**ESE 356 Digital System Specification and Modeling**
**Project 1: MINI RISC Processor**
**Phase 2 (Final Version) Requirement**

**Due on 10/6/2020**

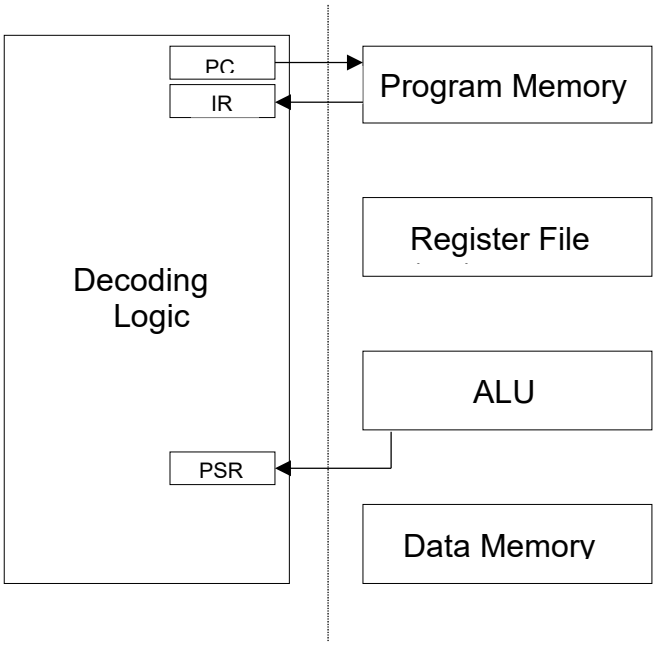
**Total Points (25): No late submission (submit the file by midnight of the due date)**
*****Submit Program listing by September 24 midnight.**

## 1. Final Phase Specification

Revise your data-path if necessary.

Based on the control signals defined in the control signal table, create a controller module. All input and output ports must match the ports defined in your data-path modules.

For PSR, Z, N, C, F bits will be implemented.




Controller design strategy will be discussed in the lecture.

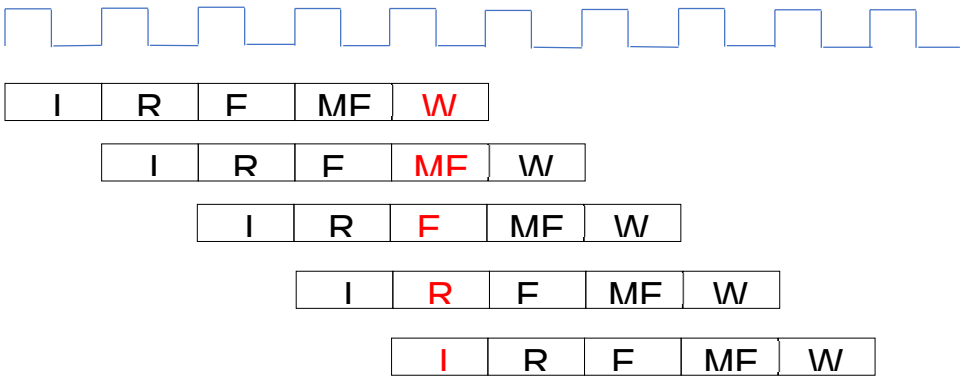
Obtain a table for indicating all control signals necessary to control the data-path.

Table 1: Control Signals from Controller to Data-path (Revised from Phase 1)

| Instruction | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_{ARITH}$ | $RF_{R1}$ | $RF_{R2}$ | $RF_W$ | $DM_R$ | $DM_W$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADD | 0 | 0 | 1 | 0 | 1 | X | 1 | 110 | 1 | 1 | 1 | 0 | 0 |
| ADDI | | | | | | | | | | | | | |
| …. | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |

5-stage pipeline execution is assumed in this project.

IF: Instruction Fetch – At the rising edge of the clock cycle, an instruction is read out from the program memory based on the address located by the program counter (PC).
At the next rising edge, this instruction is latched by the instruction register (IR).




RD: At the rising edge of the clock cycle, the IR content and the values in the PSR are used to decode an instruction. All control signals are generated, and some will be stored in the shift registers to be used later.

