# 1. Name and Usage of 32 registers in MIPS

|  |  |  |
| --- | --- | --- |
| Register number | Name | Usage |
| $0 | $zero | constant 0 |
| $1 | $at | Reserved for pseudo-instructions |
| $2 - $3 | $v0, $v1 | Return values from functions |
| $4 - $7 | $a0 - $a3 | Arguments to functions - not preserved by subprograms |
| $8 - $15 | $t0 - $t7 | Temporary data, not preserved by subprograms |
| $16 - $23 | $s0 - $s7 | Saved registers, preserved by subprograms |
| $24 - $25 | $t8 - $t9 | More temporary registers, not preserved by subprograms |
| $26 - $27 | $k0 - $k1 | Reserved for OS kernel. Do not use. |
| $28 | $gp | Global Area Pointer (base of global data segment) |
| $29 | $sp | Stack Pointer |
| $30 | $fp | Frame Pointer |
| $31 | $ra | Return Address |
| $f0 - $f3 | - | Floating point return values |
| $f4 - $f10 | - | Temporary registers, not preserved by subprograms |
| $f12 - $f14 | - | First two arguments to subprograms, not preserved by subprograms |
| $f16 - $f18 | - | More temporary registers, not preserved by subprograms |
| $f20 - $f31 | - | Saved registers, preserved by subprograms |

# 2. Register PC, HI, LO

The multiplication instructions treat *HI* and *LO* as a logical 64-bit register, where the high-order 32 bits are in the *HI* register and the low-order 32 bits are in the *LO* register.

MUL rd, rs, rt ; rd = rs \* rt, corrupts HI and LO

MULT rs, rt ; HI:LO = rs \* rt (signed)

MULTU rs, rt ; HI:LO = rs \* rt (unsigned)

The next group of multiplication instructions performs accumulation.

MADD rs, rt ; HI:LO += rs \* rt (signed)

MADDU rs, rt ; HI:LO += rs \* rt (unsigned)

MSUB rs, rt ; HI:LO -= rs \* rt (signed)

MSUBU rs, rt ; HI:LO -= rs \* rt (unsigned)

They divide a 32-bit value by another 32-bit value and store the quotient and remainder in in *HI* and *LO*.

DIV rd, rs, rt ; LO = rs / rt, HI = rs % rt (signed)

DIVU rd, rs, rt ; LO = rs / rt, HI = rs % rt (unsig

How do we get the answer out? (And how do you put the initial values in, if you are using MADD or MSUB?)

MFHI rd ; rd = HI "move from HI"

MFLO rd ; rd = LO "move from LO"

MTHI rs ; HI = rs "move to HI"

MTLO rs ; LO = rs "move to LO"

PC = Program Counter, a CPU register which keeps track of execution

# 3. MIPS instruction format

