Integrated Device Technology, Inc.

IDT/sim™ SYSTEM INTEGRATION MANAGER ROMABLE DEBUGGING KERNEL

IDT79S901

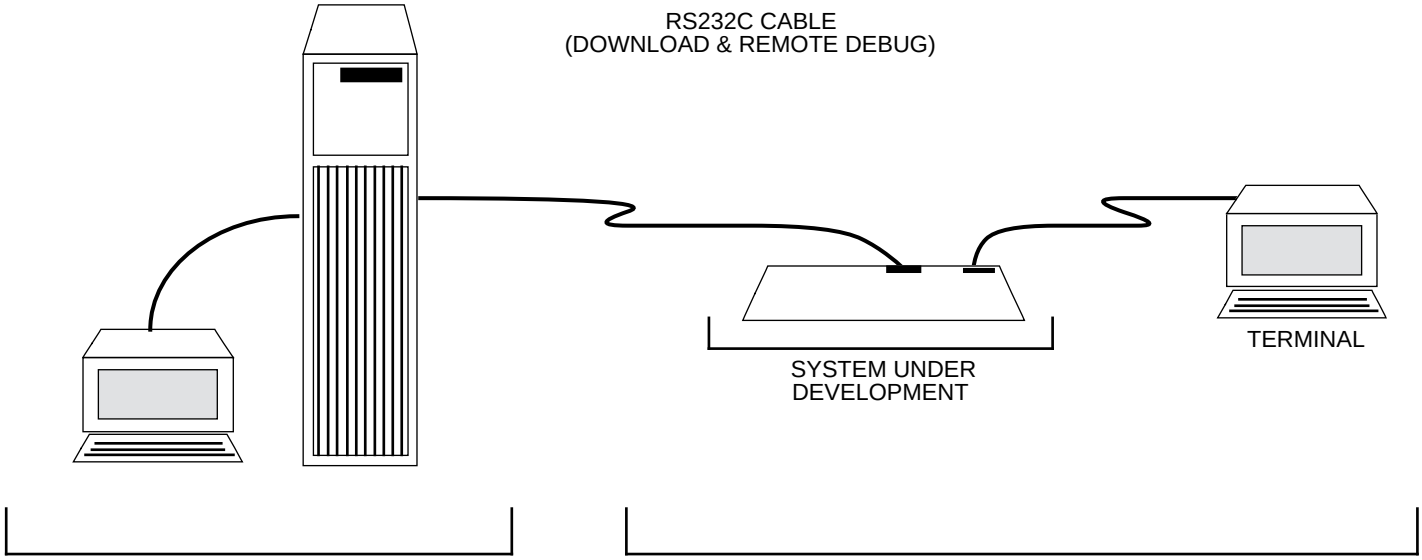
FEATURES

- Complete source code provided
- Robust debug monitor
- Supports remote source-level debug
  GDB—IDT/c tool chain
  DBX—MIPS tool chain
- Remote file access—connects target to remote host file system
- Ethernet and Centronics support for fast download
- Diagnostic tests for memory, cache, MMU, FPA, and system
- Adaptable to systems with or without hardware floating point accelerator
- Includes a variety of device drivers
- Easy to add new user interface commands and I/O device drivers

Powerful Tool for Integration of RISController Based systems

The IDT79S901 System Integration Manager (IDT/sim) is a ROMable debug monitor product that permits convenient control and debug of RISC systems built around IDT's R30xx RISController™, R4400™, R4600™, R4700™ and R4650™ CPUs. Facilities are included to operate the CPU under controlled conditions: examining and altering the contents of memory, manipulating and controlling R30xx, R4xxx 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 which requires only a UART and ROM to be functional for performing the initial debugging and integration of new hardware.



