

Project A - Phase I

ECE 6913 Computer System Architecture

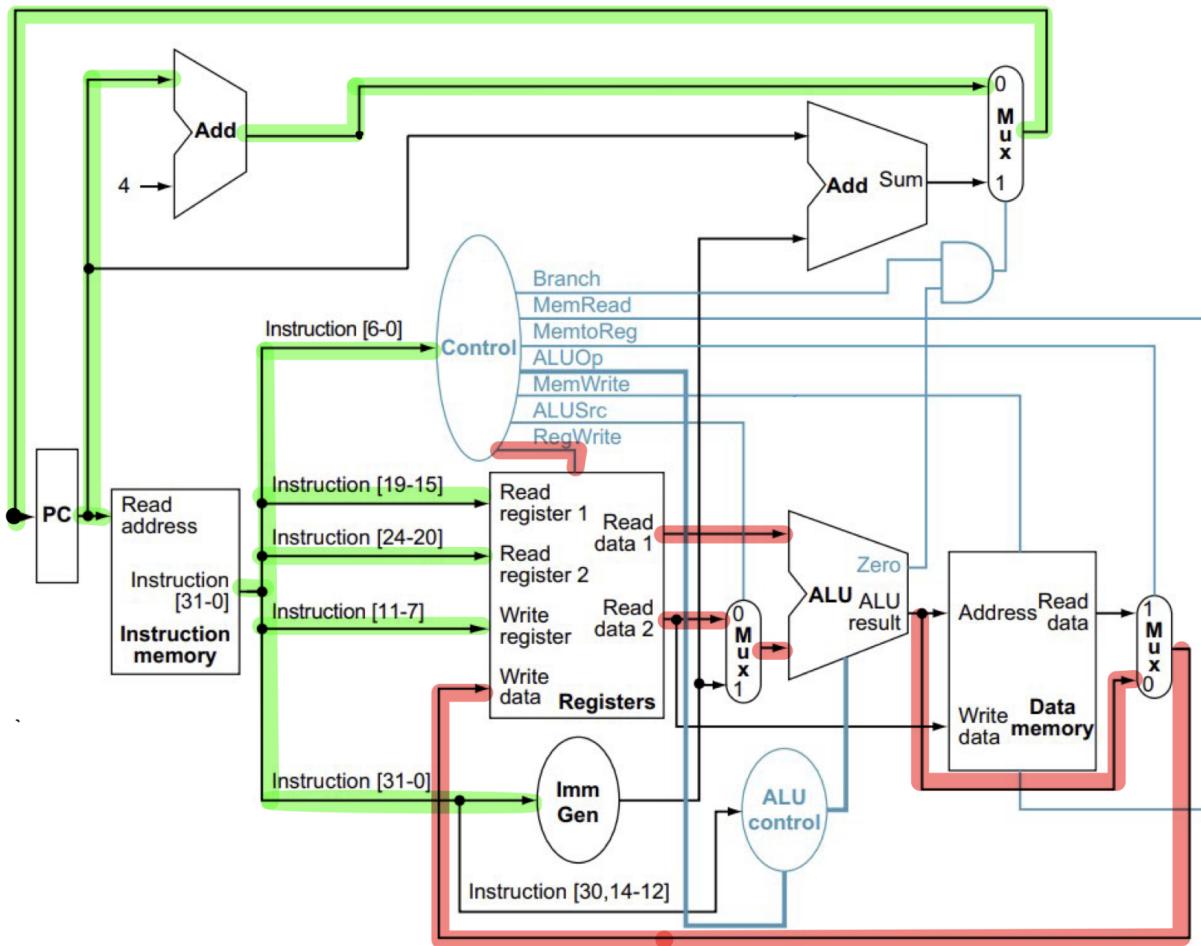
Jingjing He (jh8244@nyu.edu), Ziqi Gao (zg2346@nyu.edu)

Performance Metrics:

The CPI of single stage processor is 1.0. Total execution cycle depends on the input instructions.
Instruction per cycle of single stage processor is 1.0.

