**List of Mini Projects for CSE360: Computer Architecture**

**Guidelines:**Mini project is to be done by a *group of three students*. The project is to be implemented in any of the programming language: Verilog or C/C++. Each project work carries documentation in the form of a project report to be submitted in printed form - A4 size with soft binding. The project report 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, derivations, etc.)
4. **Design:**(The design part comprises flowcharts, 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, robustness 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.

The report should have a front cover in the standard form, generally used for seminar/dissertation, giving project title, class, name of student, guide, name of Institution, year, and month & year of submission, all in standard acceptable formats.

**List of Mini Projects (each group will choose only one project and there should not be any conflict with other groups):**

1. Design a processor with minimum number of instructions, so that it can do the basic arithmetic and logic operations.
2. Design a serial interface to connect the 8085 micro-processor with a keyboard for that on pressing of entering key of keyboard, it can receive the characters typed.
3. Design pipeline architecture for various stage pipelines.
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 holds the appropriate values.
5. Write a program to generate assembly code from prefix code.
6. Suggest a high speed addition method and logic for 4-bit addition.
7. Simulate Cache policies/performance experimentation.
8. Simulate Cache coherence policies and compare performance.
9. Simulate Various Pipeline processors and compare performance (Superscalar, super pipeline, etc.).
10. Simulate Brach Prediction policies and compare performance.
11. Experiments on Graphics Processing Unit (GPU) Memory hierarchy and performance.
12. Experiments on Multicore Cache coherence and performance.
13. Experiment on shared memory and cache memory hit/miss and access performance and also overall execution performance due to memory access patterns, optimization, parallelism, etc.
14. Design a instruction set for a limited functionality machine having all instructions of 8-bits fixed length only, including opcode and operands.
15. Write/create a tool for bench-marking of hardware (CPU).
16. Suggest and design a minimal CPU architecture for controlling the washing machine.
17. Design a stack machine, its instruction set must be stack oriented (no register!)
18. Design and implement an arbitrary precision four function calculator.
19. Construct an interpreter 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 | Subtract X from Acc. |
|  |  |
| MUL X | Multiply X with Acc. |
|  |  |
| 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 |
|  |  |
|  |  |

**Note:** You are required to study about these projects, explore, think, and try to find out how they will be carried out. Make assumptions where details are not provided or go as per what is standard. Some of the projects may be done totally theoretical, and no coding to be done. But, they should be exhaustive in mathematical and descriptive part.