LiteChip-8 Instruction Set:

Based on 8080/8085 series microprocessors

Overview:

CON (control word):

- 16-bit CON word
- ICR PO AI AO OO SU TI BI BO CI CO OI MI RO II IO

Macro Instructions:

- LDA (load to accumulator operand that follows instruction)
- ADD (add register X to accumulator)
- SUB (subtract register X from accumulator)

Note: Add and Sub instructions are isolated to the general registers only. Despite these typically being MRIs, I chose to omit that since we have the MVI instruction.

- MOV (move register contents to another register)
- MVI (Load register X with operand that follows instruction)

Note for MVI: usually the operand after MVI instruction would be data, but because of architecture limitation, our operand will be an address. Because of this, we can omit the instruction MVI A, as LDA would be the equivalent.

- OUT (output data from accumulator to output port)
- HLT (stop processes)
- NOP (no operation)

Microinstructions:

- ICR PO (increment, program counter out) - Program Counter
- AI AO (accumulator in, accumulator out) - Accumulator A
- OO SU (ALU out, ALU subtract) - ALU Adder-Subtractor
- TI (temporary register in) - TMP Register
- BI BO (register B in, register B out) - Register B
- CI CO (register C in, register C out) - Register C
- OI (output register out) - Output Register (or Port)
- MI (MAR in) - Memory Access Register (MAR)
- RO (RAM out) - Random Access Register (RAM)
- II IO (IR in, IR out) - Instruction Register (IR)

Opcode - Full Instruction List- Example Data

0000 - LDA - (0000 1001(9H))
0001 - ADD B - (0001 XXXX)

0010 - ADD C - (0010 XXXX)
0011 - SUB B - (0011 XXXX)
0100 - SUB C - (0100 XXXX)
0101 - MVI B - (0101 1110(EH))
0110 - MVI C - (0110 0000(0H))
0111 - MOV A,B - (0111 XXXX)
1000 - MOV A,C - (1000 XXXX)
1001 - MOV B,A - (1001 XXXX)
1010 - MOV B,C - (1010 XXXX)
1011 - MOV C,A - (1011 XXXX)
1100 - MOV C,B - (1100 XXXX)
1101 - OUT - (1101 XXXX)
1110 - HLT - (1110 XXXX)
1111 - NOP - (1111 0000)
Note: The only memory referenced instructions are LDA, MVI B, and MVI C.

Instruction Cycle:

- 6 T states (T state = time to complete one microinstruction)
- Divided into two parts: fetch cycle (first 3 states) and execution cycle (last 3 states)

Fetch Cycle:

- Includes T states T1, T2, and T3
- Microinstructions are the same during the fetch cycle no matter what instruction is being done.

Execution Cycle:

- Includes T state T_4 T_5 and T_6
- Microinstructions vary depending on the instruction

T-State Microinstructions:

Fetch Cycle:

Address State T1:
- PO MI

Increment State T2:
- ICR

Memory State T3:
- RO II

Execution Cycle:

LDA: 0000

- T_4 (IO MI)
- T_5 (AI RO)
- T_6 (nop)

ADD B: 0001

- T_4 (BO TI)
- T_5 (OO AI)
- T_6 (nop)

ADD C: 0010

- T_4 (CO TI)
- T_5 (OO AI)
- T_6 (nop)

SUB B: 0011

- T_4 (BO TI)
- T_5 (OO AI SU)
- T_6 (nop)

SUB C: 0100

- T_4 (CO TI)
- T_5 (OO AI SU)
- T_6 (nop)

MVI B - 0101

- T_4 (IO MI)
- T_5 (RO BI)
- T_6 (nop)

MVI C - 0110

- T_4 (IO MI)
- T_5 (RO CI)
- T_6 (nop)

MOV A,B - 0111

- T_4 (AO BI)
- T_5 (nop)
- T_6 (nop)

MOV A,C - 1000

- T_4 (AO CI)
- T_5 (nop)
- T_6 (nop)

MOV B,A - 1001

- T_4 (BO AI)
- T_5 (nop)
- T_6 (nop)

MOV B,C - 1010

- T_4 (BO CI)
- T_5 (nop)
- T_6 (nop)

MOV C,A - 1011

- T_4 (CO AI)
- T_5 (nop)
- T_6 (nop)

MOV C,B - 1100

- T_4 (CO BI)
- T_5 (nop)
- T_6 (nop)

OUT: 1101

- T_4 (AO OI)
- T_5 (nop)
- T_6 (nop)

HLT: 1111

- T_4 (HLT)
- T_5 (nop)
- T_6 (nop)

Important Information:

- Opecode (4-bit binary of mnemonic, upper nibble) + operand (4-bit binary address location, lower nibble).
- Instructions that use operand are called memory reference instructions (MRIs)
- Raw data includes upper nibble and lower nibble, doesn't go into IR, doesn't have memory address or opcode, just numbers to be crunched.
- The instruction register outputs opcodes instantaneously (IO not necessary for instructions that aren't MRIs)
- If once wanted a variable machine cycle, so their processor is faster, they could combine all the CON bits to a NAND gate, and if that NOR gate is not receiving a signal from any of them (indicating a nop), it can send a signal to the ring counter `CLR to skip the rest of the T states.

Examples for MRIs:
LDA 9H → 0000 1001
Loads the accumulator with contents of RAM memory location 9H (decimal 9 or binary 1001)

ADD B → 0001 0000
Adds the contents of B with what's in the accumulator by using the adder-subtractor