

Primary OP	RDest	ALUOP/ImmHi	Rsrc/ImmLo		
Name	Operands	15-12	11-8	7-4	3-0
INSTR	RDest, RSrc	0	RDest	OPCODE	RSrc
INSTRI	RDest, Imm	OPCODE	RDest	ImmHi	ImmLo
ADD	RDest, RDest	0	RDest	10	RSrc
SUB	RDest, RDest	0	RDest	11	RSrc
AND	RDest, RDest	0	RDest	110	RSrc
OR	RDest, RDest	0	RDest	111	RSrc
XOR	RDest, RDest	0	RDest	1000	RSrc
NOT	RDest, RDest	0	RDest	1001	RSrc
LSH	RDest, RDest	0	RDest	1010	RSrc
RSH	RDest, RDest	0	RDest	1011	RSrc
ARSH	RDest, RDest	0	RDest	1100	RSrc
MUL	RSrc2, RSrc	0	RSrc2	1101	RSrc
FMUL	RDest, RSrc	0	RDest	1110	RSrc
MOVR	RDest, RSrc	0	RDest	1111	RSrc

Compares

CMP (A-B)	A, B	0	A	100	B
CMP (Imm-A)	A, Imm	100	A	ImmHi	ImmLo
CMPR (B-A)	B, A	0	A	101	B
CMPR (A-Imm)	Imm, A	101	A	ImmHi	ImmLo

Primary OP	Data	OP	Address		
Name	Operands	15-12	11-8	7-4	3-0
MOVMR	RAddr, RDest	1	RDest	10	RAddr
MOVRL	RAddr, RSrc	1	RSrc	11	RAddr

OP	Imm		
Name	Operands	17-14	13-0
CALL	Imm Addr	100	Imm
MOVMRI	Imm Addr	101	Imm
MOVRMI	Imm Addr	110	Imm
RET	n/a	111	n/a

OP	Imm		
Name	Operands	17-13	12-0
JL	Imm Addr	10000	Imm
JLE	Imm Addr	10001	Imm
JNE	Imm Addr	10010	Imm
JE	Imm Addr	10011	Imm
J	Imm Addr	10100	Imm
JBE	Imm Addr	10101	Imm
JB	Imm Addr	10110	Imm

OP	Operand	ImmLo		
Name	Operands	17-14	13-10	9-0
POP	RDest	1100	RDest	n/a
PUSH	RSrc	1101	RSrc	n/a
PUSHI	Imm	1110	ImmHi	ImmLo

Name	Operands	17-12 (OP)	11-8 (Operand)	7-0 (unused)
INCR	R	111100	R	n/a
DECR	R	111101	R	n/a

Pseudoinstructions
with special
formatting
Moves

MOV [%R], %R		Move from a register into memory at address in register
MOV %R, [%R]		Move from memory at address in register into a register
MOV [IMM/LABEL], %R		Move from a register into memory at immediate address
MOV %R, [IMM/LABEL]		Move from memory at immediate address into a register
MOV %R, [LABEL+constant]		Move from memory at a label address plus a constant
MOV [LABEL+constant], %R		Move to memory at a label address plus a constant
MOV %R, %R		Move a register to another register
MOV %R, IMM		Move an immediate up to 16 bits into a register
Special Jump Formats		
JUMP A, B, LABEL		Jump to label if the two parameters compared as A JUMPTYPE B (e.g. A > B for JG) return true.

Works with the
following jump
types

JE	JNE	JA*	JAE*	JB	JBE	JG*	JGE*	JL	JLE
----	-----	-----	------	----	-----	-----	------	----	-----

(NOTE: jumps with
asterisks are
emulated using
reverse compares