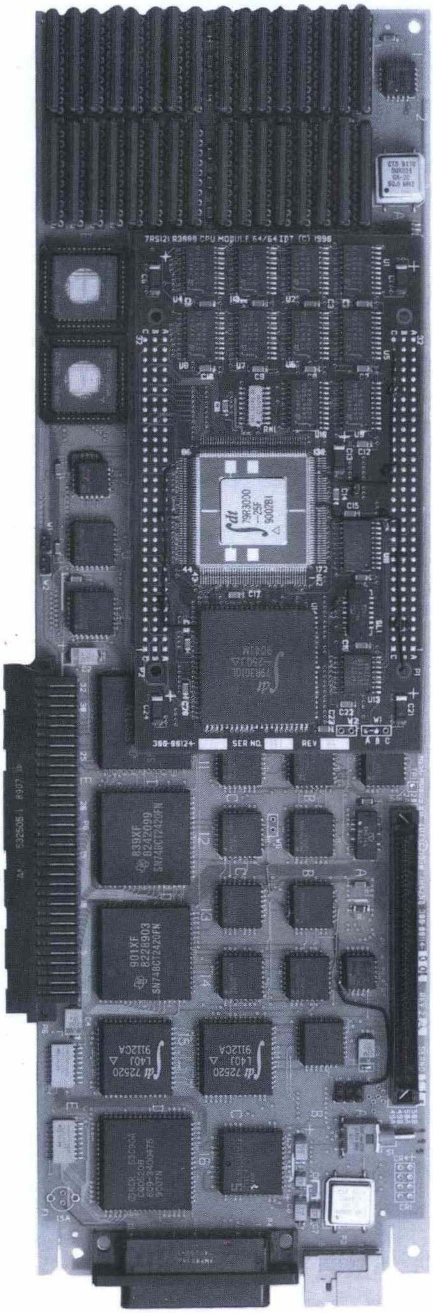
Integrated Device Technology, Inc.

Development Products Catalog

April, 1992



For R3000, R3001, R3051/52/81, and R3500 Systems

Includes:

- Prototyping and Evaluation Systems
- IDT/sim, IDT/kit, and IDT/c
- MacStation™ 3 Developer's System
- Software Licenses

Prices Effective 4/1/92 and are subject to change without notice.

Table of Contents

About This Catalog

For engineers developing software and hardware products around the R3000 Instruction Set Architecture (ISA), which includes the R3000, R3001, and R3051 family of RISControllers, IDT offers three software products, several prototyping and evaluation systems, and two versions of its MacStation bundled with IDT's software development tools. IDT is also an authorized reseller of MIPS workstations and software and has a close working relationship with many third parties who also provide support tools for this processor family. This catalog primarily focuses on products manufactured and sold directly by Integrated Device Technology.

Software Products

IDT/c IDT's optimizing ANSI C-compiler. This compiler, which uses the Gnu C front end, includes full ANSI C compatibility and highly efficient floating point emulation libraries for R3051 based systems (without hardware floating point). A unique debug control scripting language makes it easy to locate hardware problems that occur only under rare conditions. IDT/c includes the compiler, optimizer, assembler, linker, librarian, C libraries, Floating Point Libraries, and symbolic debugger.

IDT/sim IDT/sim is IDT's System Integration Manager, used to bring up new hardware and to support the symbolic debug in both the MIPS and IDT C compilers. IDT/sim is a ROMable debug kernal with extensive diagnostics built-in. It is supplied in EPROM on all IDT prototyping boards, and is available in source code for use with either the MIPS or IDT C Compilers.

IDT/kit IDT/kit is our newest software development product: The Kernel Integration ToolKit. It contains source code and compiled versions of a complete set of routines for initializing systems, servicing interrupts, handling floating point exceptions, and so forth. Also included is source code for ANSI libraries, for the Floating Point Emulation Libraries and for transcendental functions.

Prototyping Systems

Completely assembled and tested hardware systems are available for prototyping and initial software porting. All include a CPU, serial I/O, EPROM containing the IDT/sim monitor, and some amount of RAM. All have provision for simple addition of user-defined hardware. Units are available with the 3051/2 CPU, with the R3000 and R3010, and with high-performance CPU modules containing the R3000/R3010 and up to 256 KB each of I- and D- cache.

A special prototyping system is available for Laser Printer controllers. The 7RS388 "Real8" printer controller includes a PostScript™ and PCL5 emulator and plugs directly into a Canon SX laser engine. For information on the 7RS388, please contact your local IDT sales office.

MacStation Developer Systems

The MacStation Developer Systems include an R3000-based CPU card, a Macintosh II computer, a 19" monochrome monitor, the Apple extended keyboard, the complete Unix software package (MIPS RISC/os with NFS and X), and an external hard disk large enough for the Unix file systems. The Developer Systems include all three of the software tools listed above. IDT is an authorized Apple VAR for these systems.

Custom Development Programs

IDT can design and manufacture your R3000 based product for you. We manufacture very high performance CPU modules for a number of R3000 users, including Pyramid Technology and AT&T. These modules are designed according to your needs and are manufactured on 8 to 10 layer FR-4 boards using surface mount components on both sides of the board for the tightest possible layouts. We have built thousands of these modules at speeds of 33 MHz and higher.

We have also designed and manufactured R3000 and R3051 board level products for lower performance, more cost-sensitive applications such as laser printers.

In addition to our extensive design and layout experience with R3000 based products, IDT has a great deal of experience in indentifying problems as new software is brought up on in new hardware. This skill helps us bring up new systems very quickly. In production test, we use software routines that have been developed over several years of volume manufacturing experience to ensure that products operate reliably under worst-case conditions.

Third Party Development Tools

The increasing popularity of IDT's RISController family has resulted in a dramatic increase in the number of third party tools available. For information on these products, contact your local IDT sales representative.

Real-Time Operating Systems from Lynx, Ready Systems, and Wind River

Compilers from MIPS, Green Hills, and BSO Tasking

VME Boards from CES, RISQ Modular Systems, Omnibyte, and Sanders Associates

Device Simulation Models from LMSI, Zycad and HDL Systems

Peripheral Support Circuits from V3 and National Semiconductor

Page Description Language interpreters from Peerless and Adobe Systems

In-Circuit Emulators from Embedded Performance, Inc.

Logic Analyzer support from Hewlett-Packard, Fluke Instruments, and Tektronix

Integrated Device Technology, Inc.

R3051™ FAMILY EVALUATION BOARD

IDT7RS385

FEATURES:

- Complete 25 MHz RISC System
- Requires only 5V supply and terminal to operate
- Supports R3051, R3051E, R3052, or R3052E highly integrated RISC CPUs
- Supplied with complete schematics and PAL equations
- 1 MB of non-interleaved DRAM, expandable to 4 MB
- 128 KB of EPROM, expandable to 2 MB. IDT's System Integration Manager included in EPROM
- Serial and Parallel Ports
- Connectors provided for easy connection to HP Logic Analyzer
- Wire-wrap area on the board

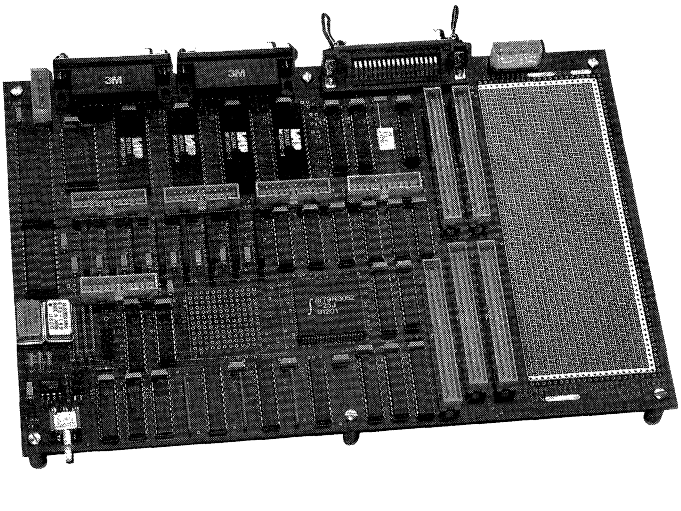
DESCRIPTION:

The IDT7RS385 is a complete working RISC system intended as a complete design example using the R3051 family of highly integrated RISC CPUs. The board requires only a simple CRT terminal and a 5 volt power supply for operation.

The board contains 128 KB of EPROM expandable to 2 MB. Main memory consists of 1 MB of DRAM in ZIPS. There are two serial ports, a free-running programmable timer, and a parallel Centronics port for high speed download of software. A set of expansion connectors permits external hardware to be connected to the board, and a wire-wrap area on the board can be used to build additional hardware without using a second board.

The EPROMs contain IDT's powerful System Integration Manager (IDT/sim), a debugging monitor that supports download of code from host systems, execution control commands, memory probing, and I/O.

The board is designed to be placed on a flat table-top surface. Standoffs are provided for physical support.



7RS385 Module. Actual Size 8.5" x 11"

R3051 is a trademark of Integrated Device Technology Inc.

SEPTEMBER 1991

COMPLETE SINGLE BOARD COMPUTER

The 7RS385 Evaluation Boards is a complete single board RISC computer, requiring only a 5 volt power supply and an RS-232 terminal for operation.

The board is designed around IDT's R3051 family of highly integrated RISC CPUs. An R3052E CPU chip (8KB I-cache and 2KB D-cache, with on-chip TLB) is included in a socket, but any member of the family can be substituted. A large wire-wrap area is available on the board for adding additional hardware. All the schematics and details of the designs are supplied with the board.

IDT's System Integration Manager (IDT/sim) is included in EPROMs on the board. This software permits downloading of code from a host system, execution control with breakpoints, in-line assembly and disassembly, and a variety of commands to control main memory, cache memory, and the internal TLB. It provides all the resources needed to bring up new hardware and software.

The 7RS385 is supplied with 1 MB of DRAM in socketed 256Kx4 ZIPs; the ZIPs can be replaced with 4 megabit devices to obtain 4 MB of DRAM on the board. Other hardware on board includes a 2681 DUART and an 8254 counter/timer; both these devices are supported with drivers in IDT/sim. A parallel Centronics port is available for higher speed download of code into the board.

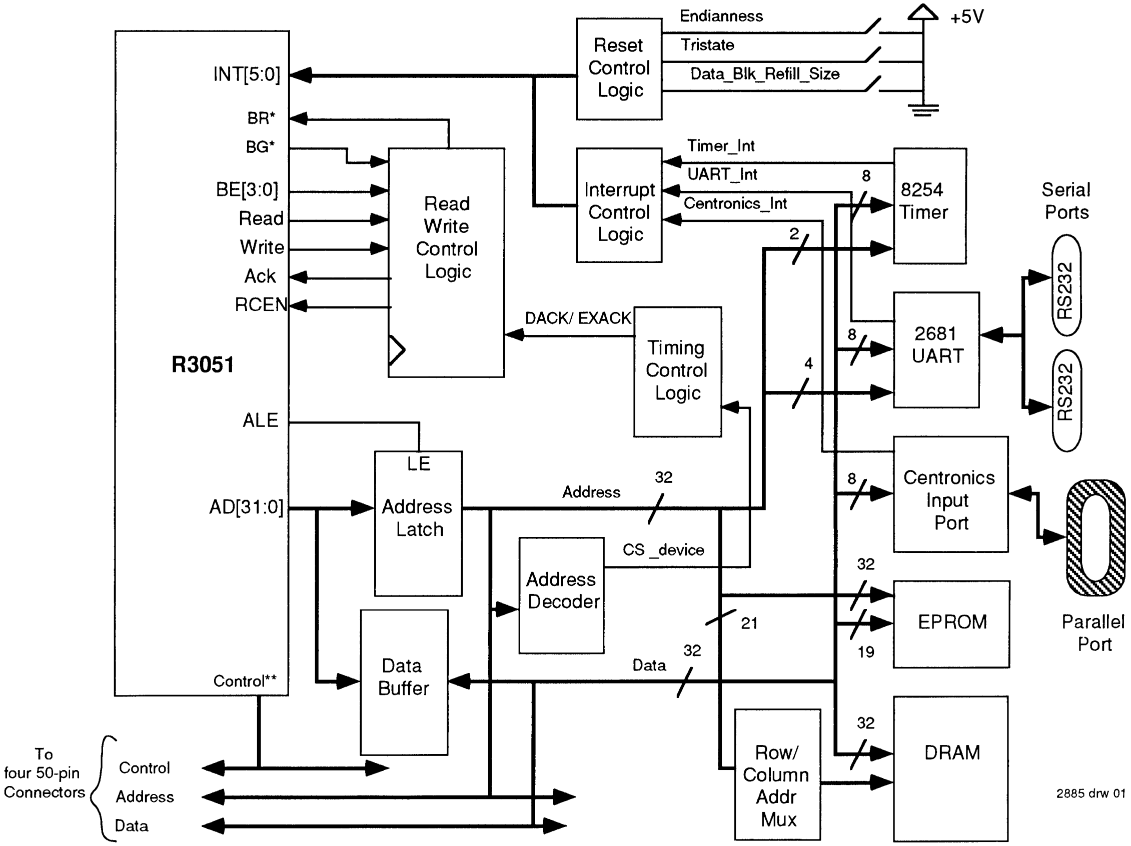
The 3051 Bus, along with other control signals, is connected to a set of pins in the center of the board next to the wire wrap area. These signals can be used to connect additional hardware on either the wire-wrap area or on another board via a ribbon cable. DMA control is provided.

SIGNALS SUPPLIED ON EXPANSION CONNECTOR

Signal Name	I or O	Description
EA00-EA31	I/O	32-bit buffered address bus
ED00-ED31	I/O	32-bit buffered data bus
SYSOUT	O	Buffered SYSCLK Clock from CPU; used to synchronize data transfers
MRES#	O	Copy of the Reset signal to the CPU
MREQ	O	Memory Request output (handshaking signal for data transfers)
EXACK#	I	Acknowledge input (handshaking for data transfers)
IP4-IP5	I	Auxillary input pin to the 2681 UART
WEA-WED	O	Write Enables for the four bytes of the data word
UCS	O	Chip select signal decoded from the high order address bits for external hardware
INT0:INT5	I	Interrupt inputs to the R3052
RD#	O	Memory Read output signal from the 3052
WR#	O	Memory Write output signal from the 3052
BREQ#	I	Bus Request input to the 3052
BUSGNT#	O	Bus Grant output from the 3052

2885 tbl 01

FUNCTIONAL BLOCK DIAGRAM



** These control signals include R3051 and the on-board control logic signals as well.

SPECIFICATIONS

CPU
25 MHz R3052E

Cache Ram
8KB I-cache, 2KB D-cache (in 3052 chip)

Cacheable Address Space
4 GBytes

DMA Support
Bidirectional tri-stateable buffers can be used to write to DRAM from external logic

Block Refill
4 word instruction block size
1 or 4 word data block size programmable via jumper

Endianness (Byte Ordering)
User programmable via jumper

Read/Write Buffers
Both are 4 words deep (inside 3052 chip)

Interrupts
6 User Interrupts, three synchronized with SYSCLK

I/O characteristics
TTL levels from FCT logic devices, PALs and R3052

Power Supply
2 amps (typical) at 5.0 V, 25°C, at rated speed

Environmental Conditions
Ambient temperature 0°C to +50°C.
Relative Humidity 5% to 95%.

Clock Frequency
25 MHz

Interconnection
Five 50-pin connectors, containing Address, Data, and Control signals and R3052 signals.
Five 20-pin plugs for use with HP logic analyzer.
Two RS-232 serial ports on DB-25 connectors.
One parallel Centronics port for input.

User Selectable Options
Endianness, data block refill size
Tri-State mode of 3052

ORDERING INFORMATION

Each unit is shipped with complete schematics and PAL equations. A user's manual includes instructions on downloading code, operating the Software Integration Manager, and providing the correct timing interfaces to additional hardware. Boards are shipped with the 3052E CPU, but any member of the 3051 family can be plugged in.

Evaluation Boards

R3052E Based Evaluation Board 7RS385

EPROM Upgrades

The following part numbers update the evaluation board hardware to the latest version of the monitor.