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, R4400, R4600, R4700 and R4650. The software is supplied in EPROMs on most IDT RISC Development products, and may be purchased in source-code form so it can be modified, 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 is needed for the binary code, and 71KB of RAM is 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. The diagnostic suite includes: main memory tests which exercise all address and data paths; a cache memory test which runs memory tests on both caches, checks tag memory, and verifies that instructions can be executed from cache; a system test which checks the ability to read and write full words, half-words, and bytes and checks the cache operation for valid, hit/miss, and invalidation; a MMU test which checks the operation of the TLB inside the CPU; and a Floating Point test which tests the functionality of the on-chip FPA, 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 IDT/c, the MIPS RISCross Compiler tools, or several third party compiler toolchains.

The IDT/sim console may also be used as a terminal to a software development system accessed through a second serial port. On targets which implement ethernet, utilities are also available to support ethernet downloads, and remote file access.

Debug Commands

A variety of commands are included in IDT/sim to support software/hardware debug. IDT/sim commands can be grouped into several categories, including: 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 gdb on and IDT/c Host or dbx on a MIPS RISC/os system; and Communications (remote file access, terminal emulator and set baud rate).

Run-Time Support

IDT/sim includes over 50 functions that can be called by user's programs to perform common I/O and R30xx, R4xxx control operations. A complete list of commands is listed later in this document.

Feature Set

R4650 Orion Support: IDT/sim has been upgraded to support the latest 64-bit RISController family member, the 79R4650. This support has been implemented as a series of "IFDEF" options to the base 32-bit source tree. This approach allows a common set of features across 32 and 64-bit targets with a single development environment.

R3710/15 Support: IDT/sim now supports IDT79R3710/15 Laser printer integrated system controller for IDT R30xx RISController family. Features supported are: ROM controller, DRAM controller, Centronics, Timer, Printer interface diagnostics.

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

Source-level Debug: Fully integrated with IDT's new compiler toolchain (IDT/c Version 5.1) which supports source-level debug using gdb. IDT/sim supports gdb as a front end (with full access to /sim's commands) or use of an ASCII terminal in a stand-alone mode.

Remote File Access: IDT/sim has implemented features to allow connection of the target with a remote host file system allowing file transfer between target and host at run time. As an example, this can be useful for accessing large data images residing on a host without linking them with the application.Ethernet support is also included.

Trace Facility: Traces the memory accesses of a user program. Provides for tracing the path of execution, reads-from memory, or writes-to memory. Trace qualifiers allow the tracing of a specific instruction or class of instructions or references to particular memory ranges. 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 R30xx, and R4xxx family mnemonics.

