

Marie Instruction Set Cheat Sheet

Each instruction is 16 bits with the first 4 bits being the opcode (i.e. the operation code selects which instruction to execute). For example the instruction 31F0 is 3-1f0 so it's the Add X instruction and X is the address 1F0. The instruction 10ff is the load instruction Load X so the contents of address 0ff will be loaded into the AC.

Mnemonic	Hex	Description
Add X	3	Add the contents of address X to AC
AddI X	B	Add indirect: Use the value at X as the actual address of the data operand to add to AC
Clear	A	Put all zeros in AC
Input	5	Input a value from the keyboard into AC
Halt	7	Terminate program
Jump X	9	Load the value of X into PC
JumpI X	C	Use the value at X as the address to jump to
JnS X	0	Store the PC at address X and jump to X+1
Load X	1	Load contents of address X into AC
LoadI X	D	Load indirect: Use the value at X as the address of the value to load.
Output	6	Output the value in AC to the display
Skipcond X	8	Skip next instruction on condition (See note below.)
Store X	2	Store the contents of AC at address X
StoreI X	E	Store indirect: Use X the value at X as the address of where to store the value.
Subt X	4	Subtract the contents of address X from AC

Note regarding use of SKIPCOND:

The two address bits closest to the opcode field, bits 10 and 11 specify the condition to be tested. If the two address bits are 00, this translates to "skip if the AC is negative". If the two address bits are 01, this translates to "skip if the AC is equal to 0". Finally, if the two address bits are 10 (or 2), this translates to "skip if the AC is greater than 0".

Example: the instruction Skipcond 800 will skip the instruction that follows if the AC is greater than 0.

Example: Type the following program in and single step it noting changes in AC and memory. Make sure you note the commas in the data declarations below. They are REQUIRED in the .mas file. Also set the output windows format to Dec not ASCII as it is normally set.

```
LOAD num1
ADD num2
STORE result
OUTPUT
HALT
num1, DEC 3
num2, DEC 5
result, DEC 0
```