Evaluation boards 7RS901BGP
Use with 7RS385 only

Auxillary Download Programs

For downloading code from a MIPS machine into an evaluation board. This software includes programs to convert MIPS object code into S-records and to download either ASCII or binary S-records to a remote target. This software is only needed if you are running the MIPS C-compiler and do not have SPP. If you are using IDT/c or you have IDT/sim or MIPS SPP you already have these utilities.

MIPS download utility 7RS950BUU
Supplied on QIC-24 TAR tape

Integrated Device Technology, Inc.

PROTOTYPING PLATFORM For Any IDT RISC CPU Module

7RS400 Series

FEATURES:

- **Provides a complete R3000 based prototyping system.**
- **Fastest way to build an R3000 prototype**
- **Fastest way to validate software on known working hardware**
- **Includes 1 MByte of static RAM main memory, and 256 KBytes of EPROM**
- **IDT's System Integration Manager in EPROM. Supports downloading software, single step, and other debugging needs.**
- **Two RS232C serial ports (68681), parallel port (IDT7134 dual-port RAM), and programmable interval timer (8254).**
- **Address bus, data bus, and all necessary control signals provided on connectors (four 50-pin IDCs) to permit user expansion.**
- **Direct connections to HP 16500 Logic Analyzer**

DESCRIPTION:

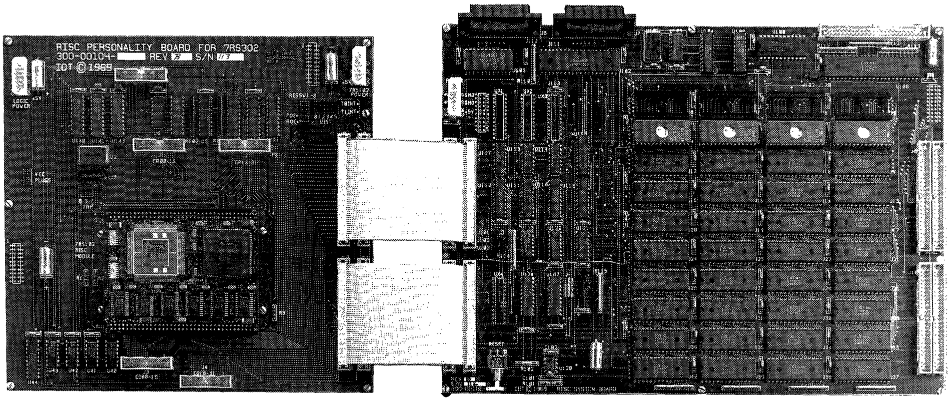
The 7RS400 series Prototyping Platforms consist of an R3000 CPU module and two separate PC boards which connect together by means of convenient ribbon cables to form the total system.

The System Board contains the non-CPU system functional units: static RAM main memory, EPROM monitor, two serial ports, a parallel port, a free-running programmable timer, and expansion connectors to permit additional hardware to be added. It is designed to be placed on a flat tabletop surface. Standoffs are provided for physical support. Access to all components on the board is readily available with this packaging approach. The Personality Board contains control logic and a socket for the RISC module. The Personality Board is also designed for table-top use. The system is available with a choice of several different RISC modules.

The EPROM on the System Board contains IDT's System Integration Manager (IDT/sim), a powerful tool for downloading software and debugging both hardware and software. Drivers can easily be added to support other I/O devices or changes in I/O addresses.

Software can be downloaded into the board from a MIPS machine, from an IDT MacStation development system, or from a PC or SUN workstation running IDT/c.

(This product was previously sold without the module as a 7RS300 series product.)



The 7RS400 series Prototyping Platform.
The System Board is on the right; the personality card and CPU module on the left.

FLEXIBLE PROTOTYPING PLATFORMS

System Description

The 7RS400 series RISC Prototyping Platform is designed to simplify the initial prototyping of both hardware and software for systems using one the the IDT RISC Subsystem CPU Modules. The System Board is very general, and is the same in all of the 400 series Platforms. It contains basic control logic, mostly in PALs, 1 megabyte of static main memory, 256K of EPROM, a counter/timer, and I/O ports. Static RAM is used for main memory to provide the simplest interface to the module. The EPROM contains IDT's System Integration Manager in about 80K; the rest is available for user software.

The System Board connects to a personality board for the module through a pair of ribbon connectors. Each module architecture uses a different personality board. The personality board provides such features as clock generation, R3000 reset and initialization, read and write buffers, etc., to the extent that they are not already on the module. The personality board also contains five 20-pin plugs that can be directly connected to an HP 16500 series Logic Analyzer, and provides a uniform interface to the System Board.

System Board Hardware

The System Board is powered by a single 5 volt supply connected to a plug on the board. The plug conforms to the standard used for PCs, so an ordinary inexpensive PC power supply works easily with the board. A terminal can be connected to one of the RS-232 ports to act as the terminal for the Software Integration Manager. The other serial port is generally used to download software from some host system. Alternatively, there is an 8-bit wide parallel port built using dual port RAM that can be used for higher speed download.

Four 50-pin IDC (3M) connectors are configured for connecting additional hardware to the System Board. They contain the following signals:

- 32 bits of address
- 32 bits of data, and 4 parity bits
- SYSOUT (buffered clock from the R3000)
- RESET# (copy of the R3000's Reset signal)
- Parity and Address output enables from the address and data registers (to permit tri-stating other data onto these lines).
- Six interrupt lines to the R3000. These are registered or not, depending on the module.
- The four byte Write Enable signals.
- Five decoded chip select outputs from the upper 16 bits of address (1FE6 through 1FEE).
- MEMRD#, used to enable output devices in the expansion system during data read cycles.
- Auxillary input and output signals from the 68681 dual UART
- MREQ# and XACK# handshaking signals for controlling the timing of data transfers.

Personality Board Hardware

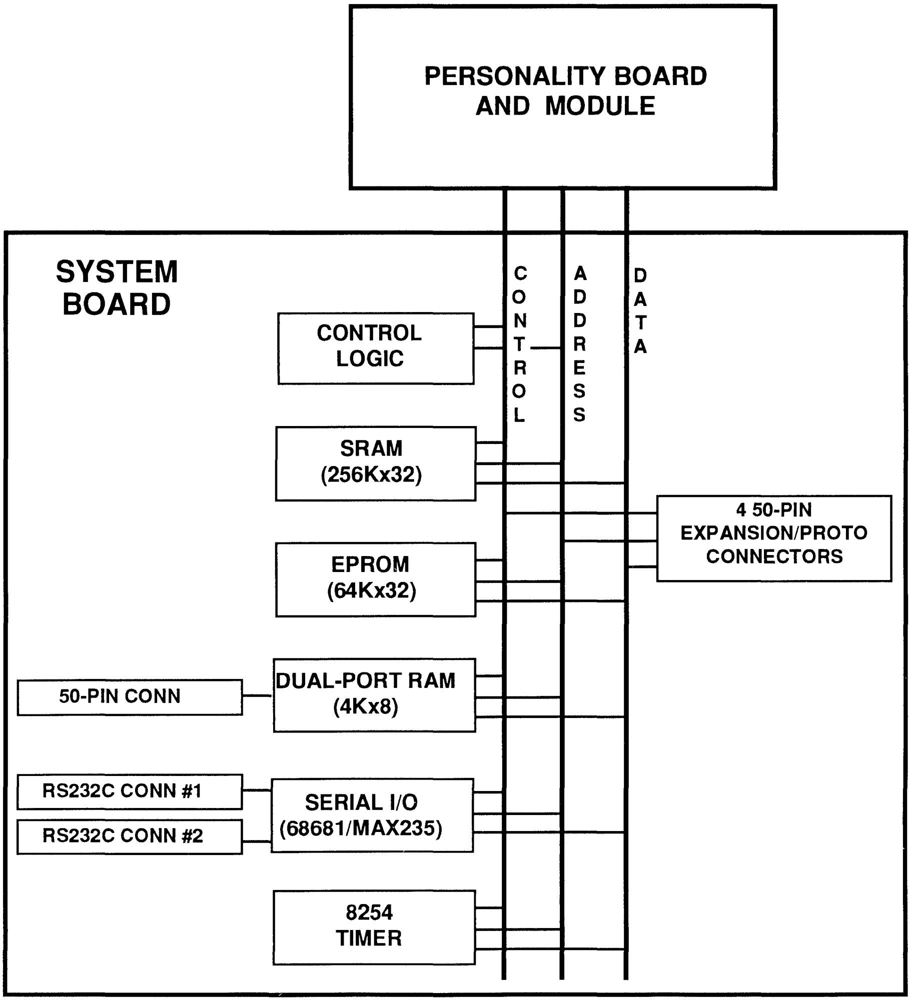
The personality board connects to the system board through two ribbon connectors. It contains a cut out area and plugs which accept the appropriate module. There are two five-volt power connectors, again using standard PC plugs. One power supply is for the personality card, the other for the module.

Five connectors are pre-wired to connect the modules signals to an HP logic analyzer. Because of the speed of the signals in the R3000 system, the connectors are placed on the slow side of the read/write buffers, so for disassembly and trace purposes, the R3000 must be run uncached.

Software Included

The System Board contains IDT's System Integration Manager (IDT/sim) in EPROM.

FUNCTIONAL BLOCK DIAGRAM



Block Diagram of the 7RS400 Series Prototyping Platforms

INCLUDED WITH SYSTEMS

Each Prototyping Platform includes the System Board, completely populated with 1 Mb of RAM and 256K of EPROM, with the Software Integration Manager in the EPROM. Each System also includes the appropriate personality card for the module architecture indicated and configured for the speed indicated. Documentation includes complete schematics for both the system board and the personality board, including all the PAL equations for the control circuitry.

ORDERING INFORMATION

7RS109 Based System

This system is built around a 33 MHz 7RS109 CPU module. The module contains an R3000A CCPU, the R3010 FPU, 64KB each of I- and D- Cache (using 16K x 4 SRAMs), and a four word deep Write Buffer using the R3020 chip. The module contains the crystal and provides buffered copies of the CPU Sys-Out Clock to the prototyping system. The module also contains interrupt synchronization logic.

7RS409	**$7500**

7RS110 Based System

This system is built around a 25 MHz 7RS110 CPU module. The module contains an R3000A CPU, the R3010 FPU, 16KB each of I- and D- Cache, and a single word deep Write Buffer using logic chips. The module contains the crystal oscillator and provides buffered copies of the CPU Sys-Out Clock to the prototyping system. The module also contains interrupt synchronization logic.

7RS410	**$7500**

Auxillary Download Program

For downloading code from a MIPS machine into an evaluation board. This software includes programs to convert MIPS object code into S-records and to download either ASCII or binary S-records to a remote target. This software is only needed with MIPS computers; all other machines (including the MacStation) have standard utilities available to perform this function.

MIPS download utility	**7RS950BUU**

Supplied on DC6150 QIC TAR tape.

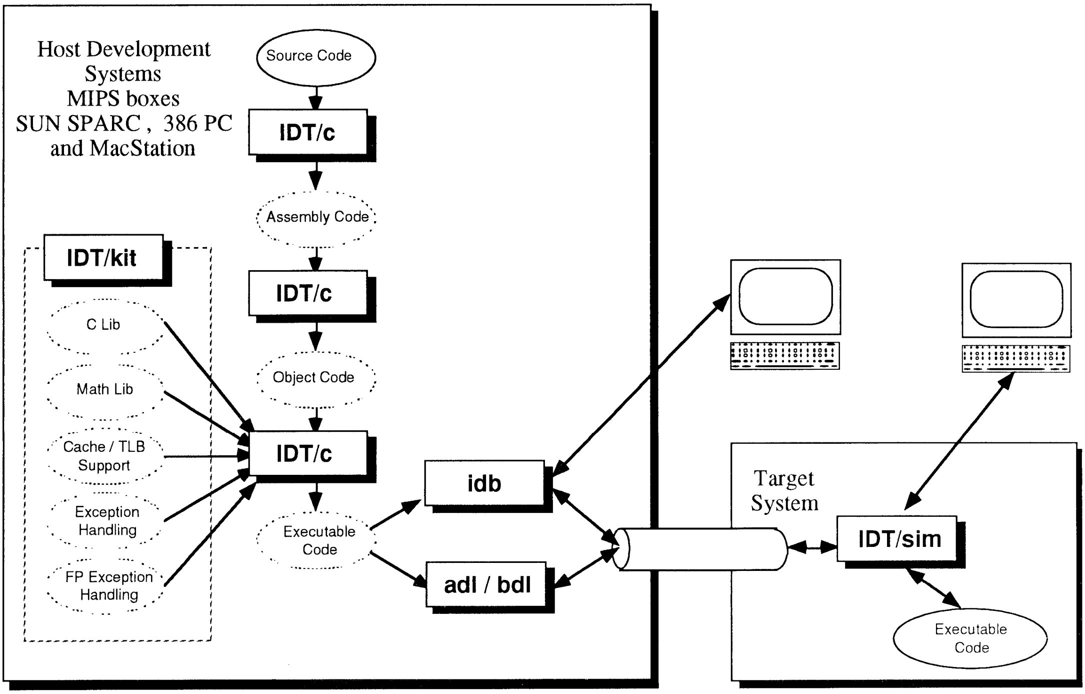
Software Tools

The IDT software development tool chain is illustrated in the drawing below. All the tools are available for use with MIPS workstations, SUN SPARCstations, 386 PCs and IDT's MacStation.

The three packages provide all the software tools needed to compile, link, and debug application code on any of the R3000 ISA (Instruction Set Architecture) CPUs. The idb Symbolic Debug resources are part of the IDT/c Compiler (version 4.1 or higher), and are supported by IDT/sim (version 4.0 or higher). IDT/sim also works with the MIPS symbolic debugger "dbx".

IDT's prototyping systems include IDT/sim in on-board EPROM. For customer-developed prototypes, IDT/sim can be purchased in source code form and can be modified and compiled for the target.

The adl/bdl utilities are ASCII and Binary Download utilities to move S-records from the host machine to the target. These utilities are included with IDT/c and with IDT/sim.



How To Order the Software Tools

1. Refer to the product briefs to determine the correct order code for the host machine and media you need. Order the appropriate item.

2. For IDT/sim and IDT/kit, you must also purchase an Internal Source License. (IDT/c is supplied in binary form and includes a single user license.) The Internal Source License right has a separate order code specified on the product brief. WE CANNOT ACCEPT AN ORDER FOR THE MEDIA UNLESS THE SOURCE LICENSE HAS ALREADY BEEN PURCHASED OR IS PURCHASED SIMULTANEOUSLY.

3. If you wish to distribute code containing portions of IDT/sim or IDT/kit in your end product, including in ROM, then you must purchase one of the two distribution licenses, also specified on the Product Brief. Please refer to the License itself for specific license grants and restrictions. The distribution licenses can be purchased at any time; they do not have to be purchased with the source license.

The complete source license is printed in this catalog. If this license is not acceptable to you, you may negotiate a different license with IDT. The modified license will have to be signed by both parties.

Integrated Device Technology, Inc.

IDT/sim *NEW! Version 4.0* IDT7RS901
SYSTEM INTEGRATION MANAGER
ROMable DEBUGGING KERNEL
FOR R3000 ISA CPUs

FEATURES:

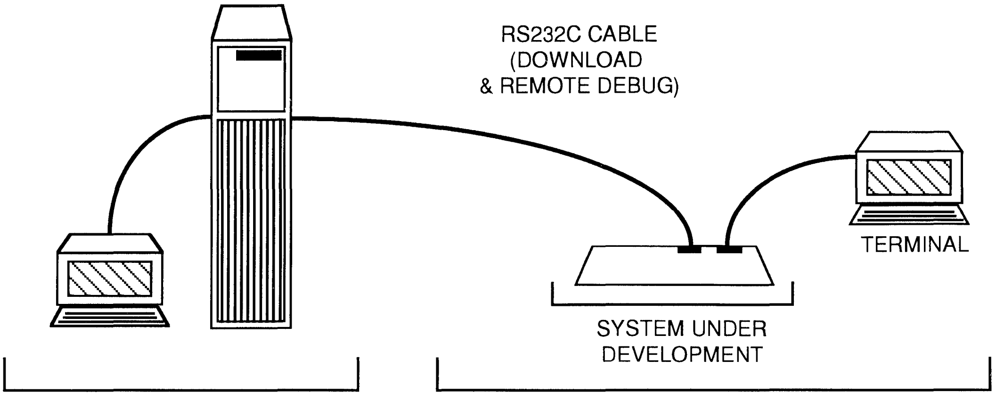
- Complete Source Code Provided
- Robust Debug Monitor
- Supports Source Level Debug
  DBX - MIPS Tool Chain
  IDB - IDT/c Tool Chain
- Remote File Access - Connects Target & Remote Host
- Diagnostic Tests for Memory, Cache, MMU, FPU, and System
- Adaptable to Systems With or Without Hardware Floating Point Accelerator
- Includes Variety of Device Drivers
- Easy to Add New Commands and I/O drivers

POWERFUL TOOL FOR INTEGRATION OF SYSTEMS BASED ON R3000 ISA CPUS :

The IDT7RS901 System Integration Manager (IDT/sim) is a ROMable software product that permits convenient control and debug of RISC systems built around R3000 ISA CPUs (R3000, R3001, R3500, 3051 Family, 3081 Family). Facilities are included to operate the CPU under controlled conditions, examining and altering the contents of memory, manipulating and controlling R3000 resources (such as cache, TLB and coprocessors), loading programs from host machines, and controlling the path of execution of loaded programs.

