General architecture info:

* Little Endian (0xFFFE -> |0xFE|0xFF|)
* 8-bit data bus
* 16-bit address bus

Registers:

General Purpose Registers:

* Accumulator (A)
* B
* C
* D

Special registers:

* Stack Pointer (SP)
  + Prepended with 00 to get the address stored.
  + Stack grows downwards.
* Base Pointer (BP)
  + Prepended with 00 to get the address stored, related to SP.
* Status (S)
  + System flags stored as bits in this register.

16-bit register:

* Instruction Pointer (IP)
  + 16-bit; holds current execution address.
  + 0xFFFE + 0xFFFF make the starting IP location:
    - LSB stored at 0xFFFE.
    - MSB stored at 0xFFFF.
* Return (R)
  + 16-bit, saves return addresses on calls, only affected by CALL and RET.

0xFFFD is the terminal buffer location.

Memory:

64KB of memory.

Memory map:

0xFFFF +-----+

|START| (0xFFFE & 0xFFFF)

0xFFFE +-----+

|PRINT| (0xFFFD)

0xFFFD +-----+

| RAM | (0x0100 -> 0xFFFC)

0x0100 +-----+

|STACK| (0x00FF -> 0x0000)

0x0000 +-----+

Status

S bit flags:

7 6 5 4 3 2 1 0

+-+-+-+-+-+-+-+-+

|H|G|T|C|E|N|Z|L|

+-+-+-+-+-+-+-+-+

* b0: Less, Set by CMP
* b1: Zero, set if a register receives a 0x00 value.
* b2: Negative, Overflow in Subtraction.
* b3: Equal, Set by CMP
* b4: Carry, Overflow in Addition.
* b5: Terminal, Active while printing to the Terminal memory address.
* b6: Greater, set by CMP
* b7: Halt, set when HLT (0x18) is called, clock will be ignored.

Arguments

Argument Types:

* IMM8: An 8-bit argument that represents raw data, one byte wide.
* IMM16: A 16-bit argument that represents raw data, 2 bytes wide.
* REG8: An 8-bit value that represents one of the four general purpose registers, one byte wide:
  + 0x00 represents A
  + 0x01 represents B
  + 0x02 represents C
  + 0x03 represents D
* REG16: A combo of 2 8-bit registers to act as a 16-bit register:
  + 0x04 represents AB
  + 0x05 represents CD

Instructions:

DIRECTIVES/NO OPCODE

1. Directives

* .setloc (IMM16): Set compiler IP to IMM16, a sort of pseudo jump.
* .start (IMM16): Set 0xFFFE and 0xFFFF to IMM16 (setting a start address).

2. No Opcode:

* LBL (label): Store the address and name of the label (pointing to the current IP value) as a pair.

NOP

0x00: NOP: Do nothing.

MOV

0x01: MOVA (IMM8) Immediate: Copy Immediate into A [Z].

0x02: MOVB (IMM8) Immediate: Copy Immediate into B [Z].

0x03: MOVC (IMM8) Immediate: Copy Immediate into C [Z].

0x04: MOVD (IMM8) Immediate: Copy Immediate into D [Z].

0x05: CPY (REG)Source, (REG)Destination: Copy Source’s value to Destination [Z].

MEMORY

0x06: STR (REG8) Source (IMM16) Address: Copy Source's value to the byte in memory pointed to by Address [Z].

0x07: LDR (IMM16) Address (REG8) Destination: Copy data from the byte in memory pointed to by Address to Destination [Z].

MATH

0x08: ADD (REG8) Destination (REG8) Source: Add Source's value to Destination's value and store in Destination [C, Z].

0x09: SUB (REG8) Destination (REG8) SRC: Subtract Source's value from Destination's value and store in Destination [Z, N].

0x0A: INC (REG8) Destination: Increment Destination's value by 1 [Z, C].

0x0B: DEC (REG8) Destination: Decrement Destination's value by 1 [Z, N].

0x0C: ADC (REG8) Destination (REG8) Source: Add Source's value and the carry bit to Destination's value and store in Destination [Z, C].

0x0D: SBB (REG8) Destination (REG8) Source: Subtract Source's value and the negative bit from Destination's value and store in Destination [Z, N].

0x0E: NEG (REG8) Destination: Make Destination's value Negative (2’s complement) [Z, N].

0x0F: FLP (REG8) Destination: Flip Destination's value around b(10101010) -> b(01010101) [Z, N].

STACK

0x10: PSHI (IMM8) Immediate:

Set the value pointed to by SP to Immediate.

Decrement the value pointed to by SP.

0x11: PSHR (REG)Source:

Set the value pointed to by SP to the Source's value

Decrement the value pointed to by SP.

0x12: POP (REG)Destination: [Z]

Increment the value pointed to by SP.

Set Destination's value to the data at SP.

Set the value pointed to by SP to 0x00.

JUMPING

0x13: JMP (IMM16) Address: Set the value pointed to by IP to the byte in memory pointed to by Address without saving IP on the stack.

