

//First draft

Evaluation of the project implemented in Digital/Quartus:

Stage I: design of the instruction set, data path, and microinstructions – maximum 60 points,

Stage II: design of the simplified control unit and instruction fetch unit – maximum 25 points,

Stage III: design of the complete control unit and memory manager; implementation and simulation of the developed CPU components and execution of a sample processor program using Digital or Quartus II – 15 points.

To receive points for Stage II, the elements associated with Stage I must be designed correctly. Similarly, to receive points for Stage III, the elements associated with Stage II must be designed correctly. In Stage III, the CPU design MUST compile without errors in Digital / Quartus II, and a representative program performing calculations on the CPU must be prepared (the program should be provided as a MIF or HEX file within the project).

Evaluation of the project implemented as a software simulation:

Stage I: design of the instruction set and microinstructions, logic implementing data path functions – maximum 60 points,

Stage II: implementation of the CPU and memory simulator (including a simplified GUI) – maximum 25 points,

Stage III: implementation of the complete simulator including a sample application – 15 points.

To receive points for Stage II, the elements associated with Stage I must be designed correctly. Similarly, to receive points for Stage III, the elements associated with Stage II must be designed correctly.

//Second draft (not formatted)

Computer Architecture Project – Requirements and Evaluation Criteria

1. Project Objective

The objective of the project is to design and implement a microprocessor and its associated components in accordance with the specification and constraints provided in the project introduction and the requirements assigned to each team.

The processor must be implemented as a hardware design using Digital or Quartus II.

The project is divided into three stages. Each stage builds upon the results of the preceding stage. Students are expected to document their design decisions, explain the proposed solutions, and demonstrate the correct operation of the developed processor.

2. Stage I – Processor Architecture and Microinstruction Design (60 points)

Students must propose and document the architecture of the processor, including its instruction set, instruction encoding, internal data paths, interconnections between processor components, and microinstruction sequences implementing individual instructions.

The design must include:

Processor architecture: the organization of the processor, its functional components, registers,

buses, and other internal resources. Instruction set: the definition of the supported instructions, including their operation, operands, addressing modes, and effects on the processor state.

Instruction encoding: the format and encoding of machine instructions, including the organization of instruction fields and the rules for decoding instructions. The instruction word must support variable-length encoding.

Data path and interconnections: the organization of the data path and the connections between registers, the ALU, buses, memory interfaces, and other processor components, as applicable to the assigned project.

Microinstruction design: the definition of microinstructions and the microinstruction sequences required to execute the supported machine instructions.

Processor state and flags: the definition of the processor state, including the registers and status flags, and the rules governing their updates during instruction execution.

The proposed architecture and encoding must be consistent with the assigned project specification and all team-specific requirements.

3. Stage II – Control Unit Design (25 points)

Students must propose a design for the processor control unit responsible for coordinating the execution of instructions and controlling the operation of the data path.

The design must include:

The organization and operating principles of the control unit. The method of decoding machine instructions and generating the required control signals. The method of sequencing microinstructions and controlling the execution of individual instructions. The cooperation between the control unit, the data path, and the instruction fetch unit. The handling of the processor state and status flags, where applicable.

For a hardware implementation, students must design the simplified control unit and instruction fetch unit.

The proposed control unit must be consistent with the architecture and microinstruction definitions established in Stage I.

4. Stage III – Complete Implementation and Testing (15 points)

Students must complete the processor design and demonstrate its correct operation.

Using Digital or Quartus II, this stage must include:

Implementation of the complete control unit, instruction fetch unit, and memory management components, as required by the assigned specification. Integration of the processor components, including the data path, control unit, registers, and memory interfaces. Implementation of the required microprogram. Compilation of the complete processor design without errors in Digital or Quartus II. Preparation of a short test program that performs representative calculations and demonstrates the functionality of the implemented processor. Inclusion of the test program in the project files in MIF or HEX format, as appropriate for the selected tool and memory configuration. The test program must demonstrate the correct execution of representative instructions and the interaction between the relevant processor components.

The project report must include the test program's pseudocode and assembly-language source code, together with an explanation of the expected results. The implementation must be tested, and the results must be presented in the report.

5. General Processor Requirements

The processor and its associated components must comply with the common requirements specified in the project introduction and the additional requirements assigned to each team.

Unless explicitly stated otherwise in the team-specific specification, the processor must:

Support addressing of 4096 words of main memory. Support the default (implicit), immediate, and direct addressing modes. Support memory segmentation, with separate code and data segments. Include an appropriate number of segment registers. Include a program counter (PC) register that can be read but not directly modified by ordinary instructions. Update the status flags after the execution of the relevant instructions. Use a variable-length machine instruction format. Any additional requirements concerning the number and types of registers, bus widths, supported instructions, addressing modes, memory organization, and other architectural parameters are defined in the team-specific project assignment.

6. Dependencies Between Stages and Assessment Rules

The project is assessed on a scale of 100 points:

Stage I – Processor architecture and microinstruction design: 60 points. Stage II – Control unit design: 25 points. Stage III – Complete implementation and testing: 15 points.

Each stage depends on the successful completion of the preceding stage. Points for Stage II can be awarded only if the relevant Stage I design elements are correct. Similarly, points for Stage III can be awarded only if the relevant Stage II design elements are correct.

The maximum number of points for a stage is not awarded automatically upon submission. The work must meet the requirements specified for that stage.

The assessment considers the correctness, completeness, and consistency of the design, its compliance with the assigned specification, and the evidence provided to demonstrate that the implemented or simulated processor operates correctly.

7. Project Documentation and Submission

The submitted project must include the materials necessary to evaluate the design and verify its operation. These materials must include, as applicable:

A description of the processor architecture and its components. The instruction set and machine instruction encoding. Data path diagrams and component interconnections. The microinstruction format and microinstruction sequences. The control unit design and instruction execution mechanism. The complete hardware design. The test program in pseudocode and assembly language, together with an explanation of its operation and expected results. The required project files, including the MIF or HEX file for hardware implementations where applicable. Evidence of successful compilation or execution and the results of the test program.

All implementation and documentation details must be consistent with the specification and the requirements assigned to the team.

8. Project is for two members team

9. There will be presentation and defense of the results of the project at least at the last classes.

Project topics:

operation for all teams:

Adding REG, st8

Adding REG, REG

Subtract REG, st8

Subtract REG, REG

Move REG, REG

Move REG, MEM

Move MEM, REG

Operation with memory - all possible addressing modes

Topic	General purpose registers	Specialised registers	Magistral size	Orders
1	8	0	8	CMP JE AND
2	6	2	16	OR JMP INC
3	4	4	8	DEC ADC NEG
4	10	0	8	XLAT [adr] NOT INC