IDT/sim source code includes IDT's MicroMonitor, a very simple monitor for performing the initial debugging of new hardware.

IDT/sim requires 115KB of EPROM space for code and data, and 71KB of RAM space for uninitialized variable data and stack.



HOST SYSTEM (MacStation™, MIPS, Sun, or PC)

TARGET WITH IDT/sim in EPROMs System Integration Manager

IDT/SIM FEATURES

IDT/sim is a software tool to help system designers debug hardware designs and port software to systems based on one of the R3000 ISA CPUs (R3000, R3001, R3500, 3051 Family, 3081 Family). The software is supplied in EPROMs on most IDT RISC SubSystem Development products, and may be purchased in source code form so it can be compiled and installed on your system.

IDT/sim provides all the basic functions needed to get a new hardware design debugged and to port and debug software on it. Typically, the monitor is compiled and burned into EPROMs that are plugged into the target system. Approximately 115KB of EPROM are needed for the binary code, and 71KB of RAM are needed for storing variables. Once installed, the designer communicates with the monitor via a simple terminal connected to an RS-232 port on the target system. Source code is included to support a variety of UARTs for this port. On start-up, the monitor will determine the cache and main memory sizes automatically.

DIAGNOSTICS

The monitor includes a set of diagnostic routines for testing the integrity of the hardware.

Main Memory Test: Finds opens, shorts, and stuck-at faults on data and address lines. A cache memory test runs memory tests on both caches, checks tag memory, and verifies that instructions can be executed from cache.

System Test: Checks the ability to read and store full words, half words, and bytes. Checks cache operation for valid, hit/miss, and invalidation.

MMU Test: Checks operation of TLB inside the R3000.

Floating Point Test: Tests the functionality of the R3010 FPU, including exception interrupts.

DOWNLOAD SUPPORT

Object code created on a software development system can be downloaded in either ASCII S-records or binary formats to the target system's memory. The code can be produced with the MIPS development tools or with IDT/c on any of a number of development platforms: MIPs, Sun, IDT's MacStation, and 386/486 PCs under Xenix or DOS.

IDT/sim source code includes utilities to convert object code from the MIPS compiler to S-records, to convert the S-records to a binary format (which is more compressed and downloads faster), and to download the binary records to the target. Similar utilities for use with the IDT/c multi-host C compiler are supplied with IDT/c.

A terminal emulation feature allows the terminal, used as the IDT/sim console, to also be used as a terminal to a software development system accessed through a second serial port. This mode supports remote file download.

DEBUG COMMANDS

There are a variety of commands included in IDT/sim to support software/hardware debug.

Execution Control: Breakpoint, call, continue, go, gotill, next, step, unbreak.

Memory Commands: Assemble, cache flush, compare,disassemble, dump, dump cache, dump registers, fill, fill registers, move,read/write cache, search and substitute.

TLB Commands: Dump, flush, map, pid and probe.

Remote Debug: Source level debug with DBX on a MIPS RISC/os system and with IDT Cross development 'c' compiler tools.

Communications: Remote file access, terminal emulator and set baud rate.

RUN-TIME SUPPORT

IDT/sim includes over 40 functions that can be called by user's programs to perform common I/O and R3000 control operations. A complete list of the commands is listed later in this document.

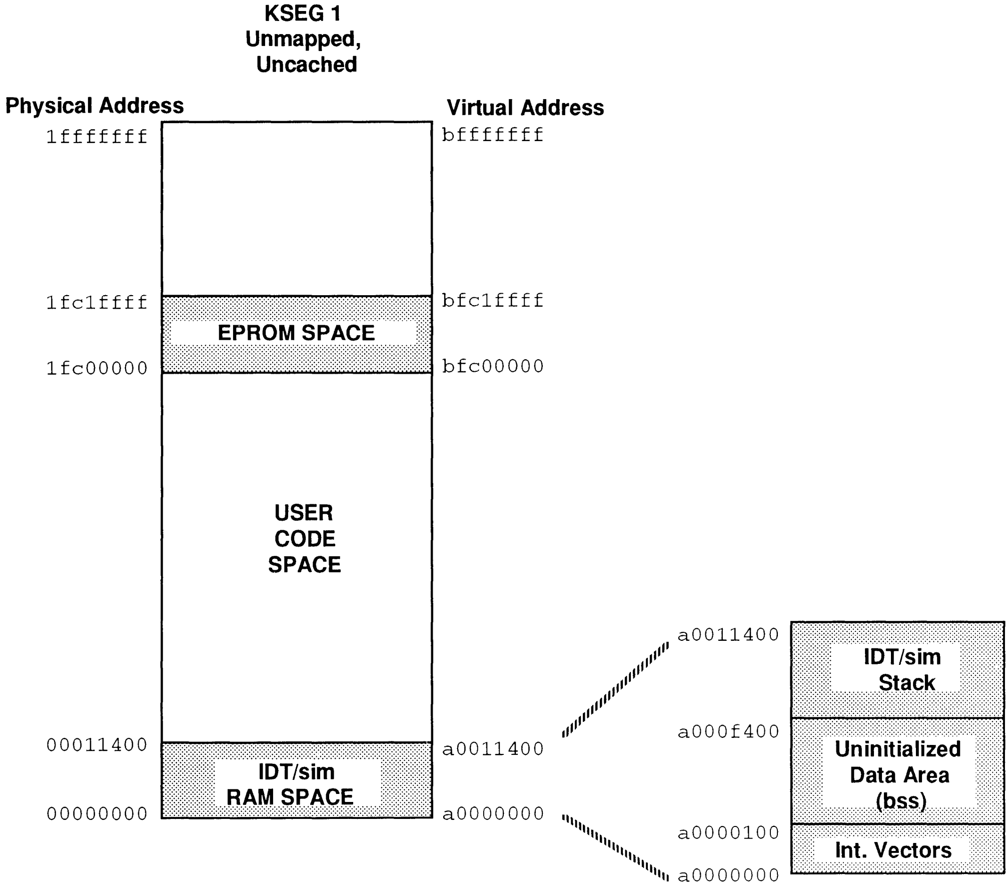
NEW FEATURES IN VERSION 4.0

IDT MicroMonitor: IDT/sim includes IDT's MicroMonitor, a very simple monitor for performing the initial debugging of new hardware. The only hardware which must be functioning to run the MicroMonitor is the CPU, EPROM and a serial port function. This allows for immediate debugging of hardware even when the DRAM memory is $\overline{\text{not}}$ functioning.

Source Level Debug: Supports source level debug using either IDT/c or the MIPS Tool Chain.

Remote File Access: Connects target with remote host file system allowing file transfer between target and host.

Trace Facility: Traces the memory accesses of a user program. Provides for tracing the path of execution reads from and writes to memory. Trace qualifiers allow the tracing of a specific instruction or class of instructions. Also specific memory ranges may be specified. The user may stop tracing on the following conditions: Trace buffer full, hitting a breakpoint, executing a specific instruction or accessing a specific memory range. The trace buffer contents may be displayed using standard R3000 family mnemonics.



IDT/sim Memory Map

The figure above shows the memory utilized by IDT/sim. The EPROM space starts at virtual address bfc00000, which is the R3000's start-up address. The compiled version of IDT/sim with all features included occupies about 115 KB of EPROM space, and is normally placed in 128 KB of EPROM. IDT/sim uses main memory to store interrupt vectors, variables, and a stack. 71 KB of RAM space is reserved for this data.

IDT/SIM COMMANDS

asm <addr> Examine and change memory interactively using standard assembler mnemonics.

brk|b [addresslist] Set/display Breakpoints.

cacheflush/cf [-i|-d] Flush I-cache and /or D-cache.

call|ca <address> [arg1 arg2 ... arg8] Call Subroutine with up to 8 arguments.

checksum|cs [start_addr num_bytes] Display the checksums for an address range.

compare|cp [-w|-b|-h] <RANGE> <destination> Compare the block of memory specified by *RANGE* to the block of memory that starts at *destination.*

cont|c Continues execution of the client process from where it last halted execution.

dbgint|di [<-e|-d DEV>] Debug interrupt enable/disable - allows 'break key' to generate external interrupt.

debug|db [DEV] Enter remote debug mode.

dis <RANGE> Disassemble target memory specified by *RANGE.*

disptag|dt [-i] RANGE Displays the instruction or data cache tag values and data contents.

dr [reg#|name\reg_group] Dump the current contents of register(s).

dt Dump the trace buffer.

dump|d [-w|-h] <RANGE> Dump the memory specified by *RANGE* to the display .

enable DEVICE Connect to remote host for file access

fill|f [-w|-h|-b|-l|-r] <RANGE> [value_list] Fills memory specified by range with *value_list.*

fr [-s|-d] <reg#|name> <value> Fill <reg#name>with<value>.

go|g [-n] <address> Start execution at address <address>.

gotill|gt <address> Continue execution until address <address>.

help|? [commandlist] This command will print out a list of the commands available in the monitor. If a command list is supplied, only the syntax for the commands in the list is displayed.

history|h Display last 16 commands entered.

idb [DEVICE] Connect to remote host source level debugger.init|i Initialize prom monitor (warm reset).

load|l [options] DEV Download code to target.

move|m [-w|-b|-h] <RANGE> <destination> Move the block of memory specified by *RANGE* to the address specified by *destination.*

next|n [count] Step over subroutine calls.

rad [-o|-d|-h] Set the default radix to the requested base.

rc [-i] [-w|-b|-h] <RANGE> Isolate and read from cache.

rdfile <filename> <RANGE> Read file from remote host file system.

regsel|rs [-c|-h] Select either the compiler names or the hardware names for registers.

search|sr [-w|-b|-h]<RANGE> <value> [mask] Search area of memory for value.

seg [-0|-1|-2|-u] Set the default segment to the requested k-segment.

setbaud|sb DEV Set the baud rate on a serial channel.

step|s [count] Single step count times.

sub [-w|-h|-b|-l|-r] <address> Examine and change memory interactively.

t {-a/-o/-e/-d/-r RANGE/-w RANGE/-c RANGE/-i INS/-m MSK} Trace command.

tc [-e BPNUM] [-d BPNUM] Trace conditionally command.

te [DEV] Connects the console port straight through to a second serial port.

tex [RANGE] Exclude tracing calls to RANGE.

tlbdump|td [RANGE] Dumps the contents of the TLB.

tlbflush|tf [RANGE] Invalidates the contents of the TLB.

tlbmap|tm [-i index] [-ndgv] <vaddress> <paddress> Virtual to physical mapping of the TLB.

tlbpid|ti [pid] Set/display TLB PID.

tlbptov|tp <physaddr> Probe the TLB.

ts [-b/-f/-o/-r RANGE/-w RANGE/-i INS/-m MSK] Stop trace command.

unbrk|ub <bpnumlist> Clear breakpoints.

wc [-i] [-w|-b|-h] <RANGE> [value_list] Isolate and write to I or D cache.

wtfile <filename> [value_list] Write file to remote host file system.

LIST OF RUN TIME SUPPORT ENTRY POINTS

_exit
atob
clear_cache
cli
close
exc_utlb_code
flush_cache
get_mem_conf
get_range
getchar
gets
install_command
install_immediate_int
install_new_dev
install_normal_int
ioctl
longjmp
open
printf
putchar
puts
rclose*
read*
reinit
reset
restart
rfileinit*
rgets*
rlseek*
ropen*
rprintf*
rread*
rwrite*
set_mem-conf
setjmp
showcar
sprintf*
strcat
strcmp
strcpy
strlen
tokenize
write

* New in Version 4.0

DEVICE DRIVERS (INCLUDED IN SOURCE CODE)

68681/2681 DUART

8530 SCC

SCSI

Centronics Parallel

8254 Timer/Counter

8251 UART

ORDERING INFORMATION

To upgrade an IDT board level product, see the EPROM order codes below. To order IDT/sim in source code, order the Internal Use License AND order the software on the appropriate source media. You may also order binary distribution rights for the run-time version of the monitor. Ask your IDT sales office for information.

Licenses

Internal Source License **7RS901SLV**

Permits purchase of up to six copies of source code (any media combination) and use of source code to develop run-time binaries on up to six machines at a time, but does not permit inclusion of the run time code in an end product. Also purchase one or more of the Source Media listed below.

Limited Binary Distribution Rights **7RS901BLP-L**

Extension to Internal Source License to permit inclusion of binary code into end product. Internal Source License must be referenced on order or ordered simultaneously. This license permits up to 100 copies to be distributed royalty-free. For additional copies purchase the Unlimited Binary Distribution Rights .

UnLimited Binary Distribution Rights **7RS901BLU-L**

Extension to Limited Binary Distribution Rights to allow unlimited distribution of binary code. Internal Source License and Limited Binary Distribution Rights must be referenced on order or ordered simultaneously.

Maintenance Agreement **7RS901SSY**

One year free updates. We supply a direct telephone contact for support.

Source Media

IDT/sim source code can be compiled with either the MIPS C compiler or with IDT/c version 4.1 or later. Earlier versions of IDT/c cannot compile this code. The products listed below are media only and must be purchased with license 7RS901SLV above.

Source for 386/486 PC, MS-DOS **7RS901SBF-L**

Compile with IDT/c C-Compiler. Shipped with both 1.2 MB 5.25" and 1.44 MB 3.5" diskettes.

Source for 386/486 PC, SCO Xenix **7RS901SXX-L**

Use with IDT/c C-Compiler. Shipped with both 1.2 MB 5.25" and 1.44 MB 3.5" diskettes.

Source for IDT MacStation, on Mac Disc **7RS901SMD-L**

Use with MIPS C Compiler supplied with MacStation or with IDT/c.

Source for MIPS / SUN machines, on DC6150 QIC TAR Tape **7RS901SUU-L**

Use with MIPS C Compiler or with IDT/c.

EPROM Versions

The following versions of IDT/sim are supplied in EPROMs for the indicated hardware. These versions are for updating the hardware to the latest version of the monitor. No license required. An upgrade is NOT available for the 7RS382 and 7RS383 Evaluation Boards.

For Any 7RS30x, 7RS40x Prototyping System	**7RS901BAP**
For 7RS388 Real8™ Laser Printer Controller	**7RS901BFP**
For 7RS385 Evaluation Board	**7RS901BGP**

Integrated Device Technology, Inc.

