Hilmi Taşkın
2232700

EE446 Experiment 3 Preliminary Work

1. Datapath

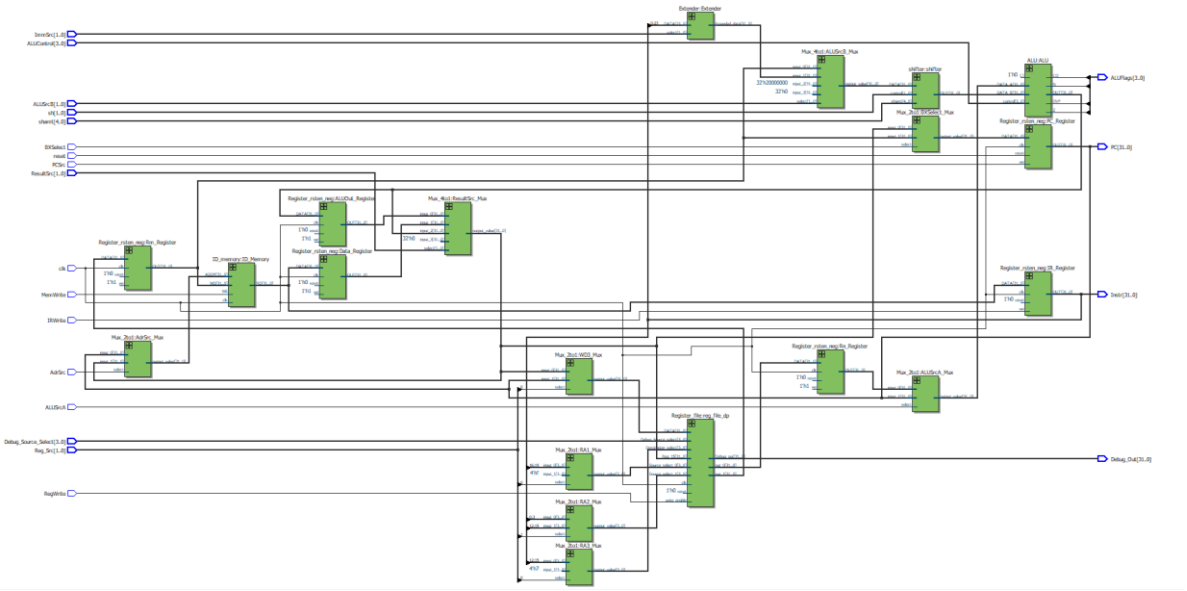

Figure 1. RTL view of Datapath.

In order to implement shifting operation in my computer I put a combinational shifter right after the ALUSrcB multiplexer. This allows me to shift both the value written in Rm and the extended immediate. If I don't want to shift I just give shamt as zeros from the controller. Shifter control is also managed in the controller. For BL operation I added a mux at A3 input of the register file to choose 14 as the destination register when branch and link operation is being handled. Also I connected another mux at WD3 input of the register file in order to choose between result and PC+4. The control signal for these muxes are RegSrc[0] which is the control signal for Branch. Therefore, I didn't need additional control signals for these muxes. For BX operation I put a mux before the PC register which chooses between result and Rm. I connected Rm output after the register A directly to this mux. When BXSelect is 1 mux chooses Rm and PC is written with the content of Rm.

