

Project Outline: Instruction Set Interpreter

Course: Computer Architecture

Deadline: May 08, 2025

Group Size: Up to 7 students per group or individual work

Objective: Design and implement a software program in either JAVA or C++ that can interpret instructions from various input sources.

Part 1: Reading from a Text File

- 1- Create a text file containing instructions in either binary or HEX format. Each line should represent a single instruction.
- 2- Write a program that reads the instructions from the text file.
- 3- Implement a parser to interpret the instructions and execute them accordingly.
- 4- Display the output or result of each instruction to the screen.

Part 2: Reading from the Screen (Binary or HEX) **BOTH ARE MANDATORY**

- 1- Prompt the user to input instructions directly from the screen in either binary or HEX format.
- 2- Implement a parser to interpret the user input and execute the instructions.
- 3- Display the output or result of each instruction to the screen.

Part 3: Reading from the Screen (Instruction String)

- 1- Prompt the user to input instructions as a string directly from the screen.
- 2- Implement a parser to interpret the instruction string and execute the instructions.
- 3- Display the output or result of each instruction to the screen.

General Requirements:

- Implement a modular and well-documented code structure.
- Handle errors and edge cases gracefully.
- Use appropriate data structures to represent and manipulate instructions.
- Include comments to explain the logic and functionality of the code.
- Test the program with various instructions to ensure correct interpretation and execution.

Additional Challenges (Optional):

- Add the ability to write the results back to a file or display them in a user-friendly format.
- Consider optimizing the program for performance or memory efficiency.

Submission Guidelines:

- Submit the source code and necessary files (e.g., input text files).
- Include a README file explaining how to run the program and any additional notes on the implementation.
- In addition, you must submit the code in a pdf file.

Evaluation:

Your projects will be evaluated based on the following criteria:

- Part 1: **5 Marks**
- Part 2: **5 Marks**
- Part 3: **5 Marks**

This project assignment is an exciting opportunity to apply your knowledge of computer architecture and develop essential skills in designing and simulating computer components.

Good luck with your projects!

Hesham Alhumyani

	Symbol	Hex Code		Description
		I = 0	I = 1	
Memory	AND	0xxx	8xxx	AND memory word to AC
	ADD	1xxx	9xxx	Add memory word to AC
	LDA	2xxx	Axxx	Load AC from memory
	STA	3xxx	Bxxx	Store content of AC into memory
	BUN	4xxx	Cxxx	Branch unconditionally
	BSA	5xxx	Dxxx	Branch and save return address
	ISZ	6xxx	Exxx	Increment and skip if zero
Register	CLA	7800		Clear AC
	CLE	7400		Clear E
	CMA	7200		Complement AC
	CME	7100		Complement E
	CIR	7080		Circulate right AC and E
	CIL	7040		Circulate left AC and E
	INC	7020		Increment AC
	SPA	7010		Skip next instr. if AC is positive
	SNA	7008		Skip next instr. if AC is negative
	SZA	7004		Skip next instr. if AC is zero
	SZE	7002		Skip next instr. if E is zero
	HLT	7001		Halt computer
I/O	INP	F800		Input character to AC
	OUT	F400		Output character from AC
	SKI	F200		Skip on input flag
	SKO	F100		Skip on output flag
	ION	F080		Interrupt on
	IOF	F040		Interrupt off