0x14: CALL (IMM16) Address:

Make a Call Stack Frame

Set the value of BP to the value of SP+2.

0x15: RET:

Pop the next 2 values (Should be IP) to IP

Set the value pointed to by SP to the value pointed to by BP

Set the value pointed to by BP to 0x00

0x16: JZ (IMM16) Address: Jump to Address if Zero (Z) is set.

0x17: JNZ (IMM16) Address: Jump to Address if Zero (Z) is not set.

END

0x18: HLT: [H]

Set H.

Ignore the clock signal.

Quit (If in Emulator).

Must reset to undo!

LESS

0x19: JL: (IMM16) Address: Jump to Address if Less Flag (L) is set.

0x1A: JNL: (IMM16) Address: Jump to Address if Less Flag (L) is not set.

STATUS

0x1B: SETL: Set Less Flag [L].

0x1C: CLRL: Clear Less Flag [L].

0x1D: SETZ: Set Zero [O].

0x1E: CLRZ: Clear Zero [O].

0x1F: SETN: Set Negative [N].

0x20: CLRN: Clear Negative [N].

0x21: SETE: Set Parity [E].

0x22: CLRE: Clear Equal [E].

0x23: SETC: Set Carry [C]

0x24: CLRC: Clear Carry [C]

LOGIC

0x25: AND (REG8) Destination (REG8) Source: Bitwise AND Destination's value and Source's value and store in Destination [Z].

0x26: OR (REG8) Destination (REG8) Source: Bitwise OR Destination's value and Source's value and store in Destination [Z].

0x27: XOR (REG8) Destination (REG8) Source: Bitwise XOR Destination's value and Source's value and store in Destination [Z].

0x28: NOT (REG8) Destination: Invert Destination's value and store in Destination [Z].

EQUAL

0x29: JE (IMM16) Address Jump to Address if Equal Flag (E) is set

0x2A: JNE (IMM16) Address (REG8) Destination (REG8) Source: Jump to Address if Destination's value and Source's value are not equal.

MISC

0x11: CLR (REG8) Destination: Set Destination to 0 [Z].

0x2C: LEA (REG16) Base (REG8) Offset (REG8) Destination:

Load 8-bit from memory at [Base + Offset's value] into Destination [Z].

PRINT

0x2D: PRNI (IMM8) Value: Set 0xFFFD to Value and set the Terminal flag [T].

0x2E: PRNR (REG8) Source: Set 0xFFFD to Source's value and set the Terminal flag [T].

0x2F: ENPR: Clear the Terminal flag (Required when done printing) [T].

CARRY

0x30: JC (IMM16) Address: Jump to Address if Carry Flag (C) is set.

0x31: JNC (IMM16) Address: Jump to Address if Carry Flag (C) is not set.

NEGATIVE

0x32: JN (IMM16) Address: Jump to Address if Negative Flag (N) is set.

0x33: JNN (IMM16) Address: Jump to Address if Negative Flag (M) is not set.

GREATER

0x34: JG (IMM16) Address: Jump to Address if Greater Flag (G) is set.

0x35: JNG (IMM16) Address: Jump to Address if Greater Flag (G) is not set.

TEMP

0x36: IBPR (IMM8) OFFSET (REG8) Destination: Copy data from the BP BP+OFFSET to Destination [Z].

OFFSET

0x37: LDRO (IMM16) Address, (REG8) OFFSET, (REG8) Destination: Copy data from the byte in memory pointed to by Address + OFFSET's value to Destination [Z].

WIDE

0x38: MOVW (REG16) Destination (IMM16) Address: Copy 16-bit address into wide register [Z].

0x39: PSHW (REG16) Source: Push 16-bit value in wide register onto stack.

0x3A: POPW (REG16) Destination: Pop 16-bit value from stack into wide register [Z].

0x3B: LDW (REG16) Source (REG8) Destination: Load 8-bit from memory address in wide register Source into Destination [Z].

0x3C: STW (REG8) Source (REG16) Destination: Store 8-bit from Source into memory address pointed to by wide register Destination [Z].

0x3D: JMPW (REG16) Destination: Jump to 16-bit address in wide register Destination.

0x3E: CALW (REG16) Destination: Call subroutine at 16-bit address in wide register Destination.

0x3F: ADDW (REG16) Destination (IMM16) Address: Add 16-bit address value to wide register Destination [Z, C].

0x40: INCW (REG16) Destination: Increment wide register Destination by 1 [Z, C].

0x41: DECW (REG16) Destination: Decrement wide register Destination by 1 [Z, N].

COMPARE

0x42: CMP (REG8) A (REG8) B: update flags based on how A is related to B [G, E, L].

0x43: SETG: Set Greater [G].

0x44: CLRG: Clear Greater [G].

WIDE (CON’T)

0x45: IBPW (IMM8) OFFSET (REG16) Destination: Copy 2 bytes of data from the BP BP+OFFSET to BP+OFFSET+1 to Destination [Z].