Hilmi Taşkın
2232700

2. Controller

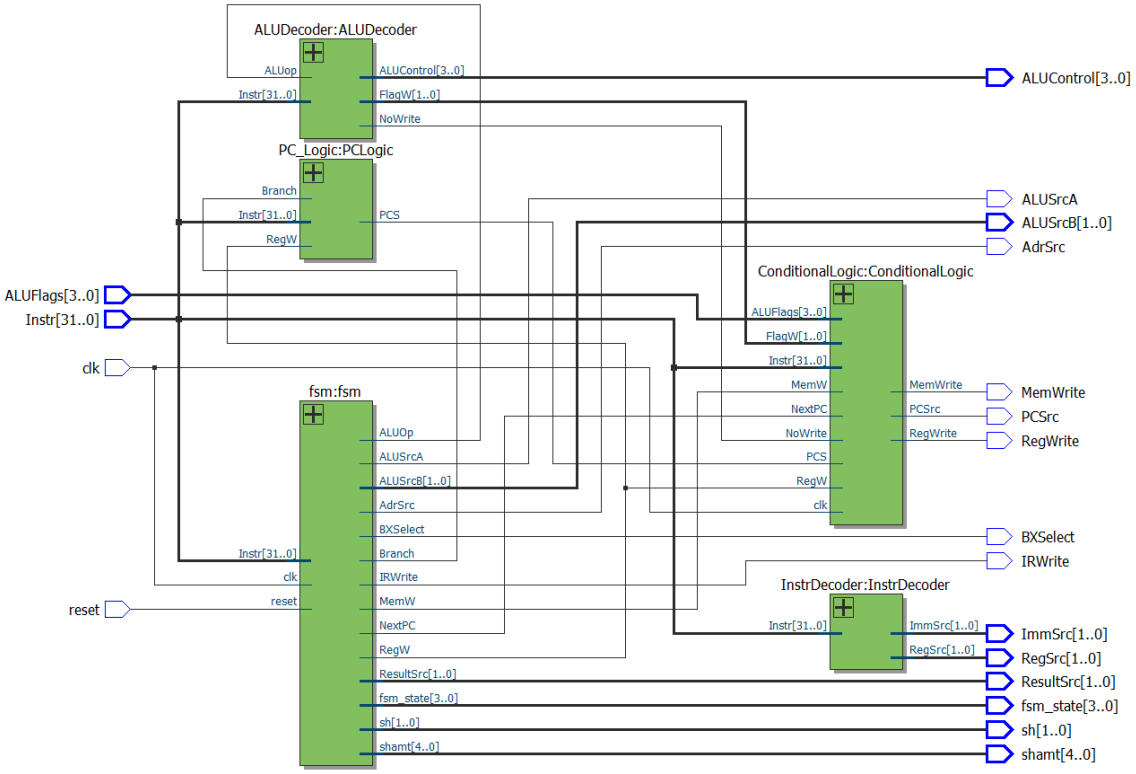

Figure 2. RTL view of Controller.

For shifter control I used 2'b00 when shifter is not used, 2'11 for rot and Instr[6:5] for data processing register operations. For shamt I used 5'b00000 for operations that do not require shift, {Instr[11:8], 1'b0} for rot and Instr[11:7] for data processing register operations. For BL I used two extra muxes with control signals RegSrc[0] because these muxes should take PC+4 and 14 only when there is branch. The branch operation is not affected by these signals because RegWrite signal is only active on the BL cycle. BXSelect signal which chooses between result and Rm is only 1 when the cycle is BX. For the CMP operation we also need a NoWrite signal in the controller between FSM and Conditional Unit which prevents writing on registers when the data processing operation is SUB during CMP.

Hilmi Taşkın
2232700

3. Testbench Results

```
************** Positive Clock Edge: 63 ********************
************ Performance Model / DUT Data  **************
PC:0x10          PC:0x10
Register0: 0x330         0x330
Register1: 0x13          0x13
Register2: 0x26          0x26
Register3: 0x2           0x2
Register4: 0x4c          0x4c
Register5: 0xa           0xa
Register6: 0x80000002    0x80000002
Register7: 0xfffffff8    0xfffffff8
Register8: 0x26          0x26
Register9: 0x0           0x0
Register10: 0x0          0x0
Register11: 0x0          0x0
Register12: 0x0          0x0
Register13: 0x0          0x0
Register14: 0x40         0x40
Register15: 0x14         0x14
Multi_cycle_test passed
*************************************************************************************
** TEST                              STATUS  SIM TIME (ns)  REAL TIME (s)  RATIO (ns/s) **
*************************************************************************************
** Multi_Cycle_Test.Multi_cycle_test   PASS      651000.00          1.47     443733.79  **
*************************************************************************************
** TESTS=1 PASS=1 FAIL=0 SKIP=0                 651000.00          1.65     393452.50  **
*************************************************************************************
```

Figure 3. Screenshot of the testbench results.

4. Drawing of the Datapath

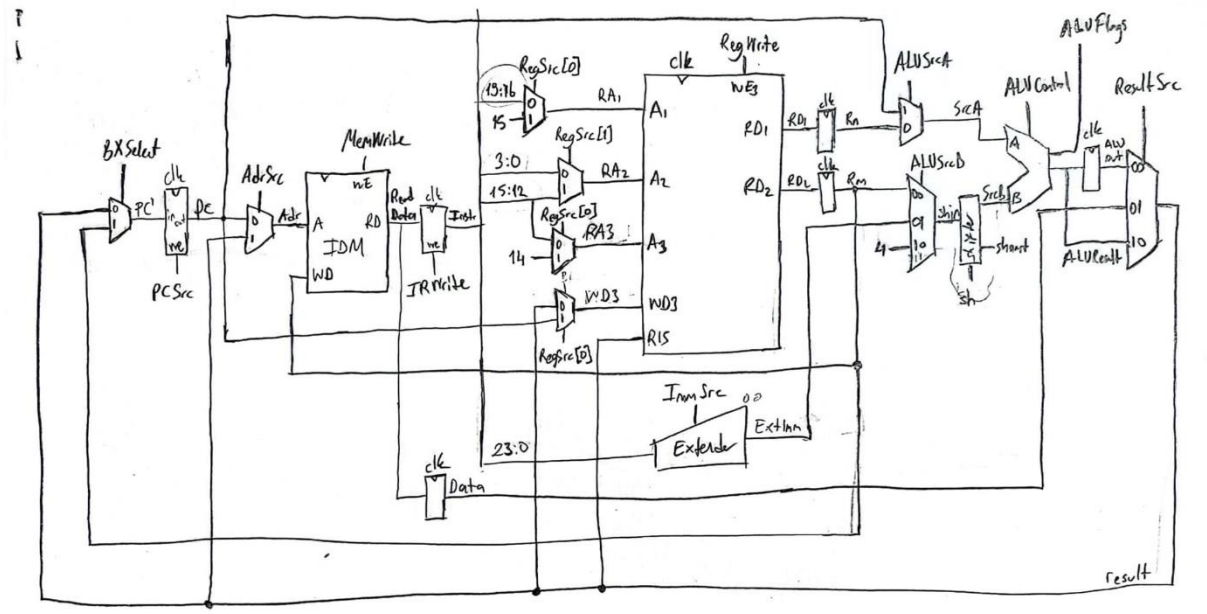

Figure 4. Hand drawing of the datapath I used.