EXE: Execution – ALU performs the operation. The ALU also updates PSR content (You need to revise your current data-path modules)

MEM: Memory access

WB: At the falling edge of the clock cycle, the result will be written

Obtain a table for indicating all control signals necessary to control the data-path. Revise the table for control signals including the stage index (Indicate when specific control signals are used in the pipeline stage)

Table 2: Pipeline Stage Dependent Control Signal Generation

| Instruction | Stage | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_{ARITH}$ | $RF_{R1}$ | $RF_{R2}$ | $RF_W$ | $DM_R$ | $DM_W$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADD | Overall | 0 | 0 | 1 | 0 | 1 | X | 1 | 110 | 1 | 1 | 1 | 0 | 0 |
| | RD | | | | | | | | | 1 | 1 | | | |
| | EXE | 0 | 0 | 1 | 0 | 1 | X | 1 | 110 | | | | | |
| | MEM | | | | | | | | | | | | 0 | 0 |
| | WB | | | | | | | | | | | 1 | | |
| ADDI | Overall | | | | | | | | | | | | | |
| | RD | | | | | | | | | | | | | |
| | EXE | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |

Once the controller modules are completed, integrate with the data-path modules.

Make sure the processor has reset capability.

**2. Verification and Simulation**

Testing:

Program 1:
// DM[10] = 36;
// DM[12] = 12;
// DM[14] = -24;
// All other locations are initialized to ZEROS

// RF[R3] = 10;
// RF[R5] = 12;
// RF[R7] = 14;
// RF[R8] = 12;
// All other locations are initialized to ZEROS

// Program must start from address 10
// Addresses 0, 2, 4, 6, 8 have NOP
// Make sure the processor has reset capability.

```
10      LOAD   R2, R3;        // DM[R3] -> RF[R2]          // R2 has Integer 36
12      LOAD   R4, R5;        // DM[R5] -> RF[R4]          // R4 has Integer 12
14      ADDI   17, R8;        // RF[R8] + 17 -> RF[R8]     // R8 is changed from 12 to 29
16      NOP;
18      ADD    R2, R3;        // RF[R3] + RF[R2] -> RF[R3] // R3 has integer 46
20      LOAD   R6, R7;        // DM[R7] -> RF[R6]          // R6 has Integer -24
22      SUBI   15, R8;        // RF[R8] - 15 -> RF[R8]     // R8 has Integer 14
```

```
24      LOAD   R2, R5;        // DM[R5] -> RF[R2]             // R2 has Integer 12
26      ADD    R8, R2;        // RF[R2] + RF[R8] -> RF[R2]    // R2 has Integer 26
28      STOR   R8, R3;        // RF[R8] -> DM[R3]             // DM[10] has Integer 14
30      STOR   R4, R5;        // RF[R4] -> DM[R5]             // DM[12] has Integer 48
32      NOP
```

Program 2:

```
// RF must be initialized so that following integers are stored
// RF[R1] = 39;
// RF[R2] = -25;
// RF[R3] = 14;
// RF[R4] = -50;
// RF[R5] = 70;
// RF[R6] = 6;
// RF[R7] = 7;
// RF[R8] = 8;
// RF[R10] = 32;

// Program must start from address 10
// Addresses 0, 2, 4, 6, 8 have NOP
10      ADD    R2, R1;        // RF[R1] + RF[R2] -> RF[R1]    // Address 10, R1 has Integer 3
12      ADD    R3, R1;        // RF[R1] + RF[R3] -> RF[R1]    // R1 has Integer 6
14      ADD    R4, R1;        // RF[R1] + RF[R4] -> RF[R1]    // R1 has Integer 10
16      ADD    R5, R1;        // RF[R1] + RF[R5] -> RF[R1]    // R1 has Integer 15
18      CMPI   10, R1;
20      NOP
22      NOP
24      JNE    R10;                                          // Jump is Z = 0 in PSR
26      NOP
28      ADD    R6, R1;        // RF[R1] + RF[R6] -> RF[R1]
30      ADD    R7, R1;        // RF[R1] + RF[R7] -> RF[R1]
32      ADD    R8, R1;        // RF[R1] + RF[R8] -> RF[R1]
34      NOP
36      NOP
```

Program 3:

Repeat the same program by replacing CMPI 10, R1 to CMPI 15, R1

**3. Submission Requirements**

- Source codes for controller, data-path, top main and necessary test-bench codes
- Data-path diagram and control signal tables
- Program list (16-bit opcodes)
- Verification/Simulation results: Execution of test programs
- Summary report (1-2 pages)

Submission through electronic files (zip version)
The report grading will be based on 1. Clarity of the report, 2. Completeness of the results.

**4. Grading**

1. Source Codes (6) – PDF formats
2. Datapath Diagram (2) – Drawings
3. Datapath/Controller Port Specification/Explanation (2) – Input/Output port list
4. Pipeline Control Signal Table (2)
5. Successful Compilation (2) – Screen capture of compilation report
6. Program Lists (3) – 16-bit opcodes (Early submission)
7. Test Program Execution (8) – Print out: PSR contents, DM contents, RF contents (Only affected Initial states and final states)