IDT7RS903

IDT/c
Multi-Host C-Compiler System
New! Version 4.1

FEATURES:

- **ANSI C-compiler, Optimizing Scheduler, Assembler, Linker, Librarian, and ANSI Libraries**
- **Efficient Floating Point Emulation Mode for systems without hardware FPU. Includes Transcendentals**
- **Symbolic and assembly level debug support.**
- **Versions available for 80386 machines under MS-DOS™, MIPS machines and MacStation under RISC/os, and Sun SparcStation**
- **Provides control over multiple memory segments**
- **Supports entire IDT family of MIPS ISA Processors (R3000, R3001, R3051/2, 3081/2, and R3500)**

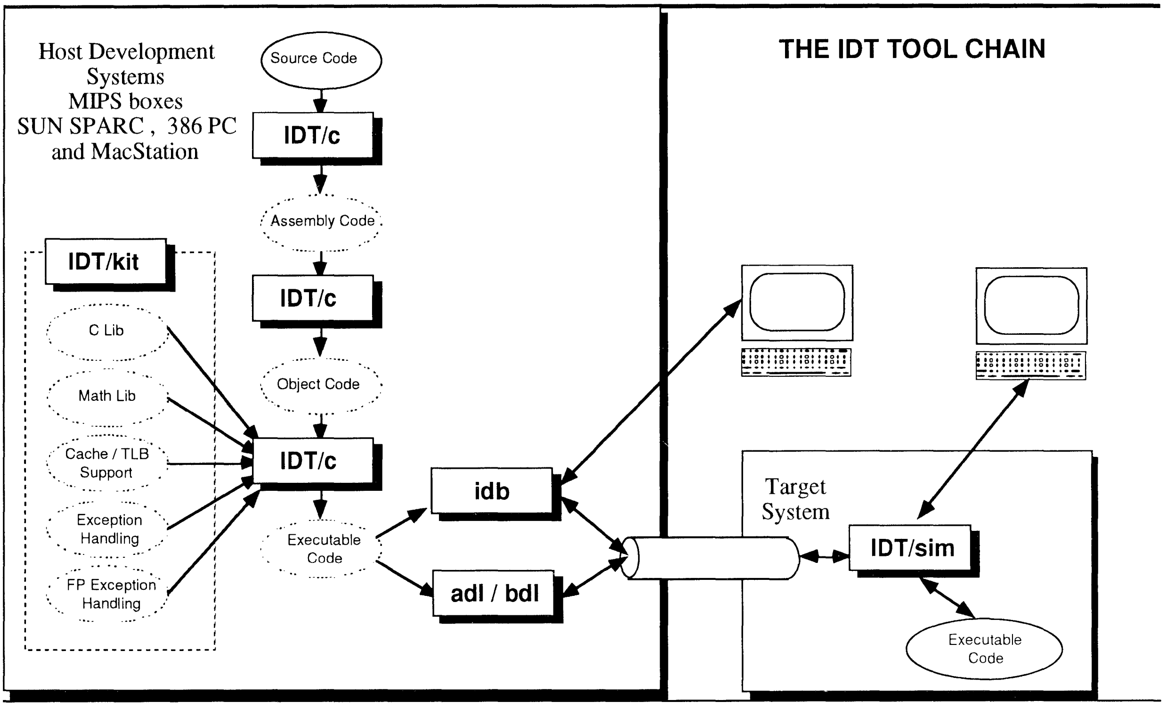
OPTIMIZING C-COMPILER SYSTEM:

IDT/c is C compiler system for the MIPS R3000 and derivatives, specifically designed for deveioping and debugging code that runs on a remote target. The compiler system includes the GNU C compiler, an assembler with instruction optimization, a linker, a librarian, and a symbolic debugger. A complete floating point emulation library is included also.

The IDT/c package is available for execution on 386 machines under MS-DOS, on the MIPS and SUN workstations, and on IDT's MacStation.

Unique features of IDT/c include the ability to divide code into segments for programming ROMs, the ability to relocate initialized variables into ROM space, extremely efficient floating point emulation (up to ten times faster than other emulation methods), and a programming language in the symbolic debugger to control execution by testing data and addresses in the program.

New features in release 4.1 of IDT/c include the symbolic debug facility, a switch to produce assembly language output for the MIPS tool chain, and a full set of transcendental arithmetic functions.



FEBRUARY, 1992

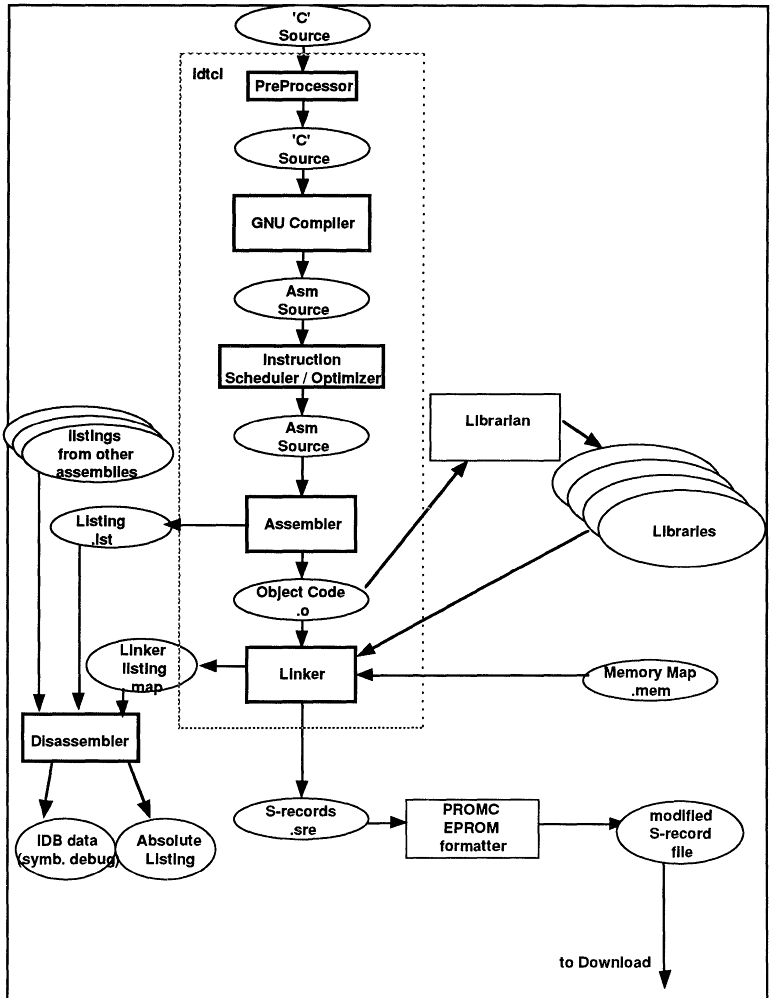
OVERVIEW

The IDT/c C-Compiler System is a complete development package for CPUs based on the R3000 architecture. It contains an optimizing cross compiler, optimizing scheduler, assembler, linker, and a downloader. The 'C' compiler is compliant with ANSI 'C' standard and performs the optimizations available in state-o-the-art 'C' compilers. The assembler supports the R3000 machine instructions and architecture described in the book by Gerry Kane, "MIPS RISC Architecture", including both native and synthetic instructions. The complete IDT/c package runs on a variety of host machines and operating systems and is compatible with other IDT development software, such as IDT/sim and IDT/kit.

Compiler

The C pre-processor is GNU cpp and the compiler itself is based on GNU C. All C-preprocessing features are supported. The combination of the compiler and assembler included in IDT/c has been tested for compliance to the ANSI C standard using the Plum Hall test suite. The C compiler performs extensive optimization in multiple passes through the code. Switches can be used to fine tune the optimizations.

The output of the compiler is an assembly language file. A switch in the compiler selects whether output is for the IDT assembler or for the MIPS assembler. If the MIPS assembler output is chosen, then modules compiled by IDT/c can be assembled and linked in the MIPS environment with modules compiled by the MIPS compiler. IDT/c is far more efficient than MIPS c for floating point emulation. The MIPS compatibility switch makes it possible to use IDT/c only for modules with floating point code, and MIPS c for everything else.



IDT/c Flow

Optimizing Scheduler and Assembler

The IDT assembler implements the R3000 native instruction set as well as the augmented synthetic instructions defined in the "MIPS RISC ARCHITECTURE" book by Gerry Kane. An optimizing scheduler first expands the synthetic instructions into the native instruction set. It then rearranges code to take advantage of R3000 pipeline architecture. The scheduler also analyzes loads of static constants and tries to make use of previously loaded constants. The assembler produces .o files which can be linked together with other files to produce an executable file.

Memory description file

The memory description file is used to instruct the linker where to place object modules in the R3000 memory map. It tells the linker what address classes are legal, what addresses exist within those classes, and what addresses should be written to output files. The file consists of a sequence of class specifications (CODE, DATA, etc.) and associated address ranges. It is possible to control placement of individual modules.

Linker

The linker combines separately assembled program files into one object module. Command line switches may be used to override the memory description file.

There are three types of output file formats supported: S-Records, Intel hex, and binary image. The S-Record files are useful in down-loading to target boards. The hex format file is useful for EPROM programming because the code can be divided into multiple files under this format. S-Records can be downloaded to a target containing the IDT/sim monitor using a supplied download utility (DOS) or uucp (UNIX). Additionally, the IDB facilities in IDT/sim v 4.0 and IDT/c v 4.1 provide fast, reliable download of S-Records.

Endianess

IDT/c includes a switch so that code may be compiled in either Big-endian or Little-endian format.

Floating Point Library

IDT/c includes a floating point emulation library. A switch in the compiler is set at compile time to determine how the compiler should handle floating point instructions. In the normal mode, it will produce R3010 Floating Point Accelerator instructions in the object code. If the switch is set the other way, the compiler will insert calls to the floating point library instead, and the floating point library must be available at link time. Because the compiler knows about the library during compile time, it can perform optimizations not otherwise possible and minimize the execution penalty for using software instead of hardware.

Librarian

IDT/c supports object code library files. Many compiled routines may stored in a single library file by using the Librarian utility. At link time, the linker extracts only the routines actually used. This technique reduces the number of files that must be dealt with explicitly during program development.

PROM-C

PROM-C is a utility included with IDT/c that permits variables initialized by the program to be moved from their normal locations in the object code into designated memory space destined for EPROM. The user program can execute a simple routine at start-up to move the variables from EPROM back into RAM space at the appropriate locations.

TYPICAL COMPILATION SEQUENCE

Assume 4 C modules, m0.c m1.c m2.c m3.c . m0.c contains 'main' function.

Compile the programs:

```
    idtcl -O -c -ZA m0.c m1.c m2.c m3.c
```

(the default compiler mode determined at installation time is used)

Make a library:

```
    ilib -c lib.a m1.o m2.o m3.o
```

Link it all :

```
    idtcl -o prog -ZT80020000 -ZD80030000 m0.o lib.a
```

IDB - SYMBOLIC REMOTE DEBUGGER

Remote debugging differs from the 'conventional' in several ways. The control of target program relies on communication line and debugging agent on the target instead of using o/s signals and related services.

Idb was specifically designed to work with IDT/sim to provide full control of the target program. The distinguishing features are: a program mode in which idb executes scripts that contain debugging and flow control commands; and host file services which provide target program with full access to host files. The required physical link between host and target is a single RS232 line capable of 19200 bps.

It is also possible to use direct low-level IDT/sim debugging from an idb session.

Program mode

The script sample on page 6 (last page) illustrates some of capabilities of the program mode.

Remote file services

IDB supports file open, close, read, write, seek; printf; and gets commands in the standard C library format.

FLOATING POINT EMULATION MODE

When floating point and double length variables are used in C programs, compilers usually produce assembly instructions that directly operate on floating point arguments. Most of the time this also requires use of designated register set to hold floating point operands. The C compiler is aware of underlying hardware and attempts to produce optimal code by using all available resources.

The R3000 architecture has the floating point coprocessor in separate chip (R3010). There are numerous floating point instructions that operate on the 32 floating point registers inside the R3010. When floating point hardware is not present — for example, an R3000 without the R3010, or an R3051 or R3052 controller chip — executing programs that use floating point arithmetic requires software that can compensate for the missing hardware. There are two basic solutions to this problem: trapping on the floating point instructions at execution time, and simulating the FPU in software in the trap handler; and modifying the compiler so that it does not produce any instructions for the FPU in its output.

Using the operating environment to trap all attempts to execute instructions on (non-existent) floating point hardware offers the advantage of using a single object code version of the application whether hardware FPU is present or not. The disadvantage is that the complete FPU state machine must be emulated to the last detail since the code produced by the C compiler expects the real hardware to be present. The code that must be executed for each FP instruction is substantial: the trapping overhead for each FP instruction, the maintenance of the FP state machine, and the instructions to execute the required FP operation on integer hardware.

For example, the sequence below requires 3 traps, each of which involves saving all the registers used in the particular trap routine, maintaining the state of the 'virtual' FP register set somewhere in memory, performing the actual FP arithmetic (double addition), and updating the 'virtual' FPU status register bits.

```
    lwc1        $f14, ($9)
    lwc1        $f15, 4($9)
    add.d       $f8, $f14, $f6
```

The solution implemented in IDT/c is to switch the compiler to a different mode for the two environments. In emulation mode, the compiler does not assume presence of any additional hardware, so only R3000 instructions are produced, and FP operations are performed by calls to special routines. The calls are compiler-generated and there is absolutely no difference on the C source level. The same C program that generated the example above would generate the following code in IDT/c emulation mode.

```
    lw          $4, ($9)
    lw          $5, 4($9)
    jal         __adddf3
```

The only overhead is that of performing FP operations on R3000. On floating point intensive applications, IDT/c typically yields execution times four to five times slower than R3010 execution times, but eight to twelve time faster than the FP hardware emulation method described above.

IDT/C PERFORMANCE

The performance of IDT/c has been measured against the MIPS C compiler, which is a well-respected tool chain offering the highest performance on RISC machines.

Floating Point Emulation

Table 1 shows the relative performance of a floating point intensive program under three different floating point options. The hardware in all three cases was the the 7RS388 Laser Printer controller board. This board uses the R3000, the R3010, and 16KB each of Instruction and Data cache. The board was operating at 25 MHz.

Bnchmrk	Hardware	Trap Emulation	IDT/c Emulation
add.s	5	1345	25
sub.s	5	1340	25
mul.s	5	1320	25
div.s	10	1830	55
sin.s	55	23490	430
cos.s	55	23650	440
ln.s	40	19045	580
sqrt.s	60	10480	235
add.d	5	2290	35
sub.d	5	2310	40
mul.d	10	2315	50
div.d	10	2485	110
sin.d	55	23295	390
cos.d	55	23535	350
ln.d	45	18930	560
sqrt.d	70	10380	220

Table 1. IDT/c FP Emulation Performance

- The column labeled "Hardware" shows the results of a MIPS C compilation using hardware FPA instructions.
- The column labeled "Trap Emulation" shows the same binary run in the same system with no hardware FPA. The operating system traps on the R3010 instructions, and the trap handler emulates the R3010 hardware in software.
- The last column shows the results for IDT/c Emulation mode. In this test, the code was re-compiled with IDT/c in FP Emulation Mode. Whenever a floating point operation is required, the compiler generates a call to the appropriate

library routine to perform the function. There is no trap overhead.

Integer Comparisons

Table 2 illustrates the results of a set of benchmarks compiled by both the IDT/c and MIPS "C" compilers. These benchmarks are commonly referred to as "The Intel Benchmark Suite", since Intel introduced them to measure the performance of various embedded processors began when they announced the i960CA.

Benchmark	MIPS C	IDT/c
Anneal	5200	5340
BubbleSort	448	542
Dhrystone	38,461	35,714
MatMult	1920	2710
PI-500	1140	1540
QuickSort	392	477

Table 2. IDT/c vs. MIPS C on Intel Benchmarks

Table 3 may be more representative of the range of differences, as the Stanford Benchmark suite tends to exercise more of the processor.

Benchmark	MIPS C	IDT/c
Perm	.059	.067
Towers	.061	.066
Queens	.039	.043
IntMatMult	.083	.089
Puzzle	.311	.396
QuickSort	.040	.047
BubbleSort	.044	.054

Table 3. IDT/c vs. MIPS C on Stanford Benchmarks

The results indicate a variety of performance differences between IDT/c and MIPS "C" across these benchmarks. Note, however, that these benchmarks may not be fully representative of either compiler, as they are extremely small programs using only integer arithmetic.

Note that the performance difference between these compilers is different across different hardware platforms. Specifically, the ability of the benchmark to remain cache resident will influence the performance gain of MIPS techniques such as procedure inlining and loop unrolling. Systems with differing cache sizes and/or memory latency may then show different results for integer code.

Mix and Match Strategy

To maximize performance, a system designer could choose to use a "mix and match" strategy in the software toolchain. For example, the bulk of the application could be compiled using the MIPS compiler, while IDT/c is utilized in the floating point intensive portions of the code.

This approach marries the best of both toolchains. The MIPS compiler extracts maximum performance from the majority of the integer only code, while IDT/c does the best job of performing floating point operations in software.

IDT/c facilitates this approach by allowing IDT/c to use the MIPS backend assembler, thus allowing code generated by IDT/c to be directly linked with code generated by the MIPS compiler. Thus, the programmer can use IDT/c (with the MIPS backend assembler) on the floating point intensive code, and the MIPS compiler on the rest of the code.

