Simple Instruction Architecture

Spring 2025

The simple instruction architecture (SIA) is designed to be an architecture that is easy to assemble, easy to create a virtual machine for and easy for beginners to computer architecture to understand. While a typical RISC (reduced instruction set computer/chip) has dozens of instructions. SIA has less than 20 instructions.

SIA has 32 registers, numbered from 0-31(5-bit selector). Unlike many RISC chips, register 0 is not a constant 0; it is a general-purpose register. All registers are 32 bits wide. All registers are interpreted as signed integers for the purpose of mathematical operations.

There are two hidden registers; the user program has no direct access to either.

The call stack is used by “Call” and “Return”. The CPU ensures that the stack appears infinite to the end user.

The status register is updated by “Compare” and is used by the Branch instructions.

## Instruction Formats

### 2R

|  |  |  |  |
| --- | --- | --- | --- |
| 5 bits | 1 bit | 5 bits | 5 bits |
| OPCODE | Format (0) | Source register | Destination register |

### Immediate

|  |  |  |  |
| --- | --- | --- | --- |
| 5 bits | 1 bit | 5 bits | 5 bits |
| OPCODE | Format (1) | Signed immediate value | Destination register |

### Call/Return

|  |  |
| --- | --- |
| 5 bits | 11 bits |
| OPCODE | Signed immediate value |

## Opcodes

All instructions use either 2R or immediate (bit-switched on “Format”) unless otherwise specified.

|  |  |  |
| --- | --- | --- |
| Opcode | Name | Description |
| 0 | Halt | Stops the virtual machine |
| 1 | Add | Adds source to destination, storing the result in destination. |
| 2 | And | Bitwise AND of source and destination, storing the result in destination. |
| 3 | Multiply | Multiplies destination by source, storing the result in destination. |
| 4 | LeftShift | Left shifts destination by source bits, storing the result in destination. |
| 5 | Subtract | Subtracts source from destination, storing the result in destination. |
| 6 | Or | Bitwise OR of source and destination, storing the result in destination. |
| 7 | RightShift | Right shifts destination by source bits, storing the result in destination. |
| 8 | Syscall | Switches to kernel mode and calls kernel function (unsigned) immediate  Format: Call/Return |
| 9 | Call | Pushes the current address + 1 on the stack. Sets PC to PC + Immediate Format: Call/Return |
| 10 | Return | Pops from the stack and sets the PC to the popped value.  Format: Call/Return (Immediate value is unused) |
| 11 | Compare | Compares Source to destination, setting the status register flags |
| 12 | BLE | If status register LESS or EQUAL are set, PC 🡨 PC + Immediate  Format: Call/Return |
| 13 | BLT | If status register LESS is set, PC 🡨 PC + Immediate  Format: Call/Return |
| 14 | BGE | If status register GREATER or EQUAL are set, PC 🡨 PC + Immediate  Format: Call/Return |
| 15 | BGT | If status register GREATER is set, PC 🡨 PC + Immediate  Format: Call/Return |
| 16 | BEQ | If status register EQUAL is set, PC 🡨 PC + Immediate  Format: Call/Return |
| 17 | BNE | If status register EQUAL is NOT set, PC 🡨 PC + Immediate  Format: Call/Return |
| 18 | Load | Loads from memory address source into destination Format: 2R |
| 18 | Load | Loads from memory address (immediate + destination) into destination register  Format: Immediate |
| 19 | Store | Stores into memory address destination from source |
| 20 | Copy | Copies source value into destination |
| Remaining opcodes are reserved for future use and should not be used | | |