RISC-V 32-Bit Assembly Cheatsheet

(for RV32I Instruction Set)

Arithmetic Instructions

Instruction	Description
add	Add
addi	Add Immediate
neg	Negate (pseudoinstruction)
sub	Subtract
mul	Multiply
mulh	Multiply High
mulhu	Multiply High Unsigned
mulhsu	Multiply High Signed Unsigned
div	Divide
rem	Remainder

Bitwise Logic Instructions

Instruction	Description
and	Logical AND
andi	AND Immediate
not	Logical NOT (pseudoinstruction)
or	Logical OR
ori	OR Immediate
xor	Logical XOR
xori	XOR Immediate

Shift Instructions

Instruction	Description
sll	Shift Left Logical
slli	Shift Left Logical Immediate
srl	Shift Right Logical
srli	Shift Right Logical Immediate
sra	Shift Right Arithmetic
srai	Shift Right Arithmetic Immediate

Load Immediate Instructions

Instruction	Description
li	Load Immediate (pseudoinstruction)
lui	Load Upper Immediate
auipc	Add Upper Immediate to PC

Load and Store Instructions

Instruction	Description
lw	Load Word
lh	Load Half
lhu	Load Half Unsigned
lb	Load Byte
lbu	Load Byte Unsigned
la	Load Symbol Address (pseudoinstruction)
sw	Store Word
sh	Store Half
sb	Store Byte

Jump and Function Instructions

Instruction	Description
j	Jump (pseudoinstruction)
jal	Jump and Link
jalr	Jump and Link Register
call	Call Function (pseudoinstruction)
ret	Return from Function (pseudoinstruction)

Branch Instructions

Instruction	Description
beq	Branch Equal
beqz	Branch Equal Zero (pseudoinstruction)
bne	Branch Not Equal
bnez	Branch Not Equal Zero (pseudoinstruction)
blt	Branch Less Than
bltu	Branch Less Than Unsigned
bltz	Branch Less Than Zero (pseudoinstruction)
bgt	Branch Greater Than (pseudoinstruction)
bgtu	Branch Greater Than Unsigned (pseudoinstruction)
bgtz	Branch Greater Than Zero (pseudoinstruction)
ble	Branch Less or Equal (pseudoinstruction)
bleu	Branch Less or Equal Unsigned (pseudoinstruction)
blez	Branch Less or Equal Zero (pseudoinstruction)
bge	Branch Greater or Equal
bgeu	Branch Greater or Equal Unsigned
bgez	Branch Greater or Equal Zero (pseudoinstruction)

Set Instructions

Instruction	Description
slt	Set Less Than
slti	Set Less Than Immediate
sltu	Set Less Than Unsigned
sltiu	Set Less Than Immediate Unsigned
seqz	Set Equal Zero (pseudoinstruction)
snez	Set Not Equal Zero (pseudoinstruction)
sltz	Set Less Than Zero (pseudoinstruction)
sgtz	Set Greater Than Zero (pseudoinstruction)

Counter Instructions

Instruction	Description
rdcycle	CPU Cycle Count (pseudoinstruction)
rdcycleh	CPU Cycle Count High (pseudoinstruction)
rdtime	Current Time (pseudoinstruction)
rdtimeh	Current Time High (pseudoinstruction)
rdinstret	CPU Instructions Retired (pseudoinstruction)
rdinstreth	CPU Instructions Retired High (pseudoinstruction)

Miscellaneous Instructions

Instruction	Description
ebreak	Environment Break (Debugger Call)
ecall	Environment Call (OS Function)
fence	I/O Ordering
mv	Copy Register (pseudoinstruction)
nop	No Operation (pseudoinstruction)

Instruction Terminology

Term	Description
imm	Immediate value (normally sign-extended)
mem	Memory
(p)	Pseudoinstruction
pc	Program counter
pc+4	Next instruction on RV32
ra	Return address register (x1)
rd	Destination register
rs1	First source register
rs2	Second source register
symbol	Symbol (may be a label in assembly)

RV32 ABI Registers

ABI Name	Description
zero	Always 0 (zero)
ra	Return address
sp	Stack pointer
gp	Global pointer*
tp	Thread pointer*
t0	Temporary
t1	Temporary
t2	Temporary
fp (s0)	Frame pointer†
s1	Saved register
a0	Function argument‡
a1	Function argument‡
a2	Function argument
a3	Function argument
a4	Function argument
a5	Function argument
a6	Function argument
a7	Function argument
s2	Saved register
s3	Saved register
s4	Saved register
s5	Saved register
s6	Saved register
s7	Saved register
s8	Saved register
s9	Saved register
s10	Saved register
s11	Saved register
t3	Temporary
t4	Temporary
t5	Temporary
t6	Temporary

RISC-V Concepts

- **Extensions:** RISC-V is modular, allowing for different extensions (e.g., M for Multiply/Divide, A for Atomic, etc.) to add functionality.
- **Base Instruction Set:** This cheat sheet focuses on the base integer instruction set (RV32I).
- **Pseudoinstructions:** These are simplified instructions that are translated into one or more actual RISC-V instructions by the assembler.

Usage Examples

- **Simple Arithmetic Operations:**

```
li t0, 10          # Load immediate 10 into t0
li t1, 20          # Load immediate 20 into t1
add t2, t0, t1     # t2 = t0 + t1 (30)
sub t3, t1, t0     # t3 = t1 - t0 (10)
mul t4, t2, t3     # t4 = t2 * t3 (300)
ecall              # Exit program
```

- **Branching Based on Comparison:**

```
li t0, 5           # t0 = 5
li t1, 10          # t1 = 10
blt t0, t1, less_than_label  # If t0 < t1, jump to less_than_label
j end_label        # Else, jump to end_label

less_than_label:
li a0, 1           # Set a0 to 1 (indicating true)
j end_label

end_label:
li a0, 0           # Set a0 to 0 (indicating false)
ecall              # Exit program
```

- **Function Call and Return:**

```
jal ra, function_label  # Jump to function_label and save return address
li a0, 0                # Set return value to 0
ecall                   # Exit program

function_label:
addi sp, sp, -16        # Allocate stack space
sw ra, 12(sp)           # Save return address
li t0, 42               # Perform some operations
lw ra, 12(sp)           # Restore return address
addi sp, sp, 16         # Deallocate stack space
jalr ra, 0(ra)          # Return to caller
```