All instructions that ends with .I (For example, ADD.I) are “Immediate” instructions that come with an immediate value.

Opcode	Instruction	Description
00000000	NOP	No operation
00000001	LDA [addr]	Loads register A with MEM[addr]
00000010	ADD [addr]	Adds register A with MEM[addr], and store the result to register A; sets flags
00000011	SUB [addr]	Subtracts MEM[addr] from register A, and store the result to register A; sets flags
00000100	STA [addr]	Stores register A to MEM[addr]
00000101	LDA.I [val]	Loads register A with val (i.e. Load immediate)
00000110	JMP [pc]	Unconditionally jumps to the specified pc
00000111	JMC [pc]	Jumps to pc if the Carry Flag of the ALU is set
00001000	JMZ [pc]	Jumps to pc if the Zero Flag of the ALU is set
00001001	ADD.I [val]	Adds register A with val, and store the result to register A
00001010	SUB.I [val]	Subtracts val from register A, and store the result to register A
00001011	LDA.REL [offset]	Loads register A with MEM[RegA + offset] (very useful for relative addressing)
00001100	JMN [pc]	Jumps to pc if the Negative Flag of the ALU is set
00001101	NOP	Another No operation
00001110	OUT	Outputs content of register A to the seven-segment display
00001111	HLT	Halts the execution of the program (halts the clock)
00010000	NOP	Another No operation
00010001	TAPI	Send register A as an instruction to the LCD display
00010010	TAPD	Send register A as data to the LCD display
00010011	PSH	Pushes register A onto the stack
00010100	POP	Pops the stack onto register A
00010101	TSA	Transfers stack pointer register to register A (stack pointer register can be used as general-purpose register when stack not needed)
00010110	NOP	Another No operation
00010111	TAS	Transfers register A to stack pointer register
00011000	SUBF [addr]	Subtracts MEM[addr] from register A, but does not store the result back to register A; just sets the flags.
00011001	SUBF.I [val]	Subtracts val from register A, but does not store the result back; just sets the flags
00011010	JSR [pc]	Jumps to subroutine; before jumping, pushes the return address onto the stack
00011011	RTS	Returns from subroutine by popping the stack to get return address, and jumping to that return address.
00011100	TIA	Transfers Input Module value to register A.
00011101	TRA	Transfers Random Module value to register A.
00011110	LDPI.I [val]	Sends val to LCD display module as instruction.
00011111	LDPD.I [val]	Sends val to LCD display module as data.
00100000	TPIA	Reads LCD display current address and “busy” flag; stores result to register A.
00100001	TPDA	Reads current LCD display data; stores result to register A.