Cambridge-1 Instruction Set

Last updated: 22/09/19

Registers

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

Operations

<< bitwise shift
|| concatenation
nop no operation

Flags

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

Mnemonic	Instruction	Operand	Details
ADDC	0 0 0 0	not used	areg ← areg + breg + cyflag
SUBC	0 0 0 1	not used	areg ← areg - breg + cyflag
CLR	0 0 1 0	not used	cyflag = 0
LDA	0 0 1 1	immediate	areg ← opreg(lower)
LDB	0 1 0 0	immediate	breg ← opreg(lower)
MVAB	0 1 0 1	not used	breg ← areg
MVBA	0 1 1 0	not used	areg ← breg
LALN	0 1 1 1	lower addr	areg ← memory[opreg](lower)
LBUN	1 0 0 0	lower addr	breg ← memory[opreg](upper)
STM	1 0 0 1	lower addr	memory[opreg] ← breg << 4 || areg
JUNC	1 0 1 0	lower addr	pc ← opreg
JUZE	1 0 1 1	lower addr	if (eqflag) then pc ← opreg
JUGZ	1 1 0 0	lower addr	if (gzflag) then pc ← opreg
PFIX	1 1 0 1	upper addr	opreg ← opreg << 4
IOP	1 1 1 0	[xx00]	outreg[a] ← memory[opreg]
		[xx01]	outreg[b] ← memory[opreg]
		[xx10]	areg ← inreg
		[xx11]	nop
HALT	1 1 1 1	not used	clock = stop