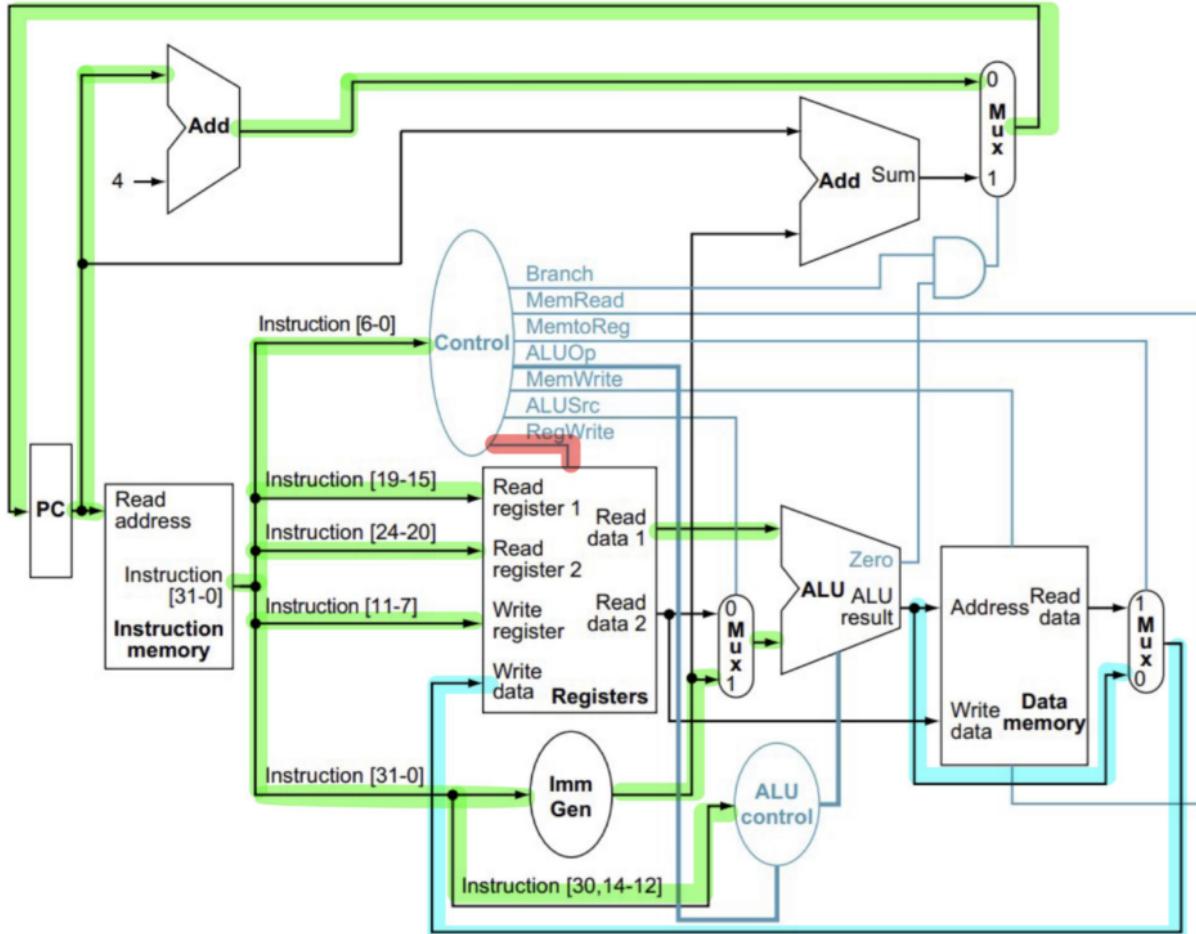
Schematic and Datapath for Single Cycle Processor

