Integrated Device Technology, Inc.

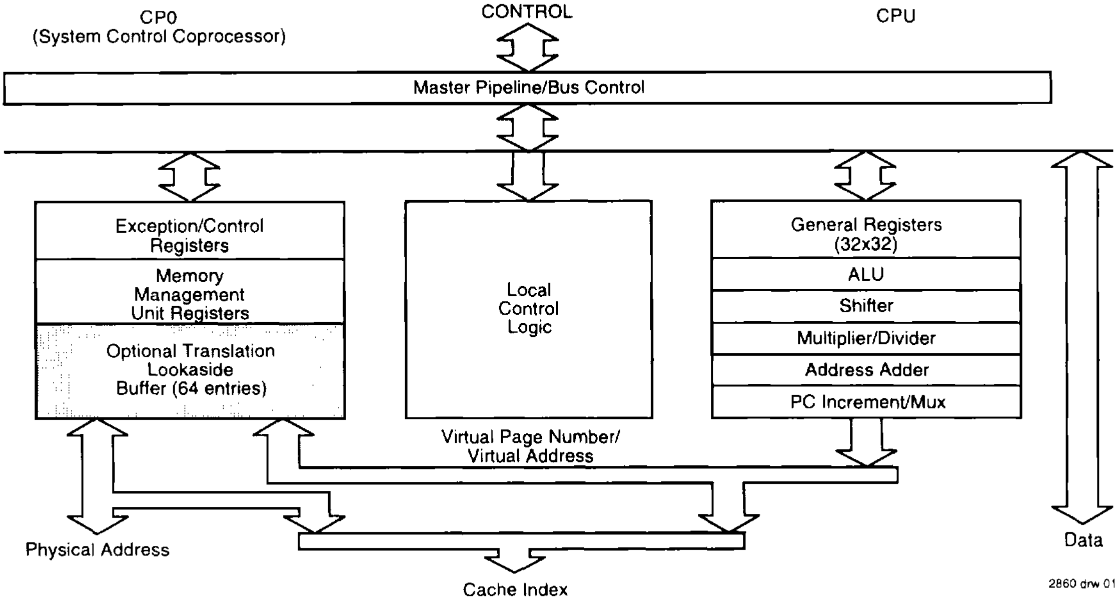
RISC CPU CORE

R3000A Core for RISController Devices

FEATURES:

- Enhanced instruction set compatible R3000A Core for integrated RISControllers
- Integrates well with R3010A Core Hardware Floating Point Accelerator
- Full 32-bit Operation—Thirty-two 32-bit registers and all instructions and addresses are 32-bit.
- Efficient Pipelining—The CPU's 5-stage pipeline design assists in obtaining an execution rate approaching one instruction per cycle. Pipeline stalls and exceptions are handled precisely and efficiently.
- Integrated Cache Control for On-Chip Caches—The CPU core contains a high-bandwidth memory interface that handles separate Instruction and Data Caches. Both caches are accessed during a single CPU cycle. All cache control is integrated into the core, allowing high-speed execution.
- "E" versions feature Memory Management Unit, including a fully-associative, 64-entry Translation Look-aside Buffer (TLB). This provides fast address translation for virtual-to-physical memory mapping of the 4GB virtual address space.
- Dynamically able to switch between Big- and Little-Endian byte ordering conventions.
- Software compatible with all R3000 devices. This insures a wide range of development support, including compilers, operating systems, libraries, and applications software.
- High-speed 0.6µ CMOS technology.
- 50MHz clock rates yield up to 40VUPS sustained throughput.
- Supports independent multi-word block refill of both the instruction and data caches.
- Supports concurrent refill and execution of instructions.
- Partial word stores executed as read-modify-write operations.
- 6 external interrupt inputs, 2 software interrupts, with single cycle latency to exception handler routine.

