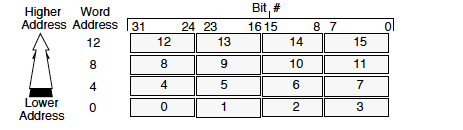
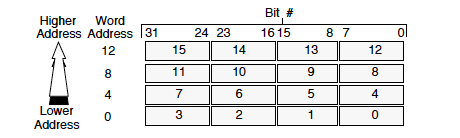
MIPS R4400

The R4400 microprocessor is a developed version of the R4000 that was developed by MIPS Computer Systems in November 1992. Both of them are Reduced Instruction Set Computer (RISC); they implement the MIPS Instruction Set Architecture (ISA). Although, the RISC microprocessors has simple instruction set, it was not the main goal of their development, but the better performance that they provide. They were expected to replace the complex instruction set computer microprocessors due to the benefits that are the result of less complexity; they can be implemented quicker than CISC processors, as it is easier to debug simplified architecture that does not have microcode than the complex architecture which has microcode, they utilize the chip area more effectively, their simplicity also help the user (programmer) to use them, and because of the simplicity of RISC processors, it can be implemented in fewer transistor than CISC processors which require more.

The primary caches in the R4400 reside on chip; each is capable of holding 16 Kbytes which is double the capacity of the primary caches of the R4000 (8 Kbytes each). However, the primary caches of the R4400 could be increased to hold 32 Kbytes. There are four data formats which the R4400 uses, an 8-bit byte, a 16-bit halfword, a 32-bit word, and a 64-bit doubleword; byte ordering in the halfword data format, the word data format, and the doubleword data format can be big endian or little endian.

 Big-Endian Byte Ordering



Little-Endian Byte Ordering

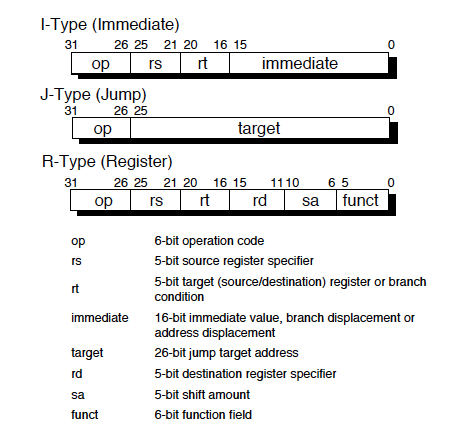
The R4400 uses byte addressing for halfword, word, and doubleword. Halfword accesses are aligned on an even byte boundary (0,2,4 …). However, word accesses are aligned on a byte boundary that is divisible by 4 such as, (0,4,8 …); doubleword accesses are aligned on a byte boundary that is divisible by 8 (0,8,16 …).

The R4400 has 32 general purpose registers each is 64-bit, but it could operates as 32-bit processor as well, in the case of operating as 32-bit processor the general purpose registers will be 32-bits wide. The register r0 is hardwired to a value of zero; it could be used as the target register to discard the result of an instruction. On the other hand, r31 is used by the Link and Jump instructions, so it is not supposed to be used by any other instruction. The r4400 has 3 special purpose registers: PC (Program Counter register), HI (Multiply and Divide register higher result), LO (Multiply and Divide register lower result). The special purpose register HI, and LO store the result of integer multiply operations, or the LO stores quotient and HI stores the remainder of integer divide operations.

The CPU has two data types: floating-point and integers. It has an Arithmetic Logic Unit (ALU) that is reasonable for the operations of the integers, and an IEEE 754-1985-compliant Floating-Point Unit for the floating point which can operate in a 32-bit mode, or 64-bit mode. In order to store single-precision floating-point numbers the 32 registers will be 32 bits wide, but to store double-precision floating-point numbers the registers are paired, so there will be 16 floating-point registers instead of 32.

The R4400 is a RISC processor. As a result, it provides five basic types of instructions which are: data movement, simple arithmetic, control such as branch and jump, multiplication and division other instructions to save the PC or save other registers. In addition, RISC architecture let compilers have the optimal working environment.

There are three instruction formats for the R4400 which are: I-type (immediate), J-type (jump), and R-type (register). Each one of the formats contains different instructions. Decoding the instruction is highly simplified by having few instruction formats. As a consequence, the R4400 is a simple yet powerful CPU.

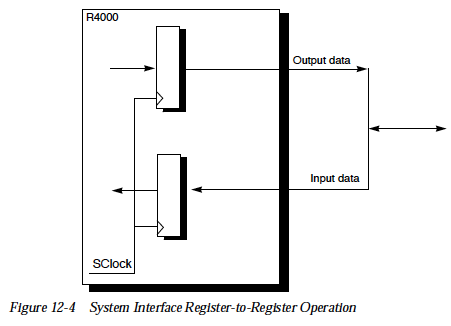


The programmer can use:

1. Immediate addressing
2. Register addressing
3. Direct addressing
4. Indirect addressing
5. Base addressing
6. Indexed addressing

These addressing modes are provided by the assembler; the ISA provides only base addressing.