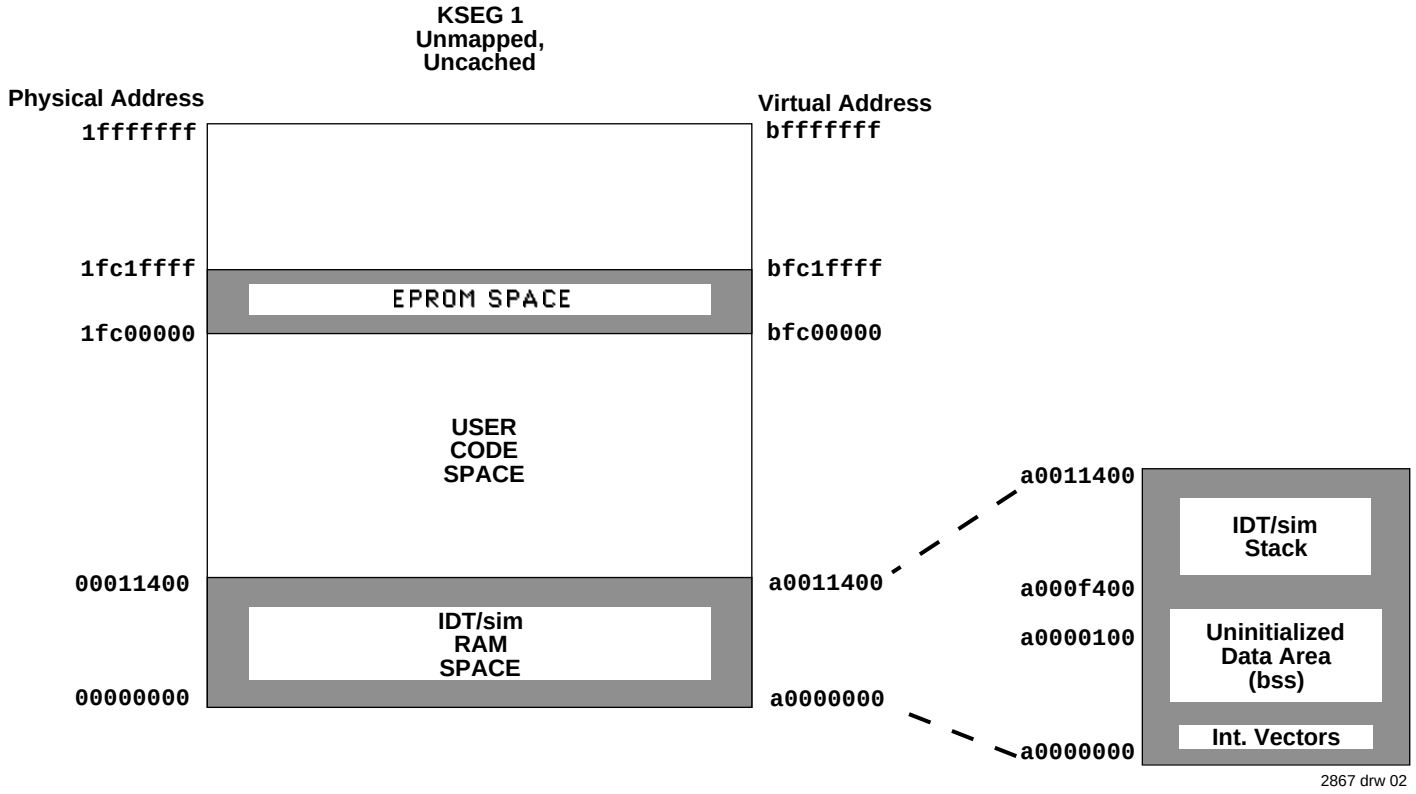


Figure 2 IDT/sim Memory Map

Figure 2 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 115KB of EPROM space, and is normally placed in 128KB of EPROM. IDT/sim uses main memory to store interrupt vectors, variables, and a stack. Approximately 71KB of RAM space is reserved for this data.

IDT/sim COMMANDS

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

benchmark/bm
Facilitates benchmarking

brk|b [addresslist]
Set/display breakpoints

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

call|ca <address> [arg1 arg2 ... arg8}
Call subroutine with up-to 8 arguments

checksum|cs
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]
Print out 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 hosfor 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
Displays the 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

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 though to a second serial port

tex [RANGE]
Exclude tracing calls to RANGE

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

tlbflush|tf [RANGE]
Displays the current process identifier (pid)

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

tlbpid/ti [pid]
Set/display TLB PID

tlbptov/tp|tm <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

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
timer_start
timer_stop
write

DEVICE DRIVERS (INCLUDED IN SOURCE CODE)

68681/2681 DUART
8530 SCC
SCSI
Centronics Parallel
8254 Timer/Counter
8251 UART

ORDERING INFORMATION

To order an IDT board-level product, see 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 79S901SL

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 79S901BDR-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 Unlimited Binary Distribution Rights.

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 79S901SL, listed above.

Source for 386/486PC, MS-DOS 79S901DOS

Compile with IDT/c C-Compiler. Shipped with 1.44MB 3.5" diskettes.

Source SUN Machines 79S901SUN

Use with MIPS C-Compiler or with IDT/c. Developmental Use License number must be referenced on order, or must be ordered simultaneously.