LAUTERBACH
DEVELOPMENT TOOLS

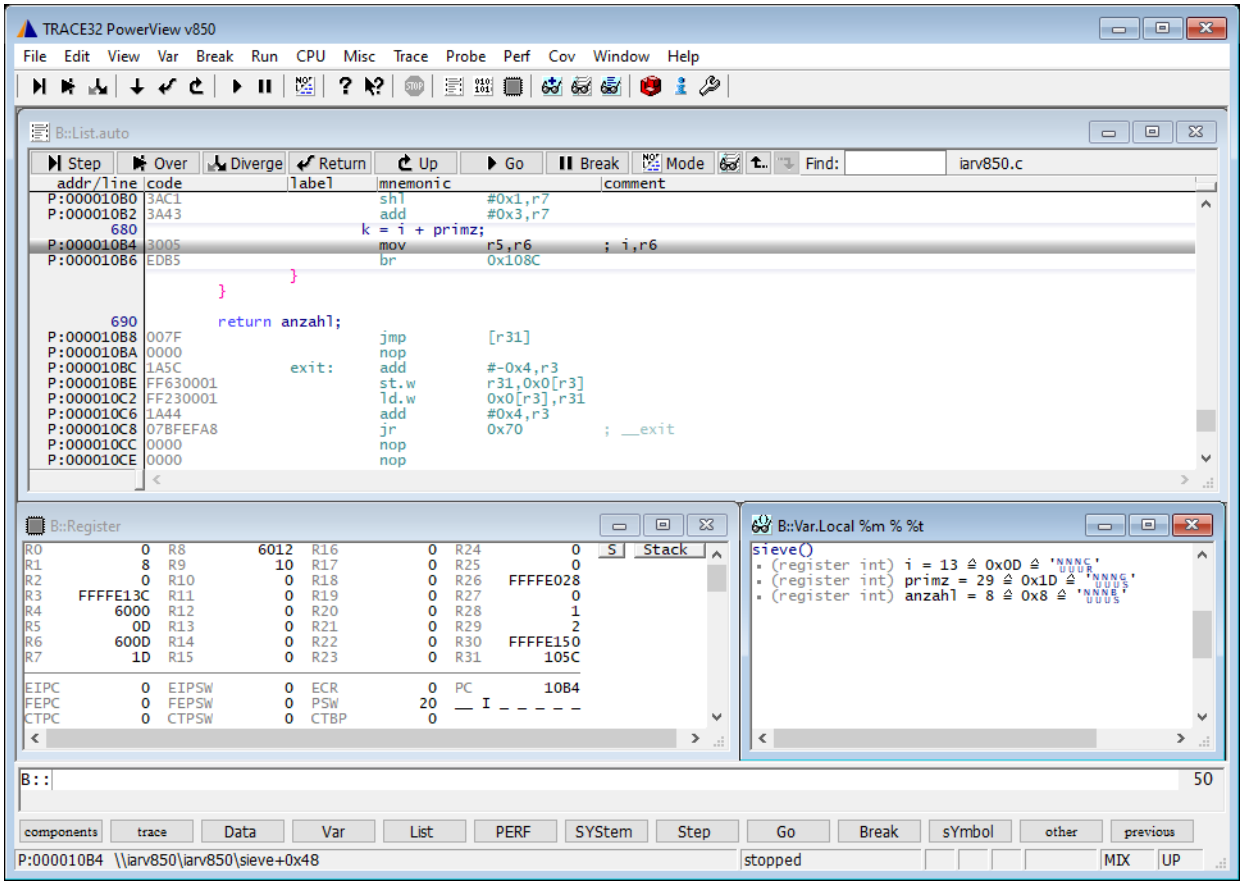
Simulator for V850

MANUAL

Simulator for V850

Simulator for V850

Version 06-Jun-2024



Introduction

This document describes the processor-specific settings and features for the TRACE32 Instruction Set Simulator for V850/RH850.

All general commands are described in the **"PowerView Command Reference"** (ide_ref.pdf) and **"General Commands Reference"**.

