Cambridge-1 Instruction Set

Registers
areg = A Register (4-bit)
breg = B Register (4-bit)
opreg = Operand Register (8-bit)
outreg = Output Register (8-bit)
pc = Program Counter (8-bit)

Operations
<< bitwise shift
|| concatenation

Flags
cyflag = Carry Flag
eqflag = Equal to Zero Flag
gzflag = Greater than Zero Flag

Mnemonic	Instruction	Action
ADDC	0 0 0 0	areg ← areg + breg + cyflag
SUBC	0 0 0 1	areg ← areg - breg + cyflag
CLR	0 0 1 0	cyflag = 0
LDA	0 0 1 1	areg ← opreg(lower)
LDB	0 1 0 0	breg ← opreg(lower)
MVAB	0 1 0 1	breg ← areg
MVBA	0 1 1 0	areg ← breg
LALN	0 1 1 1	areg ← memory[opreg](lower)
LBUN	1 0 0 0	breg ← memory[opreg](upper)
STM	1 0 0 1	memory[opreg] ← breg << 4 || areg
JUNC	1 0 1 0	pc ← opreg
JUZE	1 0 1 1	if (eqflag) then pc ← opreg
JUGZ	1 1 0 0	if (gzflag) then pc ← opreg
PFIX	1 1 0 1	opreg ← opreg << 4
OUT	1 1 1 0	outreg ← breg << 4 || areg
HALT	1 1 1 1	clock = stop