Table 4 illustrates the performance gain achievable when using such a mix and match strategy. In this table, two of the Stanford Benchmarks are shown with an IDT/c only, and with a mix and match strategy.

Benchmark	IDT/c	Mix and Match
Mn	.277	.267
FFT	.327	.303

Table 4. IDT/c with IDT/c back-end vs. MIPS back-end

```
1>bp foo1
2>bp foo2
10>c                                continue
20>if $cb .eq. 1 then 50            if stopped in foo1 just
                                      record the stack pointer
                                      value
30>if $cb .eq. 2 then 60            if in foo2 goto 60
40>stop                             stopped somewhere else (not
                                      in foo1 or foo2)
50>$stack = $29
55>goto 10
60>if $29 .gt. $stack then 10       continue
70>ub 1                             stack is too low. remove
75>ub 2                             bp 1 and bp 2.
80>bp int_handler                   set breakpoint in
                                      int_handler
90>c                                continue
100>if $cb .ne. 1 then 40           not in int_handler
110>if $29 .gt. 0xa0060000 then 90
                                    not dangerously low
120>if $4 .gt. 10 then 200          int_handler should never be
                                      called with the first arg
                                      greater then 10
130>stop
200>ar                              display argument
210>stop
```

Sample Program in IDB Control Language

This example monitors stack depth difference in two routines and if positive starts to trace stack values in the third routine.

ORDERING INFORMATION

The IDT/c C-Compiler is an efficient R3000 C-compiler system based on the popular GNU C and hosted on a variety of computers. The IDT/c system includes the compiler, assembler, scheduler and linker. All PC versions of the software are shipped with both 1.2 MB floppy discs and 1.44MB 3.5" diskettes. A "boxtop" single user license is included with the product. Contact your IDT sales office for multiple user licensing.

Media, with Floating Point Library

The software listed below includes the floating point library.

For 386 machine, MS-DOS	7RS903FBBF-N
This product uses extended memory space on the 386. 4 MB recommended.	
For 386 machine, SCO Xenix	7RS903FBXX-N
For MIPS machine RISC/os, on DC6150 QIC TAR Tape	7RS903FBUU-N
For MacStation, on Macintosh Disc	7RS903FBMD-N
Runs on MacStation R3000 board under IDT/ux.	
For SUN Sparcstation, on DC6150 QIC TAR tape	7RS903FBWU-N

Integrated Device Technology, Inc.

IDT/kit KERNEL INTEGRATION TOOLKIT

IDT7RS909

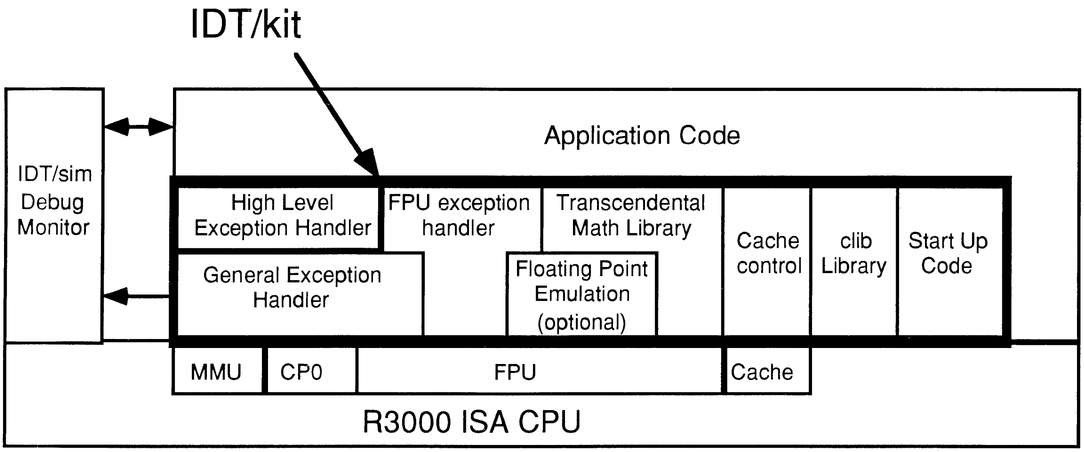
FEATURES:

- **Source code and object code versions of commonly used routines for an R3000 ISA CPU.**
- **Start-Up Code to initialize CPU, MMU, and C runtime environment**
- **Cache control code to size, initialize, flush, and clear for DMA**
- **Re-entrant Exception Handler**
- **Floating Point Emulation Library and Transcendental Math Functions**
- **ANSI Standard C Library**
- **Time Support Functions**
- **MicroMonitor for initial hardware debug**
- **Interface Library to IDT/sim monitor**

ESSENTIAL CODE FOR R30XX SYSTEMS

IDT/kit (Kernal Integration Toolkit) consists of libraries and routines for important system software operations for R3000-based CPUs. Modules are provided for initializing systems, handling interrupts, servicing floating point exceptions, and many other other common operations. Libraries are included for floating point emulation, transcendental arithmetic routines, and ANSI standard C functions. All IDT/kit libraries are supplied in source code (C and assembly) and in object modules compiled for both little- and big-endian systems and for both hardware and software emulation floating point.

IDT's MicroMonitor is also included in the IDT/kit package. The MicroMonitor is a very simple monitor for initial debug of new hardware. It requires only that the CPU, EPROM, and a serial port be operational. The MicroMonitor can be an invaluable aid for detecting state machine problems in first article hardware.



Schematic Representation of the modules in IDT/kit, showing how they control parts of the R3000 CPU and connect to IDT/sim, IDT's debug monitor.

IDT/kit FEATURES

The IDT Kernel Integration Toolkit (IDT/kit) consists of a set of libraries ready to be linked with user developed code. IDT/kit contains functions that would normally be furnished by an operating system like UNIX but without the overhead. Functions are provided for initializing the system, memory management, exception handling and time support; an ANSI standard 'C' library and a math library with transcendental functions are supplied.

This environment can be compiled with IDT/c or MIPS compilers, Big or Little Endian, Cached or Uncached and with an optional Emulation Mode if no Floating Point Accelerator is installed. With IDT/kit, floating point emulation support is transparent to the user application.

IDT/kit typically would co-exist with IDT/sim and become part of a total development and debug environment. On a system where IDT/sim is installed, all the commands, entry points and debugging facilities of IDT/sim are available to the Kernel Integration Toolkit. When using IDT/sim, IDT/kit filters exceptions first. If IDT/kit does not handle an exception, then it is passed to IDT/sim.

Default exception handlers intercept exceptions, save the environment, preserve the Exception Registers for later analysis, restore the environment and return to continue program execution. The default handlers can easily be replaced or extended with more robust handlers written by the user.

IDT/kit relieves the application from the low level tweaking necessary to get started but leaves easy hooks into the system for expansion and polishing as the development progresses. All code is supplied in source code (C and assembly), to allow easy access for modifications needed to tailor the system to specific needs. This allows the programmer to shorten the project development time by the 2 to 3 months required to understand and service the R3000 ISA resources like cache, MMU and exception handling.

IDT/kit includes IDT's MicroMonitor, a very simple monitor for performing the initial debugging of new hardware. The only hardware which must be functioning to run the MicroMonitor is the CPU, EPROM and a serial port function. This allows for immediate debugging of hardware even when the DRAM memory is not functioning.

IDT/kit COMPONENTS

1. MICROMONITOR:

A small assembly language monitor to aid in debugging the hardware design. The MicroMonitor requires only that three hardware resources are functional: the CPU can execute instructions, the EPROM can provide instructions and the UART can send and receive characters.

2. IDT_CSU.S, THE START UP MODULE

Supplies the initialization and set up code necessary for operation of the system.

- initialize the Status Register
  - a) clear parity error bit
  - b) set Coprocessor 1 usable bit correctly
  - c) clear all IntMasks enabled
  - d) set kernel/user mode
- set Cause Register
  - a) clear software Interrupts Pending
- clear bss area
- establish temporary uncached user stack
- determine memory and cache sizes
- establish permanent stack at Top of Memory
- flush I and D-Caches
- if there is a Translation Lookaside Buffer invalidate it
- initialize library if IDT's standard C Library is used
- initialize exception handlers
- jmp to users' main()

3. LIBIKIL.A, KERNEL INTEGRATION LIBRARY:

The IDT Kernel Integration Library (libikil.a) is a library which can be linked to user programs to supply functions required to support the environment of the R3000 ISA family. This library is divided into four sections: Memory Handling, General Exception Handling, Floating Point Exception Handling and Time Support Functions.

a. memory handling - the full range of functions necessary to manage main memory, cached memory and the Translate Lookaside Buffers

b. general exception handling - the functions used in enabling and handling any interrupt exceptions, hardware or software, asserted by the CPU

c. floating point exception handling - provides the support for the Floating Point Unit, the R3010, or for Emulation Software for floating point arithmetic , "strongly recommended" by the IEEE Standard 754-1985. It is transparent to the application code calling this interface whether hardware or software emulation is being employed.

d. Time support functions using the 8254 timer as a prototype

4. LIBILNK.A, THE IDT/SIM INTERFACE LIBRARY:
the IDT/sim linking module which interfaces with all the functions available with IDT/sim not defined with IDT/kit or by the user

5. LIBIC.A, ANSI STANDARD C LIBRARY:
The IDT Standard C Library (**libic.a**) is a standard archive library which, when linked with the users' filename.o files, provides the functions defined by the ANSI standard including standard I/O, String and Character functions, Utility functions, and memory allocation functions.

6. LIBIMATH.A, MATHEMATICS LIBRARY:
The IDT Math Library (**libimath.a**) is a standard archive library which, when linked with the users' filename.o files, provides the transcendental functions required for standard math processing. Whether hardware floating point or software emulation is be used is transparent to the application code calling this library.

7. UTILITIES:
Three utilities that execute under MIPS RISC/os are supplied to convert from compiler output from the MIPS coff file format to an S-record format and to downloading the S-records to a target board in either ASCII or binary forms. These utilities are need only for users of the MIPS C-compiler who do not have IDT/sim or MIPS SPP/e.

HOW IDT/kit IS USED

IDT's Kernel Integration Toolkit includes a robust set of tools for the embedded controller developer. They are "packaged" in accessible, modular containers, the IDT/kit libraries. The four libraries, arranged by function, supply most of the routines required by RISC applications. Only those libraries needed to resolve function calls must be entered on the link command line; the others are never accessed which establishes a fully modular environment.

IDT/kit serves as an envelope for the installation's application code. The source module, idt_csu.S, is linked first, then the development code and, finally, any kit libraries required to support function calls. This allows the developer to concentrate on the application and not waste resources re-developing the support routines. Although the libraries are provided complete and ready to link, source code for all the functions is also distributed to allow easy examination for information or as a template for additional routines. All the necessary Make/Batch files are included to facilitate any changes, additions or corrections. Some examples:

strcpy (or any of the C library routines) - perhaps your installation has developed a super-algorithm. It isn't quite ANSI Standard but does your job better. Simply edit the source in the clib subdirectory (or replace the one that is there), execute the makefile for your configuration (IDT/c or MIPS, Big or Little Endian, REAL or Emulation Mode) and the library, libic.a, now contains your code. Your module could also be placed on the system Link line before the corresponding library and the call would be resolved before the library is searched.

Interrupt Handling - you want the default interrupt handler (it's already there), but you need an additional flag set. Edit the routine in the killib subdirectory, run the Make/batch file for your configuration (IDT/c or MIPS, Big or Little Endian, REAL or Emulation Mode) and the library, libikil.a, now contains "your "default interrupt handler.

These kinds of simple modifications allow the IDT/kit routines to be tailored for a specific system without expending the time to investigate and understand all the routines; only the section applicable to the application need be tweaked and the Make/Batch files provide easy guidelines for doing it.

IDT/kit FUNCTION LIST

CSU_IDT.S: START UP MODULE

start()	Startup routine

LIBIKIL.A: KERNEL INTEGRATION LIBRARY

Programmable interval timer driver

install_timer_driver()	install timer driver
i8254init()	timer driver init
i8254open()	opens the device
i8254ioctl()	i/o control function

Assembly level exception handling

disable_int()	clear selected interrupts
enable_int()	set selected interrupts
exc_norm_code()	general exception code
exc_utlb_code()	UTLB Miss code
exception()	general exception code
init_erc_vecs()	init vector code
longjmp()	go to setjmp() point
other_excp()	handles other exception
setjmp()	set setjmp() state

High level exception handling

add_ext_int_func()	set default exc handler
clr_except_ptr()	clears setjmp pointer
config_memory()	size of main memory
extern_int()	external interrupt code
exception()	general exception code
get_except_ptr()	get execption pointer
init_tlb()	initializes TLB
mem_exc_hdlr()	memory exception code
sae_errmsg()	prints msg & exits
set_except_ptr()	sets setjmp pointer
spurious_int()	unexpected ext interrupt

FPU interface module

fp_defaultHdlr()	default handler
fp_disableTrap()	clears trap bits in fpcsr
fp_enableTrap()	sets trap bits in fpcsr
fp_init()	init floating Point
fp_int()	FP interrupt dispatcher
fp_signal()	user exception handler
fpclr_stickybits()	clear sticky bits in fpcsr
fpget_excregs()	get Exc Regs
fpget_fpcsr()	get FP Control/Status
fpget_RM()	get rounding mode fpcsr
fpget_stickyBits()	get sticky bits fpcsr
fpset_fpcsr()	set FP Control/Status
fpset_RM()	sets rounding mode
fpset_stickyBits()	sets sticky bits in fpcsr
fpset_excregs()	set Exc Register buffer

Assembly language FPU access

clr_CAUSE()	clears SW bits in CAUSE
get_CAUSE()	returns contents of CAUSE
get_fpcsr()	returns FPU csr
get_cp0epc()	gets epc
get_STATUS()	status register contents
set_CAUSE()	sets CAUSE Register
set_fpcsr()	sets FPU csr

Functions affecting I/D Caches

clear_Dcache()	invalidate portion of Dcache
clear_Icache()	invalidate portion of Icache
config_Dcache()	size of Data cache
config_Icache()	size Instruction cache
flush_Dcache()	invalidates entire Data cache
flush_Icache()	invalidates entire Inst cache
get _mem_conf()	gets memory configuration
size_cache()	finds size of cache

Assembly language TLB access

resettlb()	invalidates tlb entry
ret_tlblo()	returns tlb entry lo reg
ret_tlbhi()	returns tlb entry hi reg
ret_tlbpid()	returns tlb process ID field
set_tlbpid()	sets current tlb pid
tlbprobe()	probes tlb
tlbmapping()	maps tlb entry

Time support module

time_cmd_init()	starts clock
time_init()	init timer drvr
timer_int	clock interrupt routine
time()	returns timer tics
time_it()	times the selected function

Assm language Write Buff Routine

wbflush()	flushes the write buffer

LIBILNK: IDT/SIM LINK LIBRARY

Cache Routines

clear_cache()	clears portion of I and D
flush_cache()	flushes entire I and D

Character Routines

getchar()	inputs a character
putchar()	outputs a character
showchar()	makes character visible

Command Line Interpreter

cli()	Command Line Interpreter
get_range()	parses the range spec
tokenize()	parses the command line

Exit and Reenter Routines

_exit()	exit & return to monitor
promexit()	exit & return to monitor
reinit()	reinitializes monitor
reset()	resets prom monitor
restart()	restarts the debug monitor

Help Screen Routine

help()	prints Help Screen

ROUTINES TO EXTEND IDT/SIM

install_commands()	adds user commands
install_immediate_int	nstalls user interrupt
install_new_dev()	installs new device
install_normal_int()	installs user interrupt

Routines for low level I/O

close()	closes an open device
open()	opens a device
read()	reads data from device
write()	writes data to an device

I/O Control Function

ioctl()	sets I/O flags / calls drivers

Routines to save /restore context

longjmp()	restores setjmp context
setjmp()	saves the current context

Memory configuration routines

get_mem_conf()	returns mem configuration
set_mem_conf()	sets the mem configuration

Formatting print routine

printf()	formatting print routine

Dummy routines for libic

_init_file()	dummy file routine
_init_sbrk()	dummy sbrk routine

String routines

atob()	Ascii string convert
gets()	gets string function
puts()	outputs string to I/O
strcat()	concatenates two strings
strcmp()	compares two strings
strcpy()	copies one string to another
strlen()	returns length of string