R3000A CORE BLOCK DIAGRAM


CP0
(System Control Coprocessor)
CONTROL
CPU
Master Pipeline/Bus Control
Exception/Control Registers
Memory Management Unit Registers
Optional Translation Lookaside Buffer (64 entries)
Local Control Logic
General Registers (32x32)
ALU
Shifter
Multiplier/Divider
Address Adder
PC Increment/Mux
Virtual Page Number/
Virtual Address
Physical Address
Data
Cache Index

2860 drw 01

The IDT logo is a registered trademark and Orion, R3041, R3051, R3052, R3081, R3721, R4600, RISCompiler, RISController, RISCore, RISC Subsystem, and RISC Windows are trademarks of Integrated Device Technology, Inc.

MARCH 1994

DESCRIPTION

The R3000A RISC Microprocessor Core consists of two tightly-coupled processors. The first processor is a full 32-bit CPU based on RISC (Reduced Instruction Set Computer) principles to achieve a new standard of microprocessor price/ performance. The second processor is a system control coprocessor, called CP0, containing an optional fully-associative 64-entry TLB (Translation Look-aside Buffer), MMU (Memory Management Unit) and control registers, supporting a 4GB virtual memory subsystem, and a Harvard Architecture Cache Controller achieving a bandwidth of 400MB/second using integrated cache memory.

This data sheet provides an overview of the features and architecture of the R3000A core. This core is integrated into various members of the IDT RISController family, such as the R3041, R3051, and R3081. Detail on those specific devices is found in separate data sheets and user's manuals.

R3000A CPU Registers

The R3000A CPU provides 32 general-purpose 32-bit registers, a 32-bit Program Counter, and two 32-bit registers which hold the results of integer multiply and divide operations. Only two of the 32 general registers have a special purpose: register r0 is hard-wired to the value "0", which is a useful constant, and register r31 is used as the link register in jump-and-link instructions (return address for subroutine calls).

The CPU registers are shown in Figure 2. Note that there is no Program Status Word (PSW) register shown in this figure: the functions traditionally provided by a PSW register are instead provided in the Status and Cause registers incorporated within the System Control Coprocessor (CP0).

Instruction Set Overview

All R3000A instructions are 32 bits long, and there are only three instruction formats. This approach simplifies instruction decoding, thus minimizing instruction execution time. The R3000A core initiates a new instruction on every run cycle, and is able to complete an instruction on almost every clock cycle. The only exceptions are the Load instructions and Branch instructions, which each have a single cycle of latency associated with their execution. Note, however, that in the majority of cases the compilers are able to fill these latency cycles with useful instructions which do not require the result of the previous instruction. This effectively eliminates these latency effects.

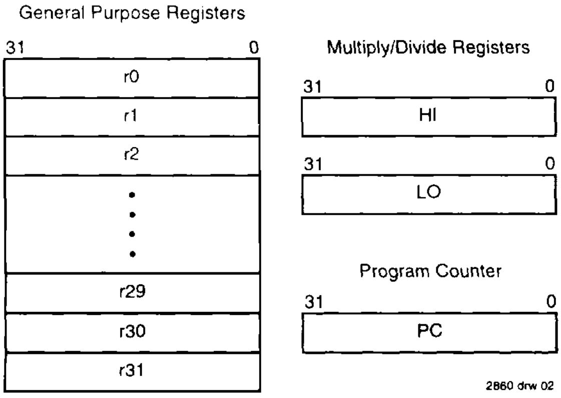


Figure 2. R3000A CPU Registers

The actual instruction set of the CPU was determined after extensive simulations to determine which instructions should be implemented in hardware, and which operations are best synthesized in software from other basic instructions.

The R3000A instruction set can be divided into the following groups:

- **Load/Store** instructions move data between memory and general registers. They are all I-type instructions, since the only addressing mode supported is base register plus 16-bit, signed immediate offset.