R-type



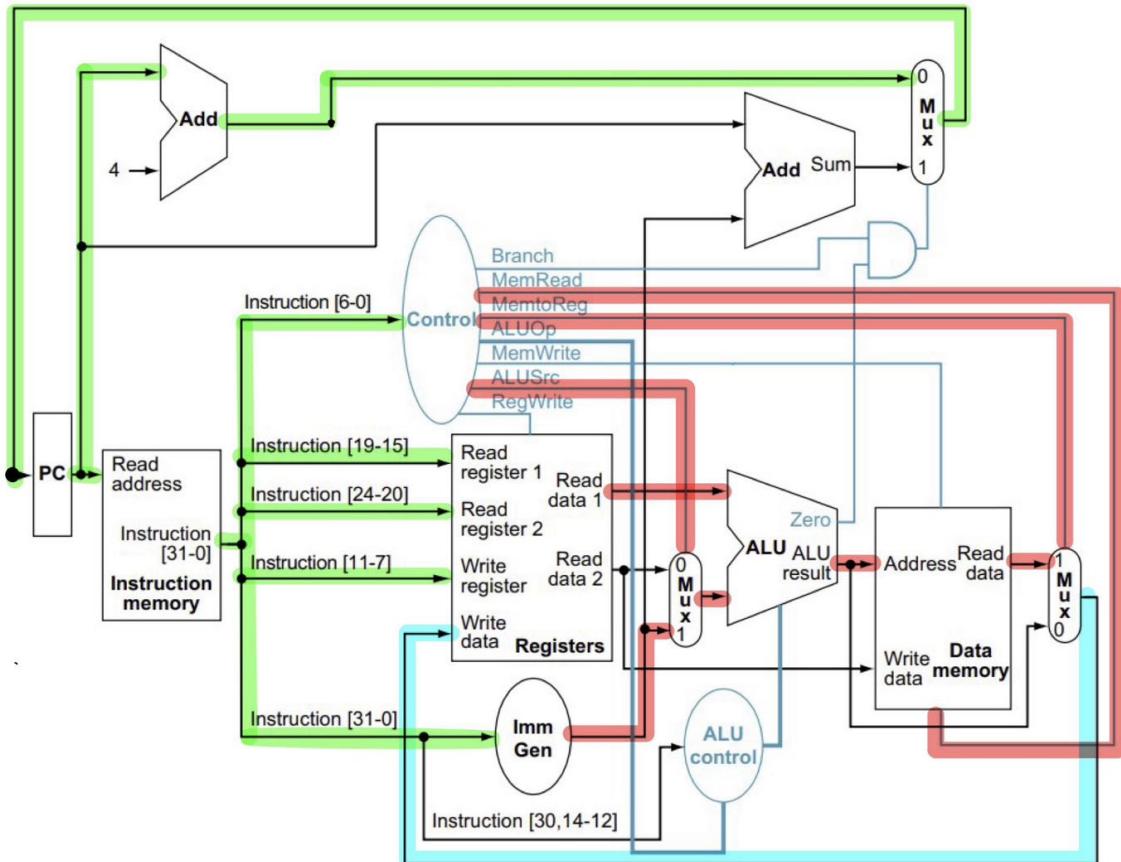
- Instruction is fetched from the instruction memory, and the PC is incremented.
- Two registers' values, are decoded and read from the register file (instruction[19-15], instruction[24-20]). ALUSrc, RegWrite are set in the control unit. ALUOp1 =1, ALUOp0 = 0
- ALU Control are set based on Instruction [14-12] (func3) with ALUOp1 =1 ALUOp0 = 0.
- The ALU operates on the data read from the register file, using ALU Control's output to generate the ALU function.
- The result from the ALU is written into the destination register (instruction[11-7]) in the register file.

