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

Due on 9/22/2020

Total Points (15): No late submission (submit the file by midnight of the due date)

1. Initial Phase Specification

Determine the complete data-path for the processor. The data-path must correspond to the instruction sets that we are implementing. Obtain a diagram of the data-path including the data memory, program memory, register file, and arithmetic logic. The inputs to data memory are 16-bit address, 16-bit data. The output from the data memory is 16-bit data. The program memory has 16-bit address input and 16-bit data output. The data memory may have write/read control signal as an input to indicate the operation of memory access.

The register file has two 4-bit addresses (can access two operands simultaneously). One of these address can be used for writing the data back to the register file. The register file also has two 16-bit data output and one 16-bit data input. There are 16 registers in the register file (R0 is pre-wired to have all zeros). R0 should not be used in the program.

Obtain a table for indicating all control signals necessary for each instruction. First identify the resources used in the data-path. Since each resources are used differently for each instruction, a table is necessary for proper operation of the data-path. In the modeling process, these control signals are considered as inputs. Example of control signal table is illustrated below.

Example:

Instruction	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_{ARITH}
ADD	0	0	1	0	1	X	1	110X
ADDI								
….								

Complete the specification for Register File, ALU, Program Memory, Data Memory. Each module must be parameterizable (i.e., program memory, data memory, and register file contents can be

initialized during elaboration stage). Then write a data path module (top module) integrating all of these modules.

2. Verification and Simulation

Write following modules:

- Register Files (rf.cpp and rf.h)
  - 2 read address input ports, 1 write address input port
  - Read/write enable input port
  - Clock input port
  - 2 read data output ports, 1 write data input port
- Data Memory (dm.cpp and dm.h)
  - 1 address input port
  - 1 data input port, 1 data output port
  - Read/Write enable input port
- Program Memory (pm.cpp and pm.h)
  - 1 address input port
  - 1 data output port
- Arithmetic Logic (al.cpp and al.h)
  - Input ports corresponding to the control signal table
  - Output ports corresponding to the processor status
  - The other inputs/outputs will be discussed in the lecture
- Data Path (dp.cpp and dp.h) (Skeleton only. Complete version in Phase 2)
  - Inputs/outputs will be discussed in the lecture
- Controller (cnt.cpp and cnt.h) (Skeleton only. Complete version in Phase 2)
  - Inputs/outputs will be discussed in the lecture

In each of these modules (dm, pm, rf, al), processes, inputs and outputs must be specified, and parameter initialization should be handled.

Data Path should instantiate rp, dm, pm, al modules in its constructor.

Controller module input and output ports should match.

3. Submission Requirements

- Data-path diagram and control signal table
- Source codes submodules (dm, pm, rf, al)
- Port Specification of four modules (input ports and type, output ports and type)
- Verification of dm, pm, rf, al modules
- 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 (4)
2. Control Signal Table (3)
3. Datapath diagram (2)
4. Port Specification (3)
5. Module Verification (3)