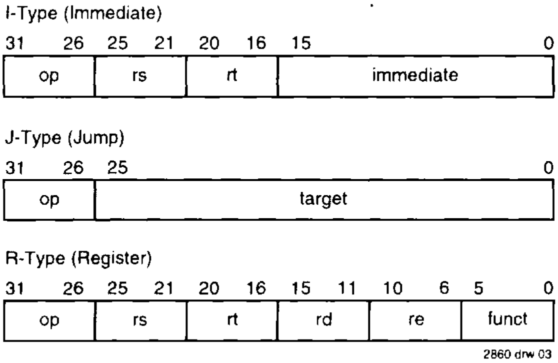


Figure 3. R3000A Instruction Formats

The Load instruction has a single cycle of latency, which means that the data being loaded is not available to the instruction immediately after the load instruction. The compiler will fill this delay slot with either an instruction which is not dependent on the loaded data, or with a NOP instruction. There is no latency associated with the store instruction.

Loads and Stores can be performed on byte, half-word, word, or non-aligned word data (32-bit data not aligned on a modulo-4 address). The CPU cache is constructed as a write-through cache.

- **Computational** instructions perform arithmetic, logical and shift operations on values in registers. They occur in both R-type (both operands and the result are registers) and I-type (one operand is a 16-bit immediate) formats. Note that computational instructions are three operand instructions; that is, the result of the operation can be stored into a different register than either of the two operands. This means that operands need not be overwritten by arithmetic operations. This results in a more efficient use of the large register set.

- **Jump and Branch** instructions change the control flow of a program. Jumps are always to a paged absolute address formed by combining a 26-bit target with four bits of the Program counter (J-type format, for subroutine calls), or 32-bit register byte addresses (R-type, for returns and dispatches). Branches have 16-bit offsets relative to the program counter (I-type). Jump and Link instructions save a return address in Register 31. The R3000A instruction set features a number of branch conditions. Included is the ability to compare a register to zero and branch, and also the ability to branch based on a comparison between two registers. Thus, net performance is increased since software does not have to perform arithmetic instructions prior to the branch to set up the branch conditions.
- **Coprocessor** instructions perform operations in the coprocessors. Coprocessor Loads and Stores are I-type. Coprocessor computational instructions have coprocessor-dependent formats (see coprocessor manuals).
- **Coprocessor 0** instructions perform operations on the System Control Coprocessor (CP0) registers to manipulate the memory management and exception handling facilities of the processor.
- **Special** instructions perform a variety of tasks, including movement of data between special and general registers, system calls, and breakpoint. They are always R-type.