LIBIC.A: ANSI STANDARD C LIBRARY

abs()	absolute value of integer
atof()	fp value of an Ascii string
atoi()	integer value of Ascii str
atol()	long value of Ascii str
bsearch()	binary search of a array
div()	rem & quot of int division
ferror()	error during a file operation?
atexit()	routines called at exit time
exit()	Terminate with status
fopen()	open file/ret file stream ptr.
fclose()	close a file
fdopen()	open stream
labs()	absolute value of long arg
ldiv()	rem & quotient of division
free()	free allocated memory
malloc()	memory allocation
realloc()	reallocation of memory
fscanf()	read data from a file
printf()	display data on the std I/O
qsort()	quick sort routine
rand()	generates random number
srand()	seed for random num genr
sbrk()	mem allocation bp routine
scanf()	read data from standard input
sprintf()	output data into a string
sscanf()	read data from a string
memcmp()	compare two memory arrays
memcpy()	memory array copy
memmove()	memory array move
memchr()	ret ptr to first matched char
memset()	place a char in memory array
strlen()	return string length
strcmp()	strings are identical?
strcpy()	copy string
strncpy()	copy n characters of a string
strchr()	ret ptr to first match of a char
strchr()	ret ptr to last match of a char
strcat()	concatenate a strings
strncat()	concatenate strings
strspn()	len of prefix of str
strcspn()	len of prefix of str
strpbrk()	ptr to first occur of any char
strstr()	string a occurs in string b?
strtok()	return tokens
strtod()	convert string to a double
strtol()	convert string to long int

LIBIMATH.A: TRANSCENDENTAL MATH LIBRARY

acosh()	inv hyperbolic cosine of x
acos()	cos -1 (x)
asin()	sin -1(x)
asinh()	inverse hyperbolic sine of x
atan()	tan -1 (x)
atan2()	tan -1 (x/y)
atan2()	tan -1 (x/y)
atanh()	inv hyperbolic tangent of x
cabs()	complex absolute value
exp()	exponential function e^x
hypot()	sqrt (x*x + y*y)
z_abs()	double-complex absolute
cbrt()	cube root of x
cosh()	hyperbolic cosine of x
exp()	exponential function e^x
expm1()	exponent (x - 1)
ceil()	smallest int not < x, double
floor()	largest int not > x, double
rint()	nearest x in dir of round
fmod()	fp remof x/y, sign of x
atan()	tan -1 (x)
cos()	cos of x
exp()	exponential function e^x
log()	natural logarithm ln(x), x>0
log10()	base 10 logarithm, x>0
sin()	sine of x
sqrt()	square root of x, x>= 0
tan()	tan of x
xtoi()	raises x to integer power, i
sim_fpint()	simulate IEEE standard trap
sim_unint	sim FP Unimplemented Op
log()	natural logarithm ln(x), x>0
log10()	base 10 logarithm, x>0
log1p()	log (1 + x)
log__L()	og(1 + x) -2s/s
pow()	x^y
cos()	cos of x
sin()	sine of x
sinh()	hyperbolic sine of x
copysign()	returns x with sign of y
drem()	x - n*y, integer nearest n
finite()	1 = real x; 0 = INF or NAN x
logb()	exponent of x^n
scalb()	x * (2**n) computed for n
sqrt()	square root of x, x>= 0
tan()	tan of x
expm1()	exponent (x - 1)
tanh()	hyperbolic tangent of x
fabs()	absolute value of number
frexp()	returns mantissa;exp in *ptr
isnan()	tests for floating point NaN
ldexp()	returns quantity *2^exp
pow()	x^y

Ordering information

To order the IDT/kit Developer's Package, order the Internal Source License AND order the software on the appropriate source media. The License Agreement is available from your local IDT sales office and is also published in IDT's development tools catalog.

Licenses

Internal Source License **7RS909SLV**

Permits purchase of up to six copies of source code (any media combination) and use of source code to develop run-time binaries on up to six machines at a time, but does not permit inclusion of the run time code in an end product.

Limited Binary Distribution Rights **7RS909BLP-L**

Extension to Internal Source License to permit inclusion of binary code into end product. Source License must be referenced on order or ordered simultaneously. This license permits up to 100 copies to be distributed royalty-free. Additional copies are subject to the royalty .

UnLimited Binary Distribution Rights **7RS909BLU-L**

Extension to Internal Source License to permit distribution of unlimited number of copies of binaries derived from the code.

Maintenance

Maintenance Agreement **7RS909SSY**

One year free updates.

Source Media

IDT/kit source code can be compiled with either the MIPS C compiler or with IDT/c version 3.5 or later. Earlier versions of IDT/c cannot compile this code. The products listed below are media only and must be purchased with license 7RS909SLV above.

Source for 386, MS-DOS **7RS909SBF-L**

Compile with IDT/c C-Compiler. Shipped with both 1.2 MB 5.25" and 1.44 MB 3.5" diskettes.

Source for 386 PC, SCO Xenix **7RS909SXX-L**

Use with IDT/c C-Compiler.

Source for IDT MacStation, on Mac Disc **7RS909SMD-L**

Use with MIPS C Compiler supplied with MacStation or with IDT/c.

Source for MIPS or SPARC machine, DC6150 QIC TAR Tape **7RS909SUU-L**

Use with MIPS C Compiler or with IDT/c.

IDT SOURCE LICENSE TERMS AND CONDITIONS

DEFINITIONS "IDT" means Integrated Device Technology, Inc., with a principle place of business at 2975 Stender Way, Santa Clara, California, 95052, or it's authorized sublicensees.

"IDT-MIPS CPU" means a board or system containing an R3000, R3001, R3051, R3052, R3081 or R4000 chip, or a derivative which is available from IDT, which employs the RISC-based MIPS computer organization, structure and content, including the MIPS instruction set.

"Binaries" means machine executable code for the Subject Software, or any machine executable code derived in any part therefrom.

"Source Code" means the human readable form of any software product including any modifications, enhancements, corrections, updates, translations, interpretations, listings, compilations, or derivations thereof, and all copies and portions in any medium.

"Documentation" means manuals, user guides, and other documentation relative to the Subject Software, including all modifications, updates derivations thereof and changes thereto, whether in written, graphical, human readable, or machine readable form and in any medium.

"Order" means a purchase order, issued and signed by Licensee, and accepted and acknowledged by IDT or its authorized representative, which specifies an ordering number that represents a licenseable software product or a license right.

"Subject Software" means Binaries, Source Code, and Documentation for the Software Product referred to by the description and ordering numbers specified on the Order.

1. LICENSE GRANTS

License Grants and restrictions are as enumerated below, according to the License rights specified on the Order and paid for by Licensee.

1.1 Internal Source License.

If the Licensee has Ordered and paid the "Internal Source License" fee for the subject software, IDT grants to licensee a personal, non-exclusive, non-transferrable, non-assignable fee-bearing license for licensee's internal use of the Source Code and binaries made from the Source Code for the Subject Software at the location specified on the Order. This license grants the right to use, to reproduce a maximum of six copies, and to modify the Source Code, and to make any number of copies of binaries from the Source Code, but does not include any right to sublicense or distribute the source code, or to distribute the resulting binaries at locations not specified, except to the U.S. Government under the conditions specified in paragraph 9.

1.2 Limited Distribution Rights.

If the Licensee has ordered and paid for the "Limited Distribution Right" fee for the Subject Software, then in addition to the right in paragraph 1.1, IDT grants to licensee a personal non-exclusive, non-transferrable right to reproduce, use, and sublicense Binaries derived from the Subject Software, each with one copy of the Documentation, to distributors, resellers, end users, and for internal use at locations not specified on the Order, up to a maximum of 100 copies.

1.3 Binary SubLicense Fee

If Licensee has ordered and paid for the Limited Distribution Right Fee, licensee shall pay to IDT or its authorized representative a Binary SubLicense Fee for each Copy made and/or distributed after the first one hundred. Sufficient SubLicenses to cover the number of copies expected to be made during at least the subsequent quarter shall be ordered by licensee on a Purchase Order. Distribution licenses ordered will be invoiced to licensee at times and in quantities requested by licensee, but licensee shall request invoicing at least once each calendar quarter to cover items distributed.

1.4 Unlimited Distribution Rights

If the Licensee has ordered and paid for the "Unlimited Distribution Right" fee for the Subject Software, then in addition to the right in paragraph 1.1, IDT grants to licensee a personal non-exclusive, non-transferrable right to reproduce, use, and sublicense Binaries derived from the Subject Software, each with one copy of the Documentation, to distributors, resellers, end users, and for internal use at locations not specified on the Order, without limitation to quantity and without any additional fees.

2 RESTRICTIONS

2.1 Copies. Sources, Binaries, and Documentation are copyrighted. Licensee can make copies only as explicitly authorized by this License. Each electronic reproduction of Binaries, Sources, or Documentation, whether for use in a stand-alone CPU, or a CPU in a network, in a multi-user system or in a multi-processor configuration, constitutes a separate copy. All proprietary notices incorporated in, marked on or fixed to Binaries or Documentation by IDT or its suppliers shall be duplicated by licensee on all copies, in whole or in part, in any form and not altered, removed or obliterated. **Binaries which contain code derived in whole or in part from this software shall be copyrighted by Licensee, and shall distributed with a copyright notice to that effect followed by the statement "Portions of this software are Licensed from and Copyrighted by Integrated Device Technology, Inc."**

2.2 Restrictions on Source Code

Licensee must not assign, license, lease, rent, loan, sell or otherwise dispose of Source Code on a temporary or permanent basis. Licensee must not retarget Source Code with the result that binaries may be generated which are capable of operating with computer systems other than an IDT-MIPS CPU.

2.3 End User Agreements.

Any product shipped with binaries derived from the Subject Software, in any form including programmed electronic components, must include a notice that the product includes software, portions of which are copyrighted by Integrated Device Technology, Inc. Licensee is not required to obtain signed sublicense agreements from end users, but is required to provide adequate notice to end users that the software is provided under license, is copyrighted, and is not to be duplicated.

2.7 Reservation. IDT reserves all rights and licenses not expressly granted to licensee.

3. FEES

3.1 Fees. Licensee shall pay IDT or its authorized representative all the applicable license fees for rights which licensee has ordered. All payments shall be made according to the ordinary terms and conditions under which licensee and IDT transact business.

3.2 Taxes and Duties.

In addition to all other fees, licensee shall pay all taxes, fees, duties, licenses, tariffs and levies imposed regarding the Subject Software, other than taxes measured by IDT's Income, unless IDT receives from licensee a valid exemption certificate.

3.3 Records. Licensee shall make and maintain until one (1) year after the termination of this license, complete and accurate records reasonably necessary in accordance with generally accepted accounting principles to ascertain the amount of fees payable to IDT.

3.4 Reports. Licensee shall, with thirty (30) days of the end of each calendar quarter after the effective date of this license, provide to IDT a report certified by an officer of licensee, (a) stating the type and number of copies of Binaries distributed; (b) stating the amount of fees payable by licensee for the quarter, and (c) containing authorization for the payment of such fees. No report is required for Internal Source licenses if no copies have been made beyond those authorized in this LICENSE AGREEMENT.

3.5 Audit. IDT shall have the right, upon reasonable request, to have an independent auditing firm examine, during regular business hours, all such books and records of licensee that may contain information relating to the amount of fees payable to IDT. Any such audit shall be at IDT's expense, and shall be conducted under non-disclosure and shall respect licensee's security requirements, but if such audit discloses an underpayment by licensee of more that five percent (5%) for any calendar quarter, licensee shall reimburse IDT for the cost of the audit.

4. WARRANTY

4.1. Limited Warranty.

IDT warrants to licensee that it has the right to grant the licenses herein granted and that for ninety (90) days from the shipment date the software will comply in material respects with the most recent documentation and that the media, under normal use, will be free from defects in materials and workmanship. This limited warranty shall not apply to any media which has been (a) modified without IDT's authorization, (b) not used in compliance with IDT's instructions, (c) subjected to misuse, improper maintenance, unusual stress or accident or (d) damaged by deviation from applicable environmental specifications.

4.2. Remedies. The entire liability of IDT and its suppliers and the exclusive remedy for licensee under this limited warranty shall be, at IDT's option, either to (a) use reasonable efforts to remedy defects reproducible on unmodified Binaries within a reasonable time, (b) replace media confirmed by IDT as defective or (c) refund the license fees actually received by IDT for the Binaries contained on the defective media.

4.3 Disclaimer.

THE LIMITED WARRANTY AND REMEDIES SET FORTH IN PARAGRAPH 4 ARE THE ONLY WARRANTY MADE BY IDT AND THE ONLY REMEDY OF LICENSEE WITH RESPECT TO THE BINARIES AND DOCUMENTATION. IDT DISCLAIMS ALL OTHER WARRANTIES, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NOTWITHSTANDING ANY FAILURE OF THE ESSENTIAL PURPOSE OF ANY REMEDY, IDT's LIABILITY FOR BREACH OF WARRANTY SHALL BE LIMITED TO A REFUND OF LICENSEE'S LICENSE FEES. IDT AND ITS SUPPLIERS DO NOT WARRANT THAT THE OPERATION OF THE SUBJECT SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE OR THAT ANY PROBLEMS WHICH ARE IDENTIFIED WILL BE CORRECTED.

4.4. Maintenance. Licensee may purchase from IDT a maintenance agreement at the then current price. If Licensee has paid the maintenance agreement fee, then IDT will extend the term of the warranty period for an additional period of time as specified in the maintenance agreement. Any maintenance fee quoted with this license is effective only during the 90-day warranty period. Agreements entered into in the future may be priced differently.

5 TITLE

5.1 Title. Title to Sources, Documentation, all know-how, and all proprietary rights, shall at all times be with IDT. Licensee has no ownership of the Subject Software, other than ownership of the physical media. Title to modifications or enhancments made by licensee shall at all times be with licensee. Licensee agrees to make source code for its modifications available to IDT, without any restrictions on its use, for a single fee not to exceed the License fees paid by Licensee for this license.

5.2 Confidential Information.

Licensee shall not disclose or make available any portion of Binaries, Sources, or Documentation, or any information derived from them to any person or entity except to those of licensee's employees or contractors for whom access is necessary in order to perform their jobs in accordance with this license, except as specifically permitted by this agreement. IDT will not disclose any information relating to Licensee's use of this software to any person or entity except its employees who need to know. The standard of care each party must exercise to meet these obligations is the standard it exercises with respect to its own confidential information of a similar nature, but in no event less than due care. This obligation does not apply to information: (a) known by either party prior to its receipt from the other and not subject to restriction on disclosure; (b) rightfully received by either party from a third party without restriction on disclosure; (c) publicly available other than as a result of any act or omission by the receiving party; (d) publicly disclosed intentionally by the originating party; or (e) independently developed.

5.3 Copyright. A copyright notice on Binaries, Sources, or Documentation does not, by itself, constitute evidence of publication or public disclosure.

6 EXPORT CONTROLS

6.1 Before exporting or re-exporting any Binaries or Documentation, licensee must: (a) fully comply with all U.S. export control laws and regulations, or (b) take reasonable precautions, including seeking the advice of foreign legal counsel where necessary, to protect the proprietary rights of IDT and its suppliers in each foreign country in which the Subject Software is to be used.

7 LIABILITY

7.1. Defence IDT shall defend, at its expense, any action brought against licensee to the extent that it is based on a claim that the Subject Software, when used in accordance with this license, infringes a United States patent, copyright, trademark, or a trade secret or other proprietary right, and IDT shall pay any costs, settlements and damages finally awarded; provided that licensee promptly notifies IDT in writing of any claim, gives IDT sole control of the defense and settlement and provides all reasonable assistance in connection therewith. If any of the Subject Software is finally adjudged to so infringe, or in IDT's opinion such a claim is likely to succeed, IDT shall, at its option, either; (a) procure for licensee the right to continue using the Subject Software, (b) modify or replace the Subject Software to make them non-infringing, or (c) refund the fee paid upon return of the Subject Software. IDT shall have no liability regarding any claim arising out of: (a) use of other than the then current, unaltered release of Binaries, unless the infringing portion is also in the then current, unaltered release; (b) use of Binaries in combination with non-IDT supplied software, data or equipment if the infringement was caused by such use or combination, (c) any modification or marking of Binaries not specifically authorized in writing by IDT or (d) third party software.

