2303IT302 - Computer Organization and Architecture
Case Study Projects

Guidelines: Each project is to be done by a *group of students*. The projects are to be implemented in any of the programming language. Some projects are theoretical work. Each project work carries documentation and powerpoint presentation The project report/presentation structure is:

1. **Title**
2. **Objective:** (The objective describes the goal of the project work.)
3. **Theory:** (The theory is formal design comprising descriptions, essential mathematics, formulas, drivations, etc.)
4. **Design:** (The design part comprises flow-charts, algorithms, tables, diagrams, derivations, etc.)
5. **Implementation:** (The implementation is description of functional modules of code, hierarchical relationship, coding with built-in documentation, list of system requirements, like compilers, operating system, etc.)
6. **Debugging-Test-run:** (The Test-run and result part of the report contains detailed method of testing, assuring that the code is fool-proof and fully debugged.)
7. **Results analysis (if any):** (The Analysis part should discuss other aspects, like complexity of algorithms in terms of average and worst case complexity for time and space, robutsness of the approach used, finer technical details, etc.)
8. **Conclusion and Future Improvements:** (The conclusion and future aspect should summarize the project in brief, what improvements can be possible which could not be considered due to time limits, limitations (if any in the design and implementation), various applications of this design, etc.)
9. **Bibliography:** The bibliography section should provide the detailed list of references of books, journals, websites, conferences, and others in the standard accepted formats.

Note: Some of the projects may be done totally theoretical, and no coding be done. But, they should be exhaustive in mathematical and descriptive part.

List of Projects:

1. Design and computer architecture: Design a processor with minimum number of instructions, so that it can do the basic arithmatic and logic operations.
2. Design a serial interface to connect the 8085 micro-processor with a keyboard for that on pressing of enter key of keyboard, it can receive the characters typed.
3. Design and pipeline architecture for 2,3,4 stages pipeline.
4. Using microprogram as instructions directly: Consider that there is no "instruction set", no program counter (but microprogram counter), no instruction fetch in the normal sense. Your machine and "program" is THE microprogram itself. You have to add some fields into microprogram word such as : ADD R0, R1, R2 which hold the appropriate values.
5. Give a comparative study of 8085, 6502, and NSC micro-processors
6. Design a Turing machine using java, to implement basic operations of TM.
7. Write program to generate assembly code from prefix code.
8. Simulate a word multiplier.

9. Simulate a word divider.
10. Suggest a high speed addition method and logic for 4-bit addition.
11. Microprocessor-based automatic door opener.
12. Microprocessor based furnance temperature controller.
13. Microprocessor based water level controller in domestic water storage tank (when water goes above a height, it switches off the pump motor, and when it goes below some level, it switches ON the pump motor).
14. Microprocessor based automatic attendance recorder (make use of RFID: a unique for each student).
15. Microprocessor based ECG recorder.
16. Design and implement an arbitrary precision four function calculator.
17. Stack machine ISA : Design a stack machine, its instruction set must be stack oriented (no register!).
18. Implement quick sort using 8085 assembly language.
19. Implemet binary search using 8085 assembly language.
20. Implement matrix multiplication using 8085 assembly language.
21. Design a instruction set for a limited functionality machine having all instructions of 8-bits fixed length only, including opcode and operands.
22. Write/create a tool for bench-marking of a hardware (CPU).
23. Suggest and design a minimal cpu architecture for controlling the washing machine.
24. Simulate modern traffic controlsystem.
25. Project for piano sound generation.
26. Quantum Compters.
27. DNA Computers.
28. Construct an interpretor written in C language to interpret an assembly language based on the following basic instructions for a machine having only one register, which is accumulator, and all the operands are in memory:

Opcode, operand	comment
ADD X	Add memory location *x* into acc.
SUB X	Substract X from Acc.
MUL X	Multiply X with Axcc.
DIV X	Divide acc. by X.
AND X	And X with acc.
NOT X	Complement acc.
OR X	Or X with acc.
LD X	Load memory location X at acc.
ST X	Store acc. at memory location X

Assessment Pattern

Criteria	Marks
Problem Statement and System Analysis	05
Theoretical Concepts	05
Design and Implementation	10
Debugging/Result and Conclusion	10
Presentation Skill	10

Rubrics

Problem Statement and System Analysis	The problem statement is clear, precise, and well-defined. There is no ambiguity.	The problem statement is clear but may have minor ambiguities.	The problem statement is somewhat clear but lacks precision.	The problem statement is vague and unclear.
Theoretical Concepts	Requirements are thoroughly analysed and well-documented.	Requirements are analysed and documented adequately.	Requirements are somewhat analysed but lack thorough documentation.	Requirements are poorly analysed and documented.
Design and Implementation	System architecture is well-defined, detailed, and logical.	System architecture is defined and logical but lacks some details. System architecture is somewhat logical but lacks detail.	System architecture is somewhat logical but lacks detail.	System architecture is poorly defined and illogical.
Debugging/Result and Conclusion	Conducts a thorough feasibility study covering technical, economic, and operational aspects.	Conducts a feasibility study covering most aspects adequately.	Conducts a basic feasibility study with limited scope.	Conducts a poor or no feasibility study.
Presentation Skill	Content is thorough, accurate, and highly relevant. Information is well-organized and clearly supports the main points.	Content is accurate and relevant. Information is organized and supports the main points.	Content is mostly accurate but may lack relevance or organization. Main points are supported but not strongly.	Content is inaccurate, irrelevant, or disorganized. Main points are unclear or unsupported.