R3000A INSTRUCTION SUMMARY

OP	Description	OP	Description
	Load/Store Instructions		**Multiply/Divide Instructions**
LB	Load Byte	MULT	Multiply
LBU	Load Byte Unsigned	MULTU	Multiply Unsigned
LH	Load Halfword	DIV	Divide
LHU	Load Halfword Unsigned	DIVU	Divide Unsigned
LW	Load Word	MFHI	Move From HIGH
LWL	Load Word Left	MTHI	Move To HIGH
LWR	Load Word Right	MFLO	Move From LOW
SB	Store Byte	MTLO	Move To LOW
SH	Store Halfword		
SW	Store Word		**Jump and Branch Instructions**
SWL	Store Word Left	J	Jump
SWR	Store Word Right	JAL	Jump and Link
		JR	Jump to Register
	Arithmetic Instructions	JALR	Jump and Link Register
	(ALU Immediate)	BEQ	Branch on Equal
ADDI	Add Immediate	BNE	Branch on Not Equal
ADDIU	Add Immediate Unsigned	BLEZ	Branch on Less than or Equal to Zero
SLTI	Set on Less Than Immediate	BGTZ	Branch on Greater Than Zero
SLTIU	Set on Less Than Immediate	BLTZ	Branch on Less Than Zero
	Unsigned	BGEZ	Branch on Greater than or
ANDI	AND Immediate		Equal to Zero
ORI	OR Immediate	BLTZAL	Branch on Less Than Zero and Link
XORI	Exclusive OR Immediate	BGEZAL	Branch on Greater than or Equal to
LUI	Load Upper Immediate		Zero and Link
			Special Instructions
	Arithmetic Instructions	SYSCALL	System Call
	(3-operand, register-type)	BREAK	Break
ADD	Add		
ADDU	Add Unsigned		**Coprocessor Instructions**
SUB	Subtract	LWCz	Load Word from Coprocessor
SUBU	Subtract Unsigned	SWCz	Store Word to Coprocessor
SLT	Set on Less Than	MTCz	Move To Coprocessor
MFCz	Move From Coprocessor		
SLTU	Set on Less Than Unsigned	CTCz	Move Control to Coprocessor
AND	AND	CFCz	Move Control From Coprocessor
OR	OR	COPz	Coprocessor Operation
XOR	Exclusive OR	BCzT	Branch on Coprocessor z True
NOR	NOR	BCzF	Branch on Coprocessor z False
	Shift Instructions		**System Control Coprocessor**
SLL	Shift Left Logical		**(CPO) Instructions**
SRL	Shift Right Logical	MTC0	Move To CPo
SRA	Shift Right Arithmetic	MFC0	Move From CPo
SLLV	Shift Left Logical Variable	TLBR	Read indexed TLB entry
SRLV	Shift Right Logical Variable	TLBWI	Write Indexed TLB entry
SRAV	Shift Right Arithmetic Variable	TLBWR	Write Random TLB entry
		TLBP	Probe TLB for matching entry
		RFE	Restore From Exception

2860 tbl 01

Table 1 lists the instruction set of the R3000A processor core.

R3000A System Control Coprocessor (CP0)

The R3000A core can operate with up to four tightly-coupled coprocessors (designated CP0 through CP3). The System Control Coprocessor (or CP0), is incorporated on the R3000A core and supports the virtual memory system and exception handling functions of the processor. The virtual memory system is implemented using a Translation Look-aside Buffer and a group of programmable registers as shown in Figure 3.

SYSTEM CONTROL COPROCESSOR (CP0) REGISTERS

The CP0 registers shown in Figure 3 are used to control the memory management and exception handling capabilities of the R3000A. Table 2 provides a brief description of the registers common to most devices using the core. Note, however, that certain devices (e.g. non-E versions, the R3081, and R3041) implement slightly different sets of these registers, as described in their user's manuals.

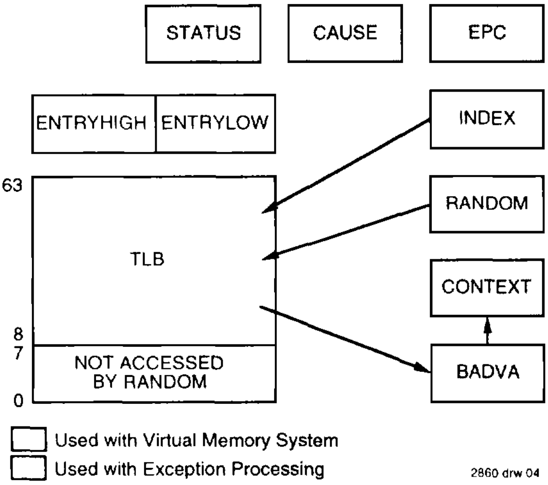


Figure 4. The System Coprocessor Registers

SYSTEM CONTROL COPROCESSOR (CPO) REGISTERS

Register	Description
EntryHIGH	HIGH half of a TLB entry
EntryLOW	LOW half of a TLB entry
Index	Programmable pointer into TLB array
Random	Pseudo-random pointer into TLB array
Status	Mode, interrupt enables, and diagnostic status info
Cause	Indicates nature of last exception
EPC	Exception Program Counter
Context	Pointer into kernel's virtual Page Table Entry array
BadVA	Most recent bad virtual address
PRId	Processor revision identification (Read only)

2860 tbl 02

Memory Management System

The R3000A has an addressing range of 4gB. However, since most R3000A systems implement a physical memory smaller than 4gB, theR3000A provides for the logical expansion of memory space by translating addresses composed in a large virtual address space into available physical memory address. The 4gB address space is divided into 2gB which can be accessed by both the users and the kernel, and 2gB for the kernel only.