TRACE32 Simulator License

[build 68859 - DVD 02/2016]

The extensive use of the TRACE32 Instruction Set Simulator requires a *TRACE32 Simulator License*.

For more information, see **www.lauterbach.com/sim_license.html**.

Brief Overview of Documents for New Users

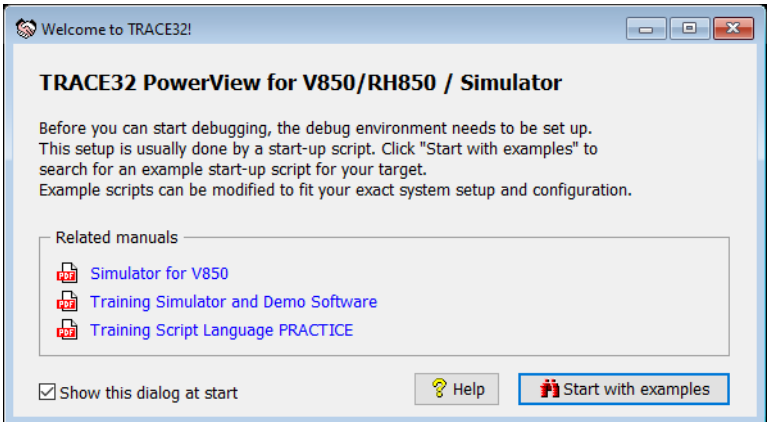
Architecture-independent information:

- **"Training Basic Debugging"** (training_debugger.pdf): Get familiar with the basic features of a TRACE32 debugger.
- **"T32Start"** (app_t32start.pdf): T32Start assists you in starting TRACE32 PowerView instances for different configurations of the debugger. T32Start is only available for Windows.
- **"General Commands"** (general_ref_*<x>*.pdf): Alphabetic list of debug commands.

Architecture-specific information:

- **"Processor Architecture Manuals"**: These manuals describe commands that are specific for the processor architecture supported by your debug cable. To access the manual for your processor architecture, proceed as follows:
  - Choose **Help** menu > **Processor Architecture Manual**.
- **"OS Awareness Manuals"** (rtos_*<os>*.pdf): TRACE32 PowerView can be extended for operating system-aware debugging. The appropriate OS Awareness manual informs you how to enable the OS-aware debugging.

To get started with the most important manuals, use the **Welcome to TRACE32!** dialog (**WELCOME.view**):



Demo and Start-up Scripts

To search for PRACTICE scripts, do one of the following in TRACE32 PowerView:

- Type at the command line: **WELCOME.SCRIPTS**
- or choose **File** menu > **Search for Script**.

  You can now search the demo folder and its subdirectories for PRACTICE start-up scripts (*.cmm) and other demo software.

You can also manually navigate in the ~~/demo/v850/ and ~/demo/rh850/ subfolders of the system directory of TRACE32.

Quick Start of the Simulator

To start the simulator, proceed as follows:

