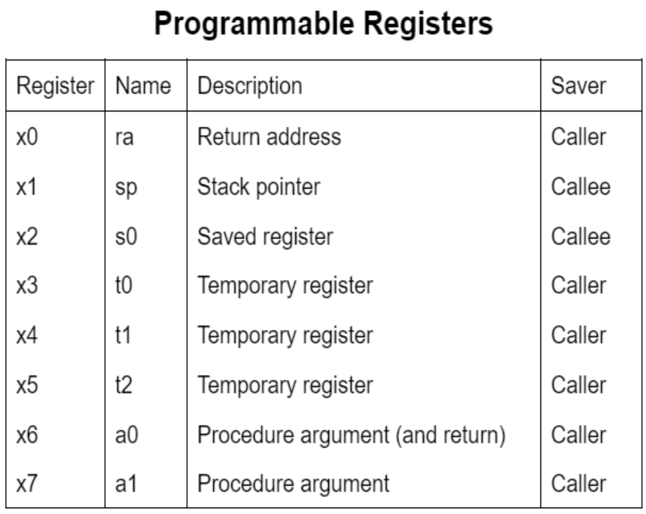
LIME Reference

Base Instructions

Instruction	Name	Instruction Type	func 4	Opcode	Description	Note
add	add	3R	0000	000	R[rd] = R[r1] + R[r2]	
sub	subtract	3R	0001	000	R[rd] = R[r1] - R[r2]	
and	and	3R	0010	000	R[rd] = R[r1] & R[r2]	
or	or	3R	0011	000	R[rd]= R[r1] | R[r2]	
xor	xor	3R	0100	000	R[rd] = R[r1] ^ R[r2]	
sll	shift left logical	3R	0101	000	R[rd]= R[r1] << R[r2]	
srl	shift right logical	3R	0110	000	R[rd]= R[r1] >> R[r2]	
sla	shift left arithmetic	3R	0111	000	R[rd]= R[r1] << R[r2]	sign extends
sra	shift right arithmetic	3R	1000	000	R[rd]= R[r1] >> R[r2]	sign extends
addi	ADD Immediate	2RI	0000	001	R[rd] = R[r1] + IG(UI,imm)	Special Note: The 3-bit immediates provided in the instructions for this type are appended to the end of the 16-bit immediate in the non-programmable UI register (UI stands for Upper Immediate).
subi	SUB Immediate	2RI	0001	001	R[rd] = R[r1] - IG(UI,imm)	
andi	AND Immediate	2RI	0010	001	R[rd] = R[r1] & IG(UI,imm)	
ori	OR Immediate	2RI	0011	001	R[rd] = R[r1] | IG(UI,imm)	
xori	XOR Immediate	2RI	0100	001	R[rd] = R[r1] ˆ IG(UI,imm)	
slli	Shift Left Logical Imm	2RI	0101	001	R[rd] = R[r1] << IG(UI,imm)	
srli	Shift Right Logical Imm	2RI	0110	001	R[rd] = R[r1] >> IG(UI,imm)	
slai	Shift Left Arith Imm	2RI	0111	001	R[rd] = R[r1] << IG(UI,imm)	
srai	Shift Right Arith Imm	2RI	1000	001	R[rd] = R[r1] >> IG(UI,imm)	
lw	Load Word	2RI	1001	001	R[rd] = M[2*(R[r1]+IG(UI,imm))]	
sw	Store Word	2RI	1010	001	M[2*(R[r1]+IG(UI,imm))] = R[rd]	
beq	Branch if equal	2RI	1100	001	if(R[rd]==R[r1]) PC+= 2*IG(UI,imm)	
blt	Branch if less than	2RI	1101	001	if(R[rd]<R[r1]) PC+= 2*IG(UI,imm)	

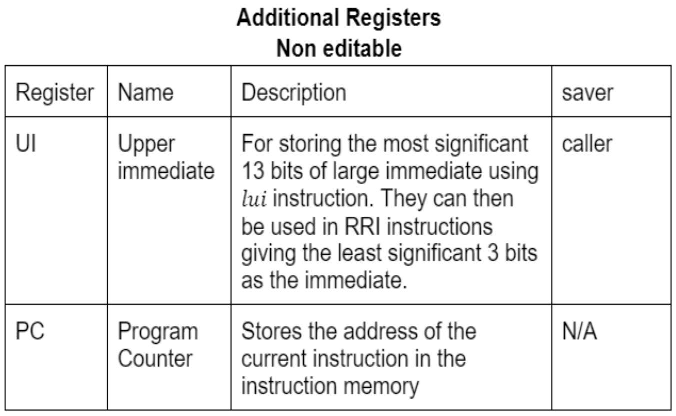
LIME Reference

bne	Branch if not equal	2RI	1110	001	if(R[rd]!=R[r1]) PC+= IG(UI,imm)	
bge	Branch if greater or equal	2RI	1111	001	if(R[rd]>=R[r1]) PC+= 2*IG(UI,imm)	
jalr	Jump And Link Reg	2RI	1011	001	R[rd] = PC + 2 PC = 2*(IG(UI,imm))	
jal	Jump and Link	UJ	-	100	Reg[IR[15:13]] = PC + 1 PC = imm[9:0]	moves execution to the instruction at address 2*(immediate 9:0)
lui	load upper immediate	L	-	011	UI = immediate[12:0]	sets non-programmable UI (upper immediate) register to be used with 2RI instructions



Programmable Registers

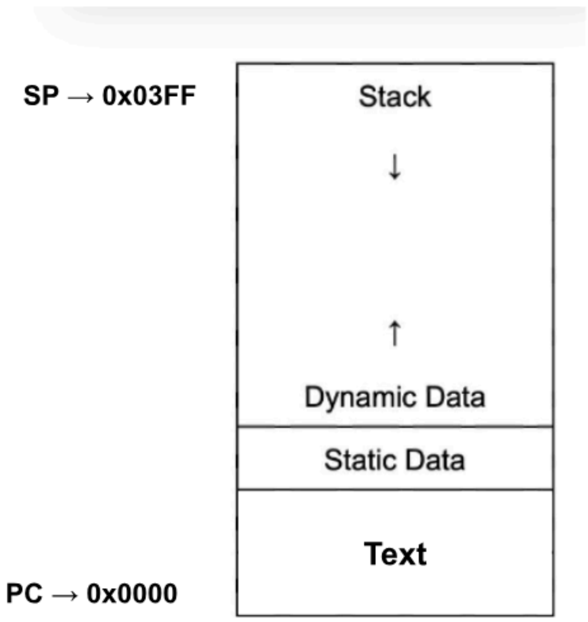
Register	Name	Description	Saver
x0	ra	Return address	Caller
x1	sp	Stack pointer	Callee
x2	s0	Saved register	Callee
x3	t0	Temporary register	Caller
x4	t1	Temporary register	Caller
x5	t2	Temporary register	Caller
x6	a0	Procedure argument (and return)	Caller
x7	a1	Procedure argument	Caller



Additional Registers
Non editable

Register	Name	Description	saver
UI	Upper immediate	For storing the most significant 13 bits of large immediate using *lui* instruction. They can then be used in RRI instructions giving the least significant 3 bits as the immediate.	caller
PC	Program Counter	Stores the address of the current instruction in the instruction memory	N/A

Memory Allocation


SP → 0x03FF
Stack
↓
↑
Dynamic Data
Static Data
Text
PC → 0x0000