WIND RIVER

Wind River® Simics® MIL-STD-1553

TECHNOLOGY GUIDE

4.6

Revision 4081
Date 2012-11-16

Corporate Headquarters
Wind River
500 Wind River Way
Alameda, CA 94501-1153
U.S.A.

Toll free (U.S.A.): 800-545-WIND
Telephone: 510-748-4100
Facsimile: 510-749-2010

For additional contact information, see the Wind River Web site:
`www.windriver.com`
For information on how to contact Customer Support, see:
`www.windriver.com/support`

Contents

Chapter 1

Overview

1.1 Introduction

MIL-STD-1553 is a military standard for a serial data bus. It runs at 1MHz, and has one *bus controller* (BC) and up to 31 *remote terminals* (RTs) connected. There are also optional *bus monitor* (BM) devices. The protocol used on the bus is not described in this document, but there are several on-line resources on the net that covers 1553 in more detail.

1.2 1553 in Simics

The base of the MIL-STD-1553 support is Simics is a 1553 bus model, called **ms1553-link**, and its API. There is an example bus controller, written in DML with source code available, that models the *SμMMIT*-based *PMC1553* device from Alphi Technology. There is also a parameterized remote terminal dummy device that can be used for testing.

1.3 Implementation

The API to the 1553 link is implemented on a protocol level of abstraction and does not model the synchronization and parity bits of a real bus. However, it is still possible to model errors in the Manchester encoding and parity, for example, by injecting errors explicitly. The 1553 link in Simics keeps track of the protocol state, and warns when any device breaks the specification. The link also supports inspection of the data sent on it. The full 1553 link API is described in section 3.1, *Interfaces*.

1.4 Connecting to a Real 1553 Bus

A bus controller model can be connected to a real 1553 bus using the **host-pmc-1553** class or the **host-condor-pci-1553** class. The **host-pmc-1553** class supports any simulated BC that implements the standard Simics 1553 interface, and requires a real PMC1553 device on the host including drivers from Alphi Technology. The source code for this host connection is available, and can used as example when interfacing other bus controllers to implement a host bus connection. The **host-condor-pci-1553** class works as a bridge between RT models

in Simics and an actual Condor PCI 1553 card on the host. The card emulates RTs which also are represented as models in Simics.

Chapter 2

Simulation Models

Simics implements the following