m11=0, m12=0, m13=0, m14=0, m15=0
#
# m16=0, m17=0, m18=0, m19=0, m20=0, m21=0, m22=0, m23=0
#
# m24=0, m25=0, m26=0, m27=0, m28=0, m29=0, m30=0, m31=0
```

Test 2:



```
(base) lily@Lily-d-Mac HW5 % vvp TestBench.vvp
WARNING: TestBench.v:35: $readmemb(CO_P5_test_data1.txt): Not enough words in the file for the reques
ted range [0:31].
Register===========================================================

r0=0, r1=31, r2=17, r3=0, r4=14, r5=31, r6=1, r7=0

r8=0, r9=0, r10=0, r11=0, r12=0, r13=0, r14=0, r15=0

r16=0, r17=0, r18=0, r19=0, r20=0, r21=0, r22=0, r23=0

r24=0, r25=0, r26=0, r27=0, r28=0, r29=128, r30=0, r31=0


Memory=============================================================

m0=0, m1=0, m2=0, m3=0, m4=0, m5=0, m6=0, m7=0

m8=0, m9=0, m10=0, m11=0, m12=0, m13=0, m14=0, m15=0

m16=0, m17=0, m18=0, m19=0, m20=0, m21=0, m22=0, m23=0

m24=0, m25=0, m26=0, m27=0, m28=0, m29=0, m30=0, m31=0
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 410000 ticks.
> 
```



```
addi r4,r0,14
AND  r3,r5,r6
OR   r5,r1,r0
SLT  r6,r2,r1


result

# Register===========================================================
#
# r0=0, r1=31, r2=17, r3=0, r4=14, r5=31, r6=1, r7=0
#
# r8=0, r9=0, r10=0, r11=0, r12=0, r13=0, r14=0, r15=0
#
# r16=0, r17=0, r18=0, r19=0, r20=0, r21=0, r22=0, r23=0
#
# r24=0, r25=0, r26=0, r27=0, r28=0, r29=128, r30=0, r31=0
#
#
# Memory=============================================================
#
# m0=0, m1=0, m2=0, m3=0, m4=0, m5=0, m6=0, m7=0
#
# m8=0, m9=0, m10=0, m11=0, m12=0, m13=0, m14=0, m15=0
#
# m16=0, m17=0, m18=0, m19=0, m20=0, m21=0, m22=0, m23=0
#
# m24=0, m25=0, m26=0, m27=0, m28=0, m29=0, m30=0, m31=0
```