“Figure 12-4 shows the System interface operates from register to register. That is, processor outputs come directly from output registers and begin to change with the rising edge of SClock. Processor inputs are fed directly to input registers that latch these input signals with the rising edge of SClock. This allows the System interface to run at the highest possible clock frequency.” ( Heinrich, Joe. [*MIPS R4000 Microprocessor User's Manual*](https://web.archive.org/web/20110607074823/http:/techpubs.sgi.com/library/manuals/2000/007-2489-001/pdf/007-2489-001.pdf), Second Edition)



The R4400 has some unusual features, such as not having a Program Statues Word (PSW) which contains bits that represent the CPU current state. Additionally, r0 is hardwired to hold a value of zero, so if it is the target of any instruction the result of the instruction will be discarded; r31 should be used for the Link and Jump instructions only.

Many companies use the R4400 in their systems including, Carrera Computers in Windows NT, and in Nile Series servers by Pyramid Technology, and the AEC to be their processor. In addition, the R4000 and R4400 were considered to be from the first 64-bit processors and they were the first to use the MIPS III. “RISC philosophy assumes that high-level language programming is used, which contradicts the older CISC philosophy that assumes assembly language programming is of primary importance. The trend toward high-level language instructions has led to the development of more efficient compilers to convert high-level language instructions to machine code. Primary measures of compiler efficiency are the compactness of its generated code and the shortness of its execution time.” ( Heinrich, Joe. [*MIPS R4000 Microprocessor User's Manual*](https://web.archive.org/web/20110607074823/http:/techpubs.sgi.com/library/manuals/2000/007-2489-001/pdf/007-2489-001.pdf), Second Edition)

Appendix

Instructions Set:

LB Load Byte

LBU Load Byte Unsigned

LH Load Halfword

LHU Load Halfword Unsigned

LW Load Word

LWL Load Word Left

LWR Load Word Right

SB Store Byte

SH Store Halfword

SW Store Word

SWL Store Word Left

SWR Store Word Right

ADDI Add Immediate

ADDIU Add Immediate Unsigned

SLTI Set on Less Than Immediate

SLTIU Set on Less Than Immediate Unsigned

ANDI AND Immediate

ORI OR Immediate

XORI Exclusive OR Immediate

LUI Load Upper Immediate

ADD Add

ADDU Add Unsigned

SUB Subtract

SUBU Subtract Unsigned

SLT Set on Less Than

SLTU Set on Less Than Unsigned

AND AND

OR OR

XOR Exclusive OR

NOR NOR

MULT Multiply

MULTU Multiply Unsigned

DIV Divide

DIVU Divide Unsigned

MFHI Move From HI

MTHI Move To HI

MFLO Move From LO

MTLO Move To LO

J Jump

JAL Jump And Link

JR Jump Register

JALR Jump And Link Register

BEQ Branch on Equal

BNE Branch on Not Equal

BLEZ Branch on Less Than or Equal to Zero

BGTZ Branch on Greater Than Zero

BLTZ Branch on Less Than Zero

BGEZ Branch on Greater Than or Equal to Zero

BLTZAL Branch on Less Than Zero And Link

BGEZAL Branch on Greater Than or Equal to Zero And Link

SLL Shift Left Logical

SRL Shift Right Logical

SRA Shift Right Arithmetic

SLLV Shift Left Logical Variable

SRLV Shift Right Logical Variable

SRAV Shift Right Arithmetic Variable

LWCz Load Word to Coprocessor z

SWCz Store Word from Coprocessor z

MTCz Move To Coprocessor z

MFCz Move From Coprocessor z

CTCz Move Control to Coprocessor z

CFCz Move Control From Coprocessor z

COPz Coprocessor Operation z

BCzT Branch on Coprocessor z True

BCzF Branch on Coprocessor z False

SYSCALL System Call

BREAK Break

Works Cited

Heinrich, Joe. [*MIPS R4000 Microprocessor User's Manual*](https://web.archive.org/web/20110607074823/http:/techpubs.sgi.com/library/manuals/2000/007-2489-001/pdf/007-2489-001.pdf), Second Edition

Julia Lobur, Linda Null. The Essentials of Computer Organization and Architecture, Fourth Edition.

Hyong-youb Kim and Scott Rixner. [EuroSys '06](http://www.cs.kuleuven.be/conference/EuroSys2006) Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006 <https://dl-acm-org.ezproxy.una.edu/citation.cfm?id=258686 (Links to an external site.)Links to an external site.>