THE FOREGOING STATES THE ENTIRE LIABILITY OF IDT AND ITS SUPPLIERS AND THE EXCLUSIVE REMEDY FOR LICENSEE RELATING TO INFRINGEMENT OR CLAIMS OF INFRINGEMENT OF ANY PATENT, TRADEMARK, COPYRIGHT, TRADE SECRET OR OTHER PROPRIETARY RIGHT.

8 LIMITATION OF LIABILITY

8.1 EXCEPT FOR INTELLECTUAL PROPERTY INDEMNITY IN PARAGRAPH 7, THE LIABILITY OF IDT AND ITS SUPPLIERS SHALL IN NO EVENT EXCEED THE FEE RECEIVED FROM LICENSEE FOR THE SOFTWARE COMPLAINED OF. IN NO EVENT SHALL IDT OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL OR EXEMPLARY DAMAGES HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY WHETHER CONTRACT, STRICT LIABILITY OR TORT ARISING IN ANY WAY OUT OF THIS LICENSE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY REMEDY.

9 GOVERNMENT USE

9.1 Licensee shall not, without the prior written consent of IDT, deliver Binaries, Sources, or Documentation, any portion thereof, or any technical data relating thereto, to any branch or agency of the United States Government without a written predetermination that such items will be protected by limited or restricted rights as set forth in DOD FAR 52.227-7013 or equivalent rights and without taking all required actions to preserve such rights including, without limitation; (a) marking Binaries or Sources with the then currently prescribed Restricted Rights Legend, (b) marking technical data with the then currently prescribed Limited Rights Legend, and (c) ensuring that the contract with the government agency contains the standard Department of Defense "Rights in Technical Data and Computer Software: clause at DOD FAR 52.227-7013 and the "Restrictive Markings on Technical Data" clause at DOD FAR 52.227-7018 or the equivalent clauses for other government agencies.

10 TERM

10.1 Term. This license is effective for a term of five years from the date of signature by both parties. The license will renew automatically each year thereafter unless 90 days written notice to terminate is provided by either party. Either party may terminate this License effective immediately upon written notice to the other party if the other party files a voluntary petition in bankruptcy or otherwise seeks protection under any law for the protection of debtors; has a proceeding instituted against it under any provision of the bankruptcy laws which is not dismissed with sixty days; is adjudged a bankrupt; has a court assume jurisdiction of its assets under a reorganization act; has a trustee or receiver appointed by a court for all or a substantial portion of its assets; becomes insolvent, suspends or ceases to do business; makes an assignment of all or a substantial portion of its assets for the benefit of it creditors; or admits in writing its inability to pay its debts as they become due. Notwithstanding the foregoing terms, the license will terminate automatically if any breach or violation by licensee is not cured within thirty days after written notice thereof from IDT.

10.2. Termination. Upon termination, licensee must return to IDT or destroy all Binaries, Sources, and Documentation in licensee's possession, and shall cease distribution of any binaries derived from it. Binary subicenses granted to End Users by licensee remain in effect.

10.3. Survival. The provisions of Sections 4 (Limited Warranty and Disclaimer), 5 (Title and Proprietary Rights, 7 (Intellectual Property Indemnity), 8 (Limitation of Liability, 11.1 (Governing Law), 11.3 (Attorneys' Fees), 11.4 (Assignment) and 11.5 (Enforcement) shall survive the expiration, cancellation or termination of this license.

11 GENERAL PROVISIONS

11.1 Governing Law.

This license shall be governed by the laws of the State of California, excluding its conflict of laws rules.

11.2 Modification.

No modification to the license, nor any waiver or any rights, shall be effective unless consented to in writing as an amendment to this LICENSE AGREEMENT, and the waiver of any breach or default shall not constitute a waiver of any other right or of any subsequent breach or default.

11.3. Attorneys' Fees.

In the event of any dispute resolution proceeding between the parties, the prevailing party shall be entitled to recover its costs and reasonable attorneys' fees.

11.4 Assignment Neither IDT nor licensee shall assign or transfer this Agreement nor any right or license hereunder without the prior written consent of the other party, and assignment permitted shall be subject to the written consent of the assignee to all the terms of this Agreement.

11.5 Enforcement In the event of a breach or default of this license by licensee, IDT shall have a right to seek enforcement of this license directly against licensee without the need for any acts or omissions by IDT or the participation of IDT.

11.6 Severability If any provision of this License shall be ruled unenforceable, then the remainder shall be enforced to the extent permissible.

The MacStation 3: A MIPS Development Environment Built Around an Apple™ Macintosh™ CPU.

The MacStation is an R3000-based CPU card that plugs into a Macintosh II on the NuBus, and runs MIPS version of AT&T UNIX SVR3 (MIPS RISC/os). The product includes a number of Macintosh programs that provide interconnectivity between the Macintosh and UNIX systems, and the UNIX file systems are encapsulated into Macintosh files, so that can co-exist on the same hard disk as Macintosh files without partitioning.

The UNIX is a complete version, with NFS, X11, and BSD 4.3. It also includes complete on-line documentation.

Two performance levels are available: one at approximately 15 mips and the other at 25 mips. The lower performance version includes 8 MB of local DRAM; the higher includes 16 MB of local DRAM. Both versions have a Macintosh-compatible SCSI port for peripheral or hard disk support. A Macinstosh hard disk can be connected to the SCSI port, reformatted for UNIX, and used as a dedicated UNIX drive for higher performance than is possible going through the Macintosh's SCSI port.

For more details on the MacStation, refer to the MacStation 3 Catalog and the MacStation Technical Briefs. The MacStation Developer's systems in this catalog are only available to users who are developing systems using R3000, 3500, or 30xx CPUs. They are not listed in the regular MacStation 3 End User Catalog.

How to Order a MacStation Developer's System

1. You must agree to TWO separate licenses: one for the MacStation UNIX software, and one for the Source Code for IDT/sim and IDT/kit. The MacStation License is at the end of this catalog, and *must be signed and returned with your order*. The Terms for the IDT/sim and IDT/kit Source Code are on pages 10 - 15 of this catalog. By ordering the two source licenses (7RS901SLV and 7RS909SLV) you are purchasing Internal Source Use rights as defined by this license, and you don't need to sign and return anything other than your Purchase Order. If you need to modify the terms of the license in any way, you must have a written agreement from IDT first. Contact your IDT sales office for assistance.
2. Place an order for the single order code for the Developers System (e.g., 7RS533DCI), at the price shown.
3. Your shipment will be staged at the factory and delivered to the carrier in one shipment of multiple boxes. It should all arrive at about the same time.

Integrated Device Technology, Inc.

MacStation™ 3 RISC WORKSTATION IN A Macintosh®

IDT7RS503

FEATURES:

- 15 or 25 MIPS RISC Computer Add–In Board for Macintosh II Computers
- Includes AT&T UNIX® SVR3 Operating System with BSD 4.3, NFS, X11, and Motif
- Supplied with MIPS C-Compiler, Assembler, and Symbolic Debugger
- Uses all Macintosh peripherals for I/O
- Includes Macintosh-independent SCSI and serial I/O ports
- Multifinder and System 7 compatible
- Available as add–in board or as completely configured systems

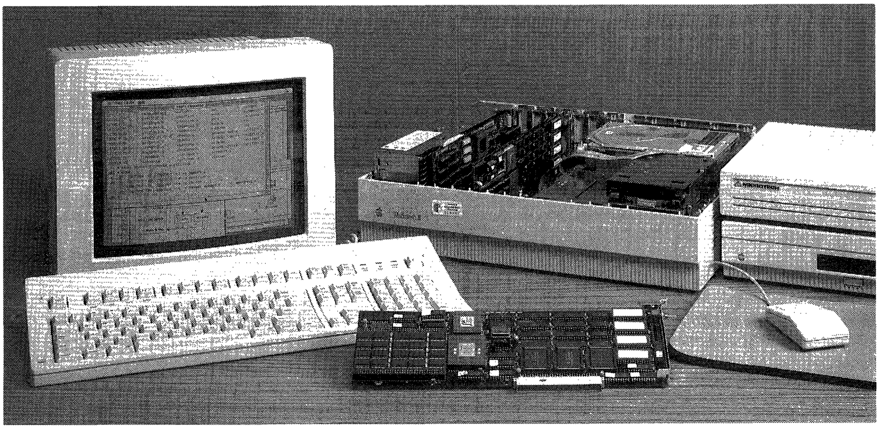
R3000 COMPUTER PLUGS INTO A MACINTOSH II

IDT's MacStation™ 3 is a high performance R3000-based workstation consisting of a MIPS® R3000 RISC CPU board that plugs into a Macintosh II computer, a complete AT&T SVR3 UNIX® operating system, and a collection of Macintosh programs used to communicate between the UNIX and Mac operating systems.

The UNIX software is MIPS RISC/os v 4.5.2, MIPS port of AT&T Unix. It is supplied with TCP/IP, NFS®, X11G3, and Motif. The UNIX software can support multiple user sessions, using either Telnet or X, running one or more Macs or external terminals.

A Macintosh application, IDT/console, is used as a console terminal and I/O handler for the UNIX OS. Opening the application creates a terminal window on the Macintosh from which UNIX can be booted. File I/O commands from UNIX are intercepted by IDT/console, which reads and writes UNIX files encapsulated inside Macintosh files. The encapsulated files can reside on the same disks as Macintosh files without partitioning. With a single command, a file can be moved between the Macintosh and Unix environments. An optionally available package, Intercon's NFS/share™, can be used to mount the Unix files on the Macintosh.

Other Macintosh software includes IDT/envy™, an Extension that provides TCP/IP paths among the Mac, UNIX, and an Ethernet card; NSCS Telnet, a telnet terminal application for the Macintosh; Consulair EDIT, a Macintosh text editing program; and MacTCP.



2891 drw 01

FEBRUARY 1992

MACSTATION 3 HARDWARE

RISC CPU Card

The MacStation is available with either of two CPU cards. One is approximately 15 MIPS performance, and includes 8 MB of local DRAM, 2 serial I/O ports and a SCSI port. The processor on this card runs at 20 MHz, and includes 16 KB each of Instruction and Data Cache. The other card is approximately 25 MIPS in performance, and includes 16 MB of local DRAM, 2 serial I/O ports, and a SCSI port. The processor runs at 25 MHz, and includes 64 KB each of Instruction and Data Cache.

The SCSI port can be used to connect hard disks with the UNIX file systems on them. This provides higher performance than that available when the Mac's disks are used for the UNIX files.

The serial I/O ports can be used for a separate console terminal or for printers or other common I/O devices. Drivers are included for a number of common devices.

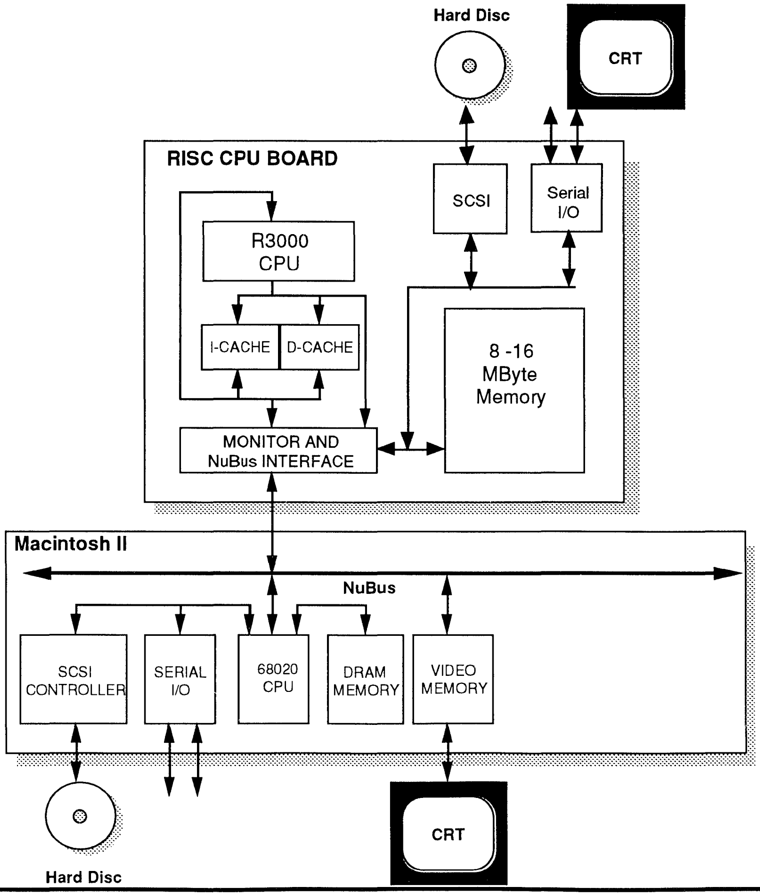
MACSTATION 3 SOFTWARE

UNIX O/S

The UNIX OS is AT&T SVR3 with BSD 4.3 extensions and fast file system. This is a full-featured UNIX, including on-line manuals, SMTP, reconfiguration files, and the MIPS C-compiler. Any application which runs under MIPS RISC/os will run directly on this OS. Ports of other SVR3 applications are easy, but generally require recompiling for the MIPS hardware. The UNIX software includes NFS file sharing services, the X11 library, and Motif (MIPS RISCwindows). The UNIX file systems are encapsulated in Macintosh files and may exist on any number of Mac disks. Swap space is preset to 40 MB, but can be changed easily.

C Compiler

The C compiler included with the MacStation is the MIPS C compiler, noted for extremely efficient optimization of code. A switch in the compiler determines whether K&R or ANSI C is supported.



2891 drw 02

Drivers

Source code is supplied for a number of I/O devices, including terminal, disk, and network. A reconfiguration directory and make files are available to rebuild the kernal with new or modified drivers.

MACINTOSH SOFTWARE

All supplied Macintosh software is compatible with System 6.07 and with System 7.0.

IDT/console

IDT/console is an application that runs on the Macintosh under MultiFinder, and provides a terminal emulation window for the UNIX console, as well as I/O services to UNIX for file systems on Macintosh disks.

MacTCP

This Apple communication tool adds TCP/IP capability to the Macintosh.

IDT/envY

IDT/envY is a Macintosh extension that provides the UNIX and Macintosh sides two different IP addresses, so they can communicate with each other using TCP/IP protocols. This communication takes place over the NuBus, so it is extremely fast and reliable. No Ethernet card is required for the Macintosh-UNIX path. IDT/envY also permits a single Macintosh Ethernet card to serve both CPUs.

Telnet

Telnet is a Macintosh application that opens a terminal window on the Macintosh through which new UNIX sessions can be started. Multiple Telnet windows can be open at once, each running different processes.

EDIT

Consulair EDIT is a Macintosh text editing program, well suited for writing C source code. The text files are easily passed to the Unix side for compilation and debug.

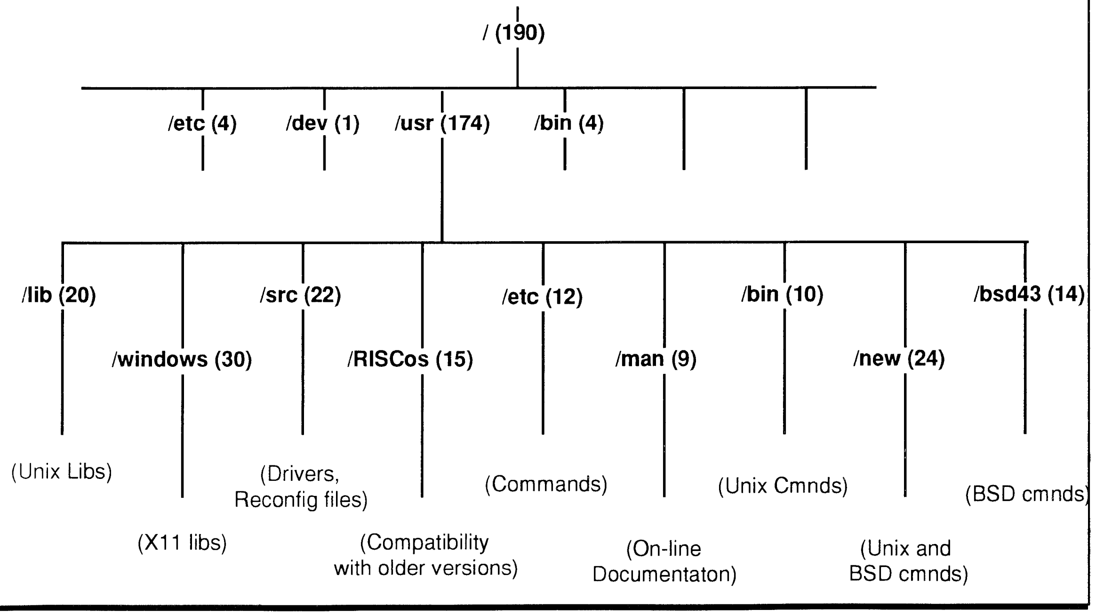
NFS for the Macintosh

NFS communications is included with the UNIX package, so files can be mounted from remote servers. Intercon's NFS/share™ is available as an option on the Macintosh side. NFS/share allows UNIX file systems to be mounted and opened on the Mac like AppleShare volumes.