I-type



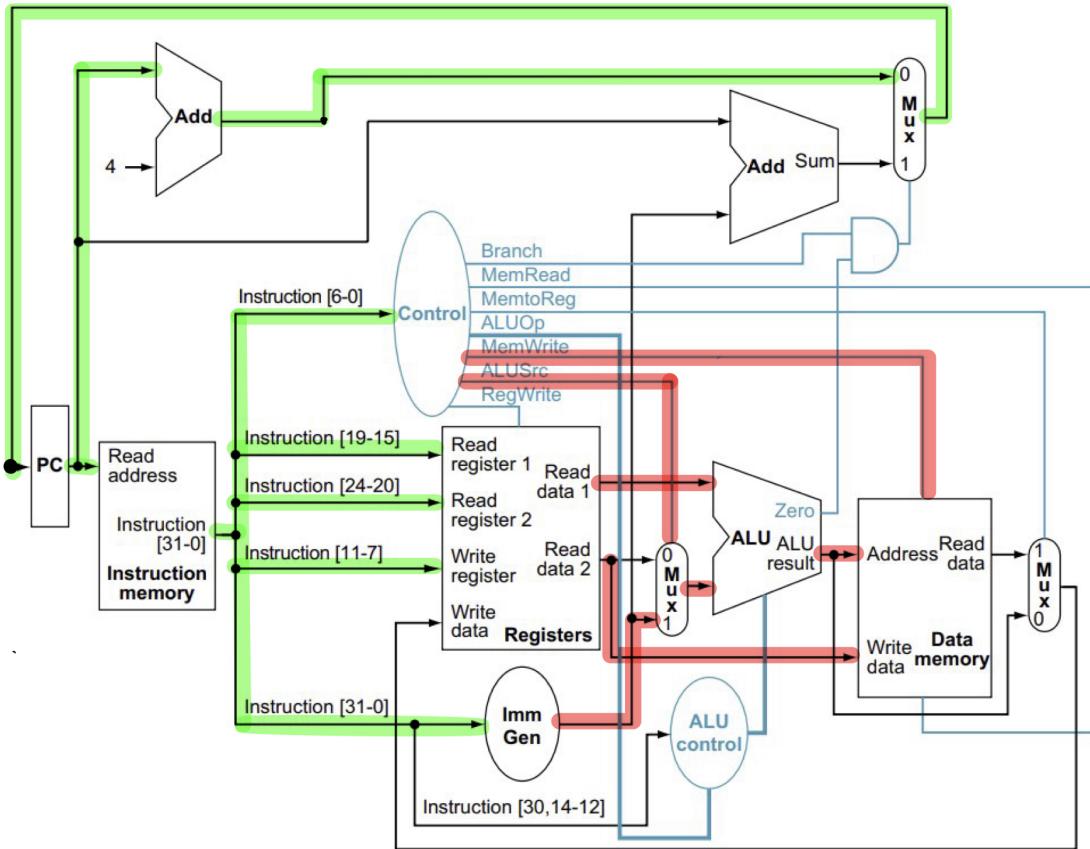
- Instruction is fetched from the instruction memory, and the PC is incremented.
- Instruction is decoded and the register (instruction[19-15]) is read from the register file. RegWrite are set in the control unit.
- ALU Control are set based on Instruction [14-12] (func3) with ALUOp1 =1 ALUOp0 = 0. And the immediate value is sign-extended.
- The ALU operates on the data read from the register file and the sign-extended immediate value, using ALU Control's output to generate the ALU function.
- The result from the ALU is written into the destination register (instruction[11-7]) in the register file.

LW:



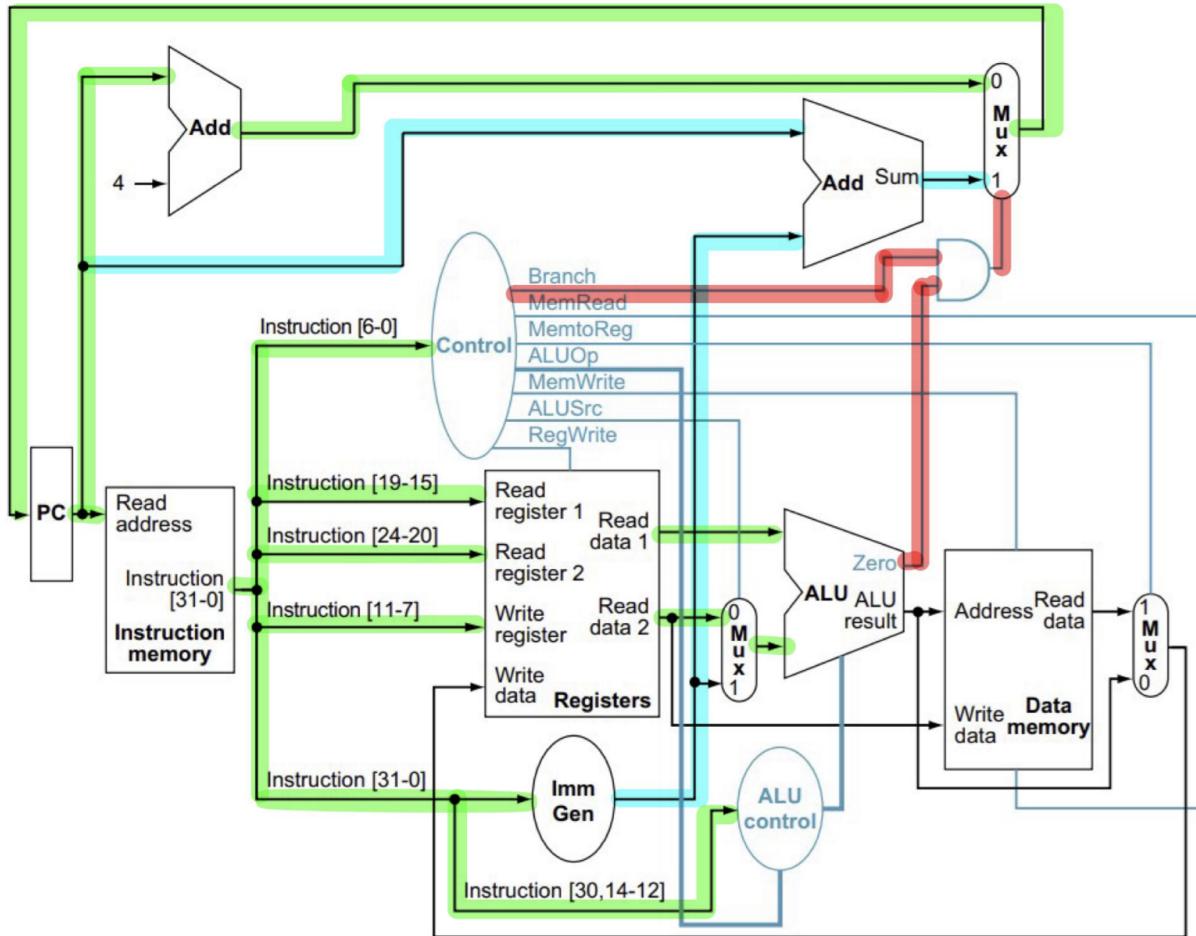
- Instruction is fetched from the instruction memory, and the PC is incremented.
- Instruction is decoded and the register (instruction[19-15]) is read from the register file. ALUSrc, RegWrite, MemtoReg, MemRead are set in the control unit. ALUOp1 = 0, ALUOp0 = 0
- ALU Control are set to relevant function based on Instruction [14-12] (func3). And the immediate value is sign-extended.
- The ALU operates on the data read from the register file and the sign-extended immediate value, using ALU Control's output to generate the ALU function.
- The sum from the ALU is used as the address for the data memory.
- The data from the memory unit is written into the register file (instruction[11-7]).

