

CORE INSTRUCTION SET

NAME,MNEUMONIC	FORMAT	OPERATIONS	OPCODE	FUNCT	
ADD	add	R	$R[rs] = R[rs] + R[rt]$	000	1
NAND	nand	R	$R[rs] =$ $NAND(R[rs],R[rt])$	001	1
SET LESS THAN (slt_0)	slt_0	R	$R[rs] < R[rt] ? 1 : 0$	010	0
SET LESS THAN (slt_1)	slt_1	R	$R[rs] < R[rt] ? 1 : 0$	010	1
SHIFT LEFT	sl	R	$R[rs] = R[rt] \ll 1$	011	0
SHIFT RIGHT	sr	R	$R[rs] = R[rt] \gg 1$	011	1
LOAD WORD	lw	I	$R[rs] = Mem(R[sp] +$ immediate)	100	0
SAVE WORD	sw	I	$Mem(R[sp] +$ immediate) = $R[rs]$	100	1
ADDI	addi	I	$R[rs] = R[rs] +$ immediate	101	0
BRANCH IF EQUAL	beq	J	if $R[slt_0]=R[slt_1]$; $PC = PC+immediate$	110	n/a
JUMP AND LINK	jal	J	$R[ra]=PC+1$; $PC = PC+immediate$	111	n/a
JUMP REGISTER*	jr	JR	$PC = R[ra]$	101	1
LOAD ADDRESS**	la		$R[rs] = label$		

* jr only jumps to $R[ra]$ and has opcode 00010011

** Pseudo Instruction loads address using shift left and add immediate

BASIC INSTRUCTION FORMAT

Type	7	6	5	4	3	2	1	0
R	opcode			func	rt		rs	
I	opcode			func	immediate		rs	
J	opcode			PC relative immediate				
JR	1	0	1	1	0	0	1	1

REGISTER NAME, NUMBER, USE, CALL CONVENTION

REGISTER NAME	NUMBER	USE	PRESERVED ACROSS A CALL?
\$s1	00	Saved Temporary	Yes
\$s2	01	Saved Temporary	Yes
\$sp	10	Stack Pointer	Yes
\$ra	11	Return Address	Yes