The actual virtual to physical translation mechanism is either through an on-chip translation lookaside buffer (TLB), or through a fixed translation mechanism, depending on the device ("E" vs. non-"E" devices). These mechanisms are explained in the data sheets and user's manuals for those devices.

R3000A Operating Modes

TheR3000A has two operating modes: User mode and Kernel/mode. The R3000A normally operates in the User mode until an exception is detected forcing it into the Kernel mode. It remains in the Kernel mode until a Restore From Exception (RFE) instruction is executed. The manner in which memory addresses are translated or mapped depends on the operating mode of the R3000A, and whether the device implements an on-chip TLB.

User Mode—in this mode, a single, uniform virtual address space (kuseg) of 2gB is available. Each virtual address is extended with a 6-bit process identifier field to form unique virtual addresses. The actual virtual to physical address mapping is either done via a fixed translation, or through the TLB, depending on the device.

Kernel Mode—four separate segments are defined in this mode:

- *kuseg*—when in the kernel mode, references to this segment are treated just like user mode references, thus streamlining kernel access to user data.
- *kseg0*—references to this 512mB segment use cache memory but are not mapped through the optional TLB. Instead, they always map to the first 0.5gB of physical address space, whether or not the device contains an on-chip TLB.
- *kseg1*—references to this 512mB segment are not mapped through the TLB and do not use the cache. Instead, they are hard-mapped into the same 0.5gB segment of physical address space as kseg0.
- *kseg2*—references to this 1gB segment are either mapped through the TLB (with use of the cache determined by bit settings within the TLB entries) or through a predetermined mapping (non-E versions; all references go through the cache).

R3000A Pipeline Architecture

The execution of a single R3000A instruction consists of five primary steps:

1) **IF** — Fetch the instruction (I-Cache).
2) **RD** — Read any required operands from CPU registers while decoding the instruction.
3) **ALU** — Perform the required operation on instruction operands.
4) **MEM**— Access memory (D-Cache).
5) **WB** — Write back results to register file.

Each of these steps requires approximately one CPU cycle, as shown in Figure 4 (parts of some operations overlap into another cycle while other operations require only 1/2 cycle).

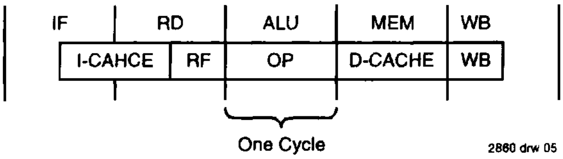


Figure 5. R3000A Instruction Pipeline

INSTRUCTION EXECUTION

The R3000A uses a 5-stage pipeline to achieve an instruction execution rate approaching one instruction per CPU cycle. Thus, execution of five instructions at a time are overlapped as shown in Figure 5.

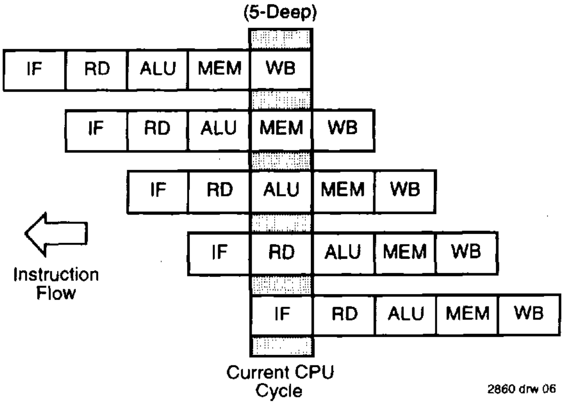


Figure 6. R3000A Execution Sequence

This pipeline operates efficiently because different CPU resources (address and data bus accesses, ALU operations, register accesses, and so on) are utilized on a non-interfering basis.

