

Assembly Operations

The right column gives the command followed by its arguments. Arguments beginning with *r* are registers. An *imm* argument is a number; its subscript is the maximum size in bits. For commands with a result, *rd* is the destination.

Basic operations

Add	<code>add rd, rs, rt</code>
Add immediate	<code>addi rd, rs, imm₁₆</code>
Subtract	<code>sub rd, rs, rt</code>
Multiply*	<code>mulo rd, rs, rt</code>
Divide <i>s</i> by <i>t</i> *	<code>div rd, rs, rt</code>
Remainder*	<code>rem rd, rs, rt</code>
Load immediate*	<code>li rd, imm₃₂</code>
System call (see table below)	<code>syscall</code>
<code>syscall</code> is controlled by the contents of register <code>\$v0</code> :	
1	Print int stored in <code>\$a0</code>
4	Print string whose address is stored in <code>\$a0</code>
5	Read int into <code>\$v0</code>
10	Exit simulation
11	Print character stored in <code>\$a0</code>

Comparison, branch, and jump instructions

Unconditional branch*	<code>b label</code>
Branch if equal	<code>beq rs, rt, label</code>
Branch if not equal	<code>bne rs, rt, label</code>
Branch if greater than*	<code>bgt rs, rt, label</code>
Branch if greater than or equal*	<code>bge rs, rt, label</code>
Branch if less than*	<code>blt rs, rt, label</code>
Branch if less than or equal*	<code>ble rs, rt, label</code>

Memory instructions

Shift left (by immediate; fill with 0s)	<code>sll rd, rs, imm₅</code>
“the following goes in the text segment”	<code>.text</code>
“the following goes in the data segment”	<code>.data</code>
“store str as a null-terminated string”	<code>.ascii str</code>
“store these words in memory”	<code>.word w1 w2 ...</code>
“reserve <i>n</i> bytes of space”	<code>.space n</code>
“align on 2 ^{<i>n</i>} -byte boundary”	<code>.align n</code>
Load address (not contents) *	<code>la rd, address</code>
Load byte	<code>lb rd, address</code>
Store byte	<code>sb rt, address</code>
Load word	<code>lw rd, address</code>
Store word	<code>sw rd, address</code>

Addresses for memory instructions can have any of the following forms:

(register)	contents of register
imm(register)	contents of register + immediate
label	address of label