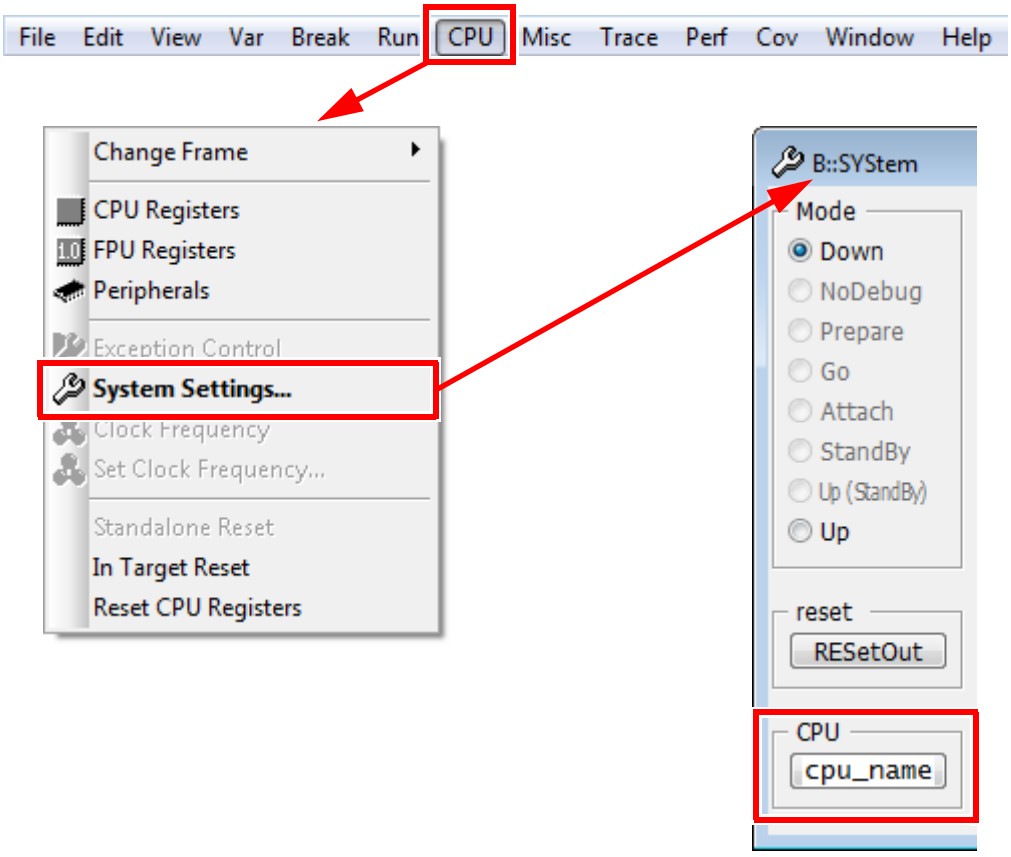
1. Select the device prompt for the Simulator and reset the system.

```
B::

RESet
```

The device prompt `B::` is normally already selected in the TRACE32 command line. If this is not the case, enter `B::` to set the correct device prompt. The **RESet** command is only necessary if you do not start directly after booting TRACE32.

2. Specify the CPU specific settings.




```
SYStem.CPU <cpu_name>
```

The default values of all other options are set in such a way that it should be possible to work without modification. Please consider that this is probably not the best configuration for your target.

3. Enter debug mode.

```
SYStem.Up
```

This command resets the CPU and enters debug mode. After this command is executed it is possible to access memory and registers.

4. Load the program.

```
Data.LOAD.<file_format> <file>      ; load program and symbols
```

See the **Data.LOAD** command reference for a list of supported file formats. If uncertain about the required format, try **Data.LOAD.auto**.

A detailed description of the **Data.LOAD** command and all available options is given in the reference guide.

5. Start-up example

A typical start sequence is shown below. This sequence can be written to a PRACTICE script file (*.cmm, ASCII format) and executed with the command **DO** *<file>*.

```
B::                                 ; Select the ICD device prompt

WinCLEAR                            ; Clear all windows

SYStem.CPU <cpu_name>               ; Select CPU type

SYStem.Up                           ; Reset the target and enter
                                    ; debug mode

Data.LOAD.<file_format> <file>      ; Load the application

Register.Set pc main                ; Set the PC to function main

PER.view                            ; Show clearly arranged
                                    ; peripherals in window      *)

List.Mix                            ; Open source code window    *)

Register.view /SpotLight            ; Open register window       *)

Frame.view /Locals /Caller          ; Open the stack frame with
                                    ; local variables            *)

Var.Watch %Spotlight flags ast      ; Open watch window for
                                    ; variables                  *)
```

*) These commands open windows on the screen. The window position can be specified with the **WinPOS** command.

For more information, see **“API for TRACE32 Instruction Set Simulator”** (simulator_api.pdf).

Troubleshooting

No information available.

FAQ

Please refer to https://support.lauterbach.com/kb.

SYStem.CPU — CPU type selection

Format: **SYStem.CPU** *<cpu>*

Selects the CPU type.

SYStem.LOCK — Lock and tristate the debug port

Format: **SYStem.LOCK** [**ON** | **OFF**]

The command has no effect for the simulator.

