Team ExceptioNull Reference Data

Core Instruction Set

NAME	MNEMONIC	FORMAT	OPERATION	OPCODE
Move	mv	R	R[rs] = R[rt]	0000
Add	add	R	R[rd] = R[rs] + R[rt]	0001
And	and	R	R[rd] = R[rs] & R[rt]	0010
Not	not	R	R[rs] = $\sim R[rt]$	0011
Nor	nor	R	R[rd] = $\sim (R[rs] \mid R[rt])$	0100
Set Less Than	slt	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0	0101
Shift Left Logical	sll	I	R[rs] = R[rs] << Shamt	0110
Shift Right Logical	srl	I	R[rs] = R[rs] >> Shamt	0111
Jump	j	J	PC = PC + 1 + JumpAddr	1000
Jump and Link	jal	I	M[sp + 1] = PC + 1; PC = R[rs]	1001
Load Word	lw	R	R[rs] = M[R[rt]]	1010
Store Word	sw	R	M[R[rt]] = R[rs]	1011
Branch On Equal	beq	R	if(R[rs]==R[rt]): PC = PC+1+ R[00]	1100
Branch On Not Equal	bne	R	if(R[rs]!=R[rt]): PC = PC+1+ R[00]	1101
Add Immediate	addi	I	R[rs] = R[rs] + Imm	1110
Load Immediate	li	I	R[rs] = Imm	1111

Basic Instruction Formats

R	opcode (4 bits)	rs (2 bits)	rt (2 bits)

I	opcode (4 bits)	rs (2 bits)	immediate (2 bits)

J	opcode (4 bits)	address (4 bits)

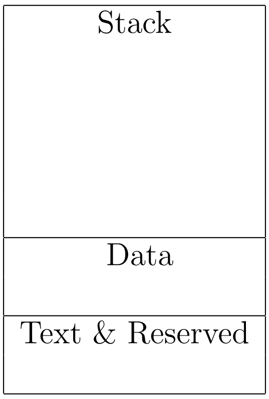
Pseudo Instruction Set

NAME	MNEMONIC	OPERATION
Add Large	addl	R[rs] = R[rs] + LargeDecimal
Load Immediate Large	lil	R[rs] = R[rs] + LargeImm
Shift Left Logical Large	slll	R[rs] = R[rs] << LargeDecimal
Shift Right Logical Large	srll	R[rs] = R[rs] >> LargeDecimal

Register Name, Number, Use, Call Convention

NAME	NUMBER	BINARY	USE	PRESERVED ACROSS A CALL?
$r0	0	00	General	No
$r1	1	01	General	No
$r2	2	10	General	No
$sp	3	11	Stack Pointer	Yes

Memory Allocation



Stack: 8 bits x 64
Data: 8 bits x 64
Text & Reserved: 8 bits x 64

Opcodes, Base Conversion, ASCII Symbols

OPCODE	Binary	Decimal	Hexadecimal	ASCII Character
Move	0000	0	0	NUL
Add	0001	1	1	SOH
And	0010	2	2	STX
Not	0011	3	3	ETX
Nor	0100	4	4	EOT
Set Less Than	0101	5	5	ENQ
Shift Left Logical	0110	6	6	ACK
Shift Right Logical	0111	7	7	BEL
Jump	1000	8	8	BS
Jump and Link	1001	9	9	HT
Load Word	1010	10	a	LF
Store Word	1011	11	b	VT
Branch On Equal	1100	12	c	FF
Branch On Not Equal	1101	13	d	CR
Add Immediate	1101	14	e	SO
Load Immediate	1111	15	f	SI