ISA 2022 - Instruction Definitions

Note: Memory addresses (shortened to Addr) which are represented as 16 bit numbers are split up into two bytes. The first byte, $\text{Addr}_{\text{High}}$ stores the first 8 most significant bits of the address and the second byte, Addr_{Low} stores the last 8 least significant bits of the address.

Math

OpCode			Description
0x10	ADD	ADD R_{Dest} R_{Addend} R_{Addend}	Adds the values of two registers and stores the sum in a destination register
0x11	SUB	SUB R_{Dest} R_{Minuend} $R_{\text{Subtrahend}}$	Subtracts the values of two registers and stores the difference in a destination register
0x12	MUL	MUL R_{Dest} R_{Factor} R_{Factor}	Multiplies the values of two registers and stores the product in a destination register
0x13	DIV	DIV R_{Dest} R_{Dividend} R_{Divisor}	Divides the values of two registers and stores the quotient in a destination register
0x14	MOD	MOD R_{Dest} R_{Dividend} R_{Divisor}	Divides the values of two registers and stores the remainder in a destination register
0x15	SHL	SHL R_{Dest} R_{Source} Pad	Performs a bitwise left shift on a given source register and stores the result in a destination register
0x16	SHR	SHR R_{Dest} R_{Source} Pad	Performs a bitwise right shift on a given source register and stores the result in a destination register

Logic

OpCode			Description
0x20	NOT	NOT R_{Dest} R_{Source} Pad	Performs a bitwise NOT operation on a given source register and stores the result in a destination register
0x21	AND	AND R_{Dest} R_{LHS} R_{RHS}	Performs a bitwise AND operation on a given source register and stores the result in a destination register
0x22	OR	OR R_{Dest} R_{LHS} R_{RHS}	Performs a bitwise OR operation on a given source register and stores the result in a destination register
0x23	XOR	XOR R_{Dest} R_{LHS} R_{RHS}	Performs a bitwise XOR operation on a given source

			register and stores the result in a destination register
0x24	GT	GT R_{Dest} R_{LHS} R_{RHS}	Calculates if the value of a given register is greater than another. Stores the result in a destination register.
0x25	LT	LT R_{Dest} R_{LHS} R_{RHS}	Calculates if the value of a given register is less than another. Stores the result in a destination register.
0x26	EQ	EQ R_{Dest} R_{LHS} R_{RHS}	Calculates if the value of a given register is equal to another. Stores the result in a destination register.

Flow Control

OpCode			Description
0x00	NOP	NOP Pad Pad Pad	No operation
0x30	SKP	SKP Pad $Addr_{High}$ $Addr_{Low}$	Moves the instruction pointer to a given address in memory.
0x31	SKPT	SKPT $R_{Condition}$ $Addr_{High}$ $Addr_{Low}$	Moves the instruction pointer to a given address in memory if the value of a given register is 1.
0x32	SKPF	SKPF $R_{Condition}$ $Addr_{High}$ $Addr_{Low}$	Moves the instruction pointer to a given address in memory if the value of a given register is 1.

Memory

OpCode			Description
0x40	SET	SET R_{Dest} $Value_{High}$ $Value_{Low}$	Sets the value of a given register to a 16-bit value represented by two 8-bit values.
0x41	MOV	MOV R_{Dest} R_{Source} Pad	Copies the value of a $register_{Dest}$ to the value of $register_{Source}$
0x42	LOAD	LOAD R_{Dest} $Addr_{High}$ $Addr_{Low}$	Loads a value from memory at a given address into $register_{Dest}$
0x43	STR	STR R_{Source} $Addr_{High}$ $Addr_{Low}$	Stores a value from $register_{Source}$ into memory at a given address

0x44	PUSH		?
0x45	POP		?