SYStem.MemAccess — Select run-time memory access method

Format: **SYStem.MemAccess Enable | StopAndGo | Denied**
SYStem.ACCESS (deprecated)

Enable	Memory access during program execution to target is enabled.
Denied	Memory access during program execution to target is disabled.
StopAndGo	Temporarily halts the core(s) to perform the memory access. Each stop takes some time depending on the speed of the JTAG port, the number of the assigned cores, and the operations that should be performed.

Format:	**SYStem.Mode** *<mode>*
	SYStem.Down **(alias for SYStem.Mode Down)**
	SYStem.Up **(alias for SYStem.Mode Up)**
<mode>:	**Down**
	NoDebug
	Go
	Up

Default: Down.

Selects the target operating mode.

Down	The CPU is in reset. Debug mode is not active. Default state and state after fatal errors.
NoDebug	The CPU is running. Debug mode is not active. Debug port is tristate. In this mode the target should behave as if the debugger is not connected.
Go	The CPU is running. Debug mode is active. After this command the CPU can be stopped with the break command or if any break condition occurs.
Up	The CPU is not in reset but halted. Debug mode is active. In this mode the CPU can be started and stopped. This is the most typical way to activate debugging.

If the mode **Go** is selected, this mode will be entered, but the control button in the **SYStem.state** window jumps to the mode **Up**.

<parameter>:	**DRPRE** *<bits>*
(JTAG):	**DRPOST** *<bits>*
	IRPRE *<bits>*
	IRPOST *<bits>*
	TAPState *<state>*
	TCKLevel *<level>*
	TriState [**ON** | **OFF**]
	Slave [**ON** | **OFF**]

The **SYStem.CONFIG** commands have no effect in Simulator. These commands describe the physical configuration at the JTAG port and the trace port of a multi-core hardware target. Since the simulator normally just simulates the instruction set, these commands will be ignored. Refer to the relevant **Processor Architecture Manual** in case you want to know the effect of these commands on a debugger.

SYStem.Option.IMASKASM Mask interrupts during assembler step

Format: **SYStem.Option.IMASKASM** [**ON** | **OFF**]

If enabled, the interrupt mask bits of the cpu will be set during assembler single-step operations. The interrupt routine is not executed during single-step operations. After single step the interrupt mask bits are restored to the value before the step.

SYStem.Option.IMASKHLL Mask interrupts during HLL step

Format: **SYStem.Option.IMASKHLL** [**ON** | **OFF**]

If enabled, the interrupt mask bits of the cpu will be set during HLL single-step operations. The interrupt routine is not executed during single-step operations. After single step the interrupt mask bits are restored to the value before the step.

The command asserts nRESET on the JTAG connector in the TRACE32 In-Circuit Debugger (ICD) but is ignored by the TRACE32 Instruction Set Simulator. However, the command is allowed in the simulator so that you can run scripts which have actually been made for the debugger. For more information about the effect in the debugger, refer to your **Processor Architecture Manual** (debugger_*<arch>*.pdf).

TrOnchip.state — Display on-chip trigger window

Format: **TrOnchip.state**

Opens the **TrOnchip.state** window.

TrOnchip.RESet — Set on-chip trigger to default state

Format: **TrOnchip.RESet**

Sets the TrOnchip settings and trigger module to the default settings.

Memory Classes

Overview

Access Class	Description
C	CPU (Program and Data)
D	Data
P	Program
ED	Dualport Data
EP	Dualport Program

C:, E:, D:, P:, ED:, EP:

C:, P: and D:

This storage classes operate on the same physical memory. They are only used to be compatible with other emulation probes.

E:, EP: and ED:

The E: prefix is used for accesses via dualport. The on-chip I/O-registers and the on-chip RAM area can be accessed via a special dualport mode (see MAP.SOnchip).

Keywords for the Display

INT	Occurrence of Interrupt
psw_EP	Processor Status Word: EP bit set
psw_ID	Processor Status Word: ID bit set
psw_NP	Processor Status Word: NP bit set
psw_SAT	Processor Status Word: SAT bit set
praddress	Instruction address
prdata	Instruction data
prcycle	Instruction cycle type