FISA 1.0 — Specification Sheet

“FISA — The Power of Flame at Instruction Level”

Overview

FISA (Flame Instruction Set Architecture) is a custom-designed ISA built to demonstrate the fundamentals of pro

Version 1.0 focuses on arithmetic, bitwise logic, memory operations, registers, and basic control flow.

Registers

- 16 Integer Registers: R0 – R15
- 16 String Registers: S0 – S15

Memory Model

- 512 RAM Elements: Accessible via PUSH, POP, and PUT instructions.
- Both registers and memory can hold values for computation and storage.

Instruction Set (Opcodes)

START | 0 | Entry point of a program

MOV | 1 | Move value into a register

ADD | 2 | Add two values

SUB | 3 | Subtract values

MUL | 4 | Multiply values

DIV | 5 | Divide values

AND | 6 | Bitwise AND

OR | 7 | Bitwise OR

XOR | 8 | Bitwise XOR

NOT | 9 | Bitwise NOT

SHOW | 10 | Print value to screen

CALL | 11 | Call label (subroutine jump)

RET | 12 | Return from subroutine

JZ | 13 | Jump if zero

JNZ | 14 | Jump if not zero

JAL | 15 | Jump if less

JAG | 16 | Jump if greater

HLT | 99 | Halt program execution

SMOV | 100 | Move string into string register

WAIT | 101 | Pause execution (milliseconds)

CLEAR | 102 | Clear the screen

PUSH | 103 | Push value to stack

POP | 104 | Pop value from stack

PUT | 105 | Puts a value from a RAM address into a register

GRSCR | 106 | (Reserved for graphics — FISA 2.0)

Notes

- Labels are supported for structured jumps and subroutine calls.
- Arithmetic and logic operations work directly on registers.
- Strings and integers are separated into different register sets.

Future Roadmap

- FISA Graphics Extension (shapes, lines, text, screen handling) will be published soon, as it is still being worked on.
- Jump instructions will be extended in FISA 2.0 for more control flow flexibility.

Credits

Made By Me, and got some help from ChatGPT!!! ■