SW:



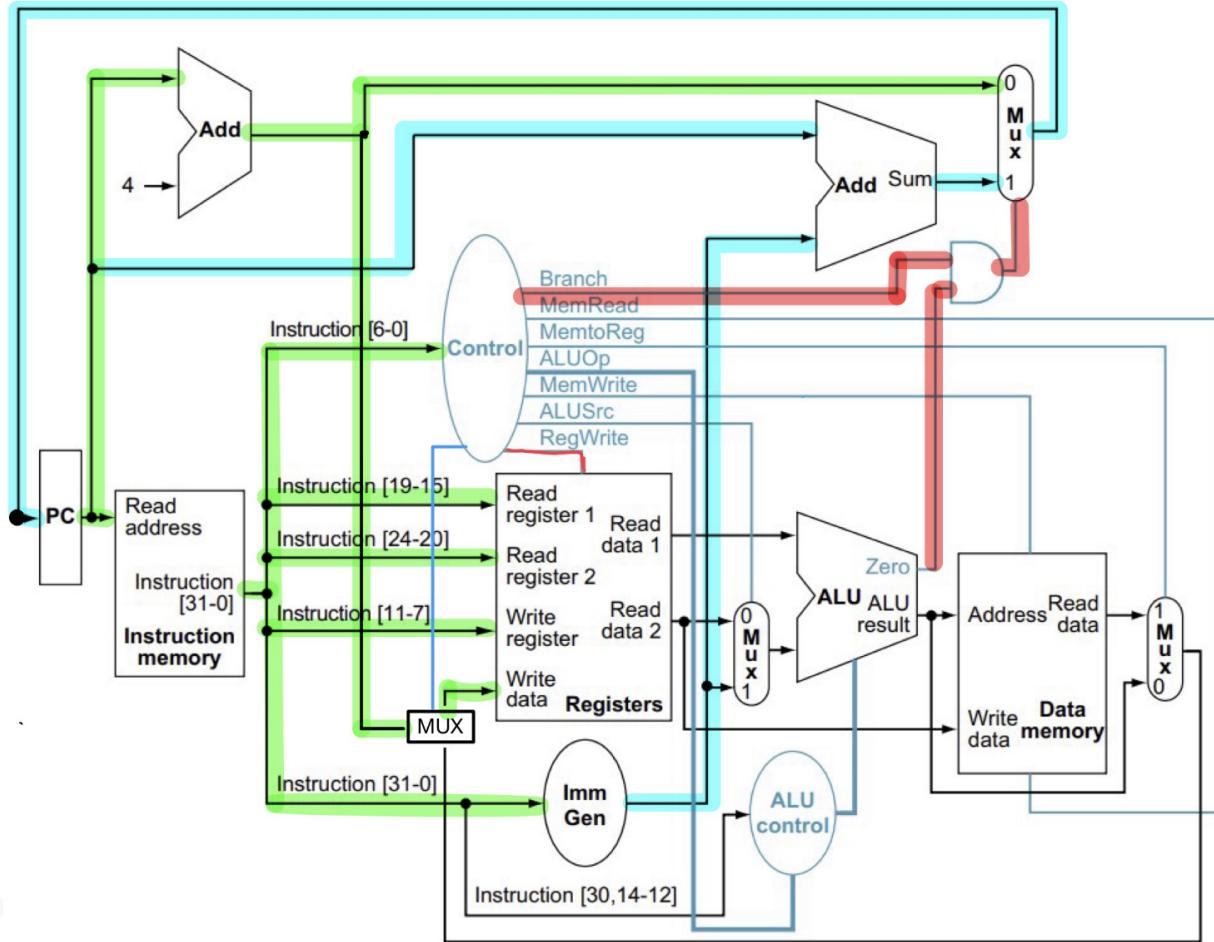
- Instruction is fetched from the instruction memory, and the PC is incremented.
- Instruction is decoded and the registers (instruction[19-15], instruction[24-20]) are read from the register file. ALUSrc, RegWrite, MemWrite, ALUSrc are set in the control unit. ALUOp1 = 1, ALUOp0 = 0 for BEQ, ALUOp1 = 0 ALUOp0 =1 for BNE.
- ALU Control are set to relevant function based on Instruction [14-12] (func3). And the immediate value is sign-extended.
- The ALU operates on the data read from the register file and the sign-extended immediate value, using ALU Control's output to generate the ALU function.
- The sum from the ALU is used as the address for the data memory.
- The value of register 2 (instruction[24-20]) is written to the data memory.

