STUMPED INSTRUCTION SET TABLE

Instruction Mnemonic	Machine Code in Hexadecimal	Action (Microcode)	Comments
NOP	0000	Do Nothing	
HALT	0F00	Stop CPU Execution	
PUSHPC	0100	PC → top	Pay close attention to sign issues here
POPPC	0200	top → PC	
LD	0300	mem[top] → top	
ST	0400	top → mem[next]	Both are popped off of the stack
DUP	0500	Copies top of stack	Stack: top → top top
DROP	0600	Removes top of stack	Stack: next top → next Note: this is like pop but the top of the stack is discarded.
OVER	0700	Copies next to top	Stack: next top → next top next
DNEXT	0800	Drops next	Stack: next top → top
SWAP	0900	Swaps top and next	Stack: next top → top next
PUSHI I	1000 + I	I → top	
PUSH A	2000 + A	mem[A] → top	
POP A	3000 + A	top →mem[A]	
JMP A	4000 + A	A → pc	
JZ A	5000 + A	A → pc if top == 0	
JNZ A	6000 + A	A → pc if top != 0	
IN	D000 + P	in[port P] → top	
OUT	E000 + P	top → out[port P]	
ADD	F000	next + top → top	
SUB	F001	next - top → top	
MUL	F002	next * top → top	
DIV	F003	next / top → top	Integer division, division by zero halts CPU
MOD	F004	next % top → top	Integer modulus
SHL	F005	next << top → top	
SHR	F006	next >> top → top	
BAND	F007	next & top → top	
BOR	F008	next | top → top	
BXOR	F009	next ^ top → top	
AND	F00A	next && top → top	Logical operations, not bitwise. Non-zero implies true, zero is false
OR	F00B	next || top → top	
EQ	F00C	next == top → top	
NE	F00D	next != top → top	
GE	F00E	next >= top → top	
LE	F00F	next <= top → top	
GT	F010	next > top → top	
LT	F011	next < top → top	
NEG	F012	-top → top	Unary minus
BNOT	F013	~top → top	Bitwise inversion
NOT	F014	!top → top	Logical negation