Memory System Hierarchy

A primary goal of systems employing RISC techniques is to minimize the average number of cycles each instruction requires for execution. In order to achieve this goal, RISC processors incorporate a number of RISC techniques, including a compact and uniform instruction set, a deep instruction pipeline (as described above), and utilization of optimizing compilers.

Figure 6 illustrates a memory system that supports the significantly greater memory bandwidth required to take full advantage of the R3000A's performance capabilities. The key features of this system are:

- **On-chip Cache Memory**—Local, high-speed memory (called cache memory) is used to hold instructions and data that is repetitively accessed by the CPU (for example, within a program loop) and thus reduces the number of references that must be made to the slower-speed main memory.
- **Separate Caches for data and Instructions**—Even with high-speed caches, memory speed can still be a limiting factor because of the fast cycle time of a high-performance microprocessor. The R3000A supports separate caches for instructions and data and alternates accesses of the two caches during each CPU cycle. Thus, the processor can obtain data and instructions at the cycle rate of the CPU.
- **Write Buffer**—in order to ensure data consistency, all data that is written to the data cache must also be written out to main memory. The cache write model used by the R3000A is that of a write-through cache; that is, all data written by the CPU is immediately written into the main memory. To relieve the CPU of this responsibility (and the inherent performance burden) the R3000A supports an interface to an on-chip write buffer. Thus, the R3000A core continues execution at high-speed, while the store data is retired at the slower memory rate.
- **Read Buffer**—The IDT RISController family typically incorporates an on-chip read buffer. This enables the system interface to match the speed of the high-speed execution core with the slower speed of a low-cost memory system, while still optimizing performance. This small on-chip FIFO enables the CPU to refill the cache and execute instructions even while additional instructions are being read from memory. This process is called instruction streaming.

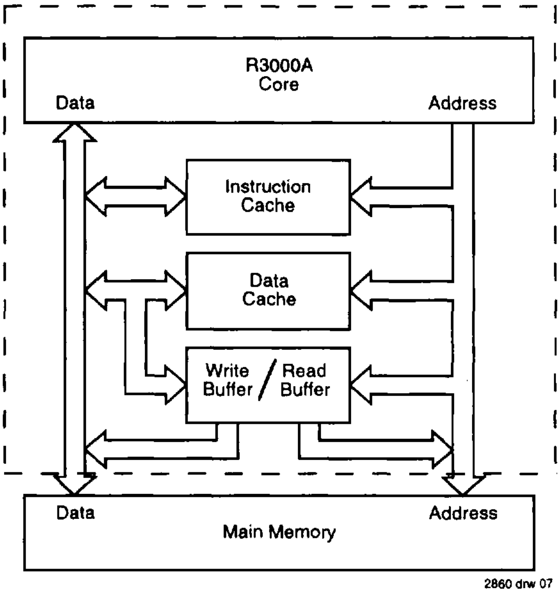


Figure 7. An R3000A System with a High-Performance Memory System

ADVANCED FEATURES

The R3000A offers a number of additional features such as the ability to swap the instruction and data caches, facilitating diagnostics and cache flushing. Another feature isolates the caches, which forces cache hits to occur regardless of the contents of the tag fields. The R3000A allows the processor to execute user tasks of the opposite byte ordering (endianness) of the operating system, and further allows parity checking to be disabled. More details on these features can be found in the various devices' Hardware User's Manuals.

Further features of the R3000A are configured by the user, in a device dependent fashion. These functions include whether byte ordering follows "Big-Endian" or "Little-Endian" protocols, particulars of the memory interface, etc.