SB-Type (BEQ, BNE):



- Instruction is fetched from the instruction memory, and the PC is incremented.
- Instruction is decoded and the registers (instruction[19-15], instruction[24-20]) are read from the register file. Branch, are set in the control unit. ALUOp1 = 0, ALUOp0 = 1
- ALU Control are set to relevant function based on Instruction [14-12] (func3). And the immediate value is sign-extended.
- The ALU operates on the data read from the register file and compare those two values. Output the Boolean result.
- The multiplexer will choose between `PC+4` or the sign-extended immediate value based on the `PCSrc (Branch&&Zero)`
- Update the PC Register.

JAL:



- Instruction is fetched from the instruction memory, and the PC is incremented.
- Branch, Jump (for the extra multiplexer), RegWrite are set in the control unit.
- The immediate value is sign-extended.
- ALU output a zero signal so that PCSrc is asserted.
- Update PC. $PC_{new} = PC_{former} + sext(Imm)$
- Update the register (instruction[11-7]) to the value of $PC_{former} + 4$

Optimization

- (1) In simulation, we are executing each stage in serial. However, in real CPU, the five stages are executed simultaneously. Thus, we can use a thread pool to better optimize and simulate the processing of instruction. For five stages, five threads would be used, and each thread is assigned with a fixed stage task (IF/ID/EX/MEM/WB).
- (2) Add branch prediction to improve CPU cycle efficiency in branch instructions.
- (3) Try some optimization features which is commonly used in compilation optimization like out-of-order execution or superscalar architecture.
- (4) Optimize the ALU: The ALU is a critical component of the CPU, and its performance has a significant impact on overall CPU performance.
- (5) Use faster memory components: According to the data from the textbook, memory fetch is the

most time-consuming task for the CPU. Reducing the delay on memory I/O can reduce the cycle time directly.

(6) Hardware Optimization: Try to implement some optimization on the hardware like the circuit, gates, and logic. Try to eliminate the components required to reduce the power required and improve the efficiency at the same time.

(7) Power Management: Implement power management techniques to reduce energy consumption. For example, we can reduce the power required when the CPU is in low utilization or idle states.