X–Windows

An X–11 Library and Motif client software is included for the UNIX side. Apple's MacX is available separately for the Macintosh side so that X–windows can be opened on the Macintosh screen.

Major Components of UNIX file systems and approximate size in MBytes.
Swap file (typically 40 MB) not shown



FOR R3000 FAMILY DEVELOPERS:

LOWEST PRICED DEVELOPMENT SYSTEM

R3000 and R3051 Developer's System

This Development System offers more power for fewer dollars than any other choice you can make. It is built around the Apple Macintosh IIx CPU – a six slot 68030 machine that can be field upgraded to a IIfx. It is bundled with the MIPS C compiler; IDT's C compiler for embedded systems; and source code for IDT/sim, IDT's popular debug monitor. There are a limited number of Macintosh IIx computers in stock.

MIPS Software Development WorkStation (Mac IIx version)

7RS503/15	25 MIPS CPU Board and IDT/ux RTU Macintosh IIx, 4 MB RAM, 80 MB HD, Ext. Keyboard	
7RS510D	425 MB External SCSI hard drive with UNIX software	
7RS581	Documentation, MIPS C compiler	
7RS550-19x	19" Monochrome monitor with card	
7RS903FBMD	IDT/c Cross C-compiler with floating point library	
7RS901SLV	Source License, IDT/sim debug kernal	
7RS901SMD	IDT/sim media and documentation	
7RS909SLV	Source License, IDT/kit kernel toolkit	
7RS909SMD	IDT/kit media and documentation	
TOTAL VALUE	**OF SOFTWARE AND HARDWARE**	**$24,958**
	YOUR SAVINGS:	**($9,963)**
7RS533DX	**DEVELOPER'S SYSTEM PRICE**	**$14,995**

Prices for delivery in U.S.A only.

FOR R3000 FAMILY DEVELOPERS:

SAVE nearly $10,000!

It's Like getting a Macintosh for Free!

R3000 and R3051 Developer's System

This is a special configuration of the new MacStation 3 just for qualified developers writing code around one of the MIPS-based CPUs available from IDT. This discount is not available under any other conditions, and the system must be ordered exactly as shown. Other items may be added from the standard MacStation price list.

MIPS Software Development WorkStation (Mac IIci version)

7RS503/25	25 MIPS CPU Board and IDT/ux RTU Macintosh IIci, 5 MB RAM, 80 MB HD, Ext. Keyboard
7RS510C	UNIX software on CD-ROM
7RS581	Documentation, MIPS C compiler
7RS575A	CD-ROM drive (Apple) with cable
7RS540-425LC	425 MB External SCSI hard drive
7RS550-19x	19" Monochrome monitor with card
7RS560-Thin	Thin-net Ethernet controller
7RS903FBMD	IDT/c Cross C-compiler with floating point library
7RS901SLV	Source License, IDT/sim debug kernal
7RS901SMD	IDT/sim media and documentation
7RS909SLV	Source License, IDT/kit Kernel Integration Toolkit
7RS909SMD	IDT/kit media and documentation

TOTAL VALUE	**OF SOFTWARE AND HARDWARE**	**$29,702**
	YOUR SAVINGS:	**($9,707)**
7RS533DCI	**DEVELOPER'S SYSTEM PRICE**	**$19,995**

Prices for delivery in U.S.A only.

MacStation Software LICENSE AGREEMENT

DEFINITIONS

"IDT" means Integrated Device Technology, Inc., with a principle place of business at 3236 Scott Boulevard, Santa Clara, California.

"MIPS" means MIPS Computer Systems, Inc., with a principle place of business at 928 Arques Avenue, Sunnyvale, California.

"APPLE" means Apple Computer, Inc., with a principle place of business at 20525 Mariani Ave, Cupertino, California.

"Binaries" means machine executable code for the Subject Software designated in the LICENSE AGREEMENT, or any machine executable code derived in any part therefrom.

"Documentation" means manuals, user guides, and other documentation relative to the Subject Software, including all modifications, updates derivations thereof and changes thereto, whether in written, graphical, human readable, or machine readable form and in any medium.

1. LICENSE GRANTS

IDT grants to Licensee a personal, non-exclusive, non-transferrable, non-assignable, perpetual (except as provided in paragraph 10) license, for Licensee's internal use only, to execute one copy at a time of the Binaries identified in the LICENSE AGREEMENT on a computer system or board product designated in the LICENSE AGREEMENT, and to reproduce a maximum of four (4) copies of the Binaries and Documentation for backup and archive, and without any right to sublicense or distribute.

2. RESTRICTIONS

2.1 Restrictions on Copies. Binaries, and Documentation are copyrighted. Licensee can make copies only as explicitly authorized by the LICENSE AGREEMENT. Each electronic reproduction of Binaries, Sources, or Documentation, whether for use in a stand-alone CPU, or a CPU in a network, in a multi-user system or in a multi-processor configuration, constitutes a separate copy for which a license fee shall be payable. All proprietary notices incorporated in, marked on or fixed to Binaries or Documentation by IDT, MIPS, or their suppliers shall be duplicated by Licensee on all copies, in whole or in part, in any form and not altered, removed or obliterated.

2.2 Restrictions on Binary License Types. Licensee must not, among others, adapt, translate, reverse engineer, decompile, disassemble or create derivative works based on the Binaries. Licensee must not modify binaries. Licensee must not, among others, transfer, assign, sublicense, lease, rent, loan, sell or otherwise dispose of Binaries on a temporary or permanent basis.

2.3 Reservation. IDT reserves all rights and licenses not expressly granted to Licensee.

3.1 Fees. Licensee shall pay IDT or its authorized representative all the applicable license fees. The terms of payment are as agreed upon between Licensee and IDT or it's authorized representative.

3.2 Taxes and Duties. In addition to all other fees, Licensee shall pay all taxes, fees, duties, sublicenses, tariffs and levies imposed regarding the Subject Software, other than taxes measured by IDT's Income, unless IDT receives from Licensee a valid exemption certificate.

4.1. Limited Warranty. IDT warrants to Licensee that for ninety (90) days from the shipment date the Binaries will comply in material respects with the then current user documentation and that the media, under normal use, will be free from defects in materials and workmanship. This limited warranty shall not apply to any media which has been (a) modified without IDT's authorization, (b) not used in compliance with IDT's instructions, (c) subjected to misuse, improper maintenance, unusual stress or accident or (d) damaged by deviation from applicable environmental specifications.

4.2. Remedies. The entire liability of IDT, MIPS, APPLE and their suppliers and the exclusive remedy for Licensee under this limited warranty shall be, at IDT's option, either to (a) use reasonable efforts to remedy defects reproducible on unmodified Binaries within a reasonable time, (b) replace media confirmed by IDT as defective or (c) refund the License Fee actually received by IDT for the Binaries contained on the defective media.

4.3 Disclaimer. THE LIMITED WARRANTY AND REMEDIES SET FORTH IN PARAGRAPH 4 ARE THE ONLY WARRANTY MADE BY IDT AND THE ONLY REMEDY OF LICENSEE WITH RESPECT TO THE BINARIES AND DOCUMENTATION. MIPS AND ITS SUPPLIERS MAKE NO WARRANTIES WHATSOEVER. IDT DISCLAIMS ALL OTHER WARRANTIES, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NOTWITHSTANDING ANY FAILURE OF THE ESSENTIAL PURPOSE OF ANY REMEDY, IDT's LIABILITY FOR BREACH OF WARRANTY SHALL BE LIMITED TO A REFUND OF LICENSEE'S LICENSE FEES. IDT AND ITS SUPPLIERS DO NOT WARRANT THAT THE OPERATION OF THE SUBJECT SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE OR THAT ANY PROBLEMS WHICH ARE IDENTIFIED WILL BE CORRECTED.

5.1 Title. Title to all Binaries, Sources, and Documentation, all know-how, and all proprietary rights, shall at all times be with IDT, MIPS, APPLE or their suppliers, as applicable. Licensee has no ownership of the Subject Software, other than ownership of the physical media.

5.2 Confidential Information. Licensee shall not disclose or make available any portion of Binaries, Sources, or Documentation, or any information derived from them to any person or entity except to those of Licensee's employees for whom access is necessary in order to perform their jobs in accordance with this License. The standard of care Licensee must exercise to meet these obligations is the standard it exercises with respect to its own confidential information of a similar nature, but in no event less than due care. This obligation does not apply to information: (a) known by Licensee prior to its receipt from IDT and not subject to restriction on disclosure; (b) rightfully received by Licensee from a third party without restriction on disclosure; and (c) publicly available other than as a result of any act or omission of Licensee.

5.3 Copyright. A copyright notice on Binaries, Sources, or Documentation does not, by itself, constitute evidence of publication or public disclosure.

6 Export Controls. Before exporting or re-exporting any Binaries or Documentation, Licensee must: (a) fully comply with all U.S. export control laws and regulations, or (b) take reasonable precautions, including seeking the advice of foreign legal counsel where necessary, to protect the proprietary rights of IDT, MIPS and their suppliers in each foreign country in which the Subject Software is to be used.

7 INTELLECTUAL PROPERTY INDEMNITY IDT shall defend, at its expense, any action brought against Licensee to the extent that it is based on a claim that the Subject Software, when used in accordance with this License, infringe a United States patent or a United States copyright, and IDT shall pay any costs, settlements and damages finally awarded; provided that Licensee promptly notifies IDT in writing of any claim, gives IDT sole control of the defense and settlement and provides all reasonable assistance in connection therewith. If any of the Subject Software are finally adjudged to so infringe, or in IDT's opinion such a claim is likely to succeed, IDT shall, at its option, either; (a) procure for Licensee the right to continue using the Subject Software, (b) modify or replace the Subject Software to make them non-infringing, or (c) refund the fee paid upon return of the Subject Software. IDT shall have no liability regarding any claim arising out of: (a) use of other than the then current, unaltered release of Binaries, unless the infringing portion is also in the then current, unaltered release; (b) use of Binaries in combination with non-IDT supplied software, data or equipment if the infringement was caused by such use or combination, (c) any modification or marking of Binaries not specifically authorized in writing by IDT or (d) third party software.

THE FOREGOING STATES THE ENTIRE LIABILITY OF IDT, MIPS, APPLE AND THEIR SUPPLIERS AND THE EXCLUSIVE REMEDY FOR LICENSEE RELATING TO INFRINGEMENT OR CLAIMS OF INFRINGEMENT OR ANY PATENT, TRADEMARK, COPYRIGHT, MASK WORK RIGHT, TRADE SECRET OR OTHER PROPRIETARY RIGHT.

8 LIMITATION OF LIABILITY EXCEPT FOR INTELLECTUAL PROPERTY INDEMNITY IN PARAGRAPH 7, THE LIABILITY OF IDT, MIPS, APPLE AND THEIR SUPPLIERS SHALL IN NO EVENT EXCEED THE FEE RECEIVED BY IDT FROM LICENSEE FOR THE SOFTWARE COMPLAINED OF. IN NO EVENT SHALL IDT, MIPS, APPLE OR THEIR SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL OR EXEMPLARY DAMAGES HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY WHETHER CONTRACT, STRICT LIABILITY OR TORT ARISING IN ANY WAY OUT OF THIS LICENSE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY REMEDY.

9 GOVERNMENT USE Licensee shall not, without the prior written consent of IDT, deliver Binaries, Sources, or Documentation, any portion thereof, or any technical data relating thereto, to any branch or agency of the United States Government without a written predetermination that such items will be protected by limited or restricted rights as set forth in DOD FAR 52.227-7013 or equivalent rights and without taking all required actions to preserve such rights including, without limitation; (a) marking Binaries or Sources with the then currently prescribed Restricted Rights Legend, (b) marking technical data with the then currently prescribed Limited Rights Legend, and (c) ensuring that the contract with the government agency contains the standard Department of Defense "Rights in Technical Data and Computer Software: clause at DOD FAR 52.227-7013 and the "Restrictive Markings on Technical Data" clause at DOD FAR 52.227-7018 or the equivalent clauses for other government agencies.

10 **TERM**This license is effective until terminated by either party upon 90 days written notice. This license will terminate automatically without notice from IDT, MIPS, APPLE or their suppliers upon any breach or violation. Upon termination, Licensee must return to IDT or destroy all Binaries and Documentation. The provisions of Sections 4 (Limited Warranty and Disclaimer), 5 (Title and Proprietary Rights, 7 (Intellectual Property Indemnity), 8 (Limitation of Liability, 11.1 (Governing Law), 11.3 (Attorneys' Fees), 11.4 (Assignment) and 11.5 (Enforcement) shall survive the expiration, cancellation or termination of this License.

11 GENERAL PROVISIONS

11.1 **Governing Law.** This License shall be governed by the laws of the State of California, excluding its conflict of laws rules.

11.2 **Modification.** No modification to the License, nor any waiver or any rights, shall be effective unless consented to in writing as an amendment to the LICENSE AGREEMENT, and the waiver of any breach or default shall not constitute a waiver of any other right or of any subsequent breach or default.

11.3. **Attorneys' Fees.** In the event of any dispute resolution proceeding between the parties, the prevailing party shall be entitled to rev cover its costs and reasonable attorneys' fees.

11.4 **Assignment.** Licensee shall not assign or transfer this Agreement nor any right or license hereunder without the prior written consent of IDT, and assignment permitted shall be subject to the written consent of the assignee to all the terms of this Agreement.

11.5 **Enforcement.** In the event of a breach or default of this License by Licensee, IDT shall have a right to seek enforcement of this License directly against Licensee without the need for any acts or omissions by IDT or the participation of IDT.

11.6 **Entire Agreement.** This LICENSE AGREEMENT constitutes the entire and exclusive agreement between the parties with respect to this subject matter. All previous discussions and agreements with respect to this subject matter are superseded by the LICENSE AGREEMENT. However, nothing in this agreement will create more restrictive conditions than those available to the licensee by virtue of other agreements with IDT, MIPS, or APPLE.

12. SUBJECT SOFTWARE AND LICENSED MACHINE

The following software is licensed only for use on a MacStation, which consists of a single Macintosh computer and a single IDT NuBUS CPU card utilizing the MIPS Instruction Set architecture. Software: MIPS RISC/os, Sun NFS, MacTCP, IDT/envY, IDT/console, MIPS RISCwindows, MacX, NFS/share.

Licensee (Company Name) ______________________

Signature ______________________

Print Name ______________________

Title ______________________

Date ______________________

Mailing Address ______________________

____

____

TRADEMARKS

MacStation, RISC CPU SubSystem, RISController, Real8 and TargetSystem are Trademarks of Integrated Device Technology, Inc.
Apple, Macintosh, AppleTalk, LaserWriter, A/UX, MultiFinder are registered Trademarks of Apple Computer, Inc. UNIX is a registered trademark of Unix System Laboratories. MIPS, RISC/os, and RISCompiler are trademarks of MIPS Computer Systems, Inc. NuBus is a trademark of Texas Instruments, Inc. MS-DOS is a registered trademark of Microsoft Corporation. TrueImage and Windows are trademarks of Microsoft Corporation. TrueType is a trademark of Apple Computer. Postscript is a trademark of Adobe Systems.
PeerlessPage is a trademark of The Peerless Group.

Integrated Device Technology, Inc.
RISC SubSystems Division
2975 Stender Way, Santa Clara, CA 95052 • Phone (800) 438–2667 ext 5627 • Fax (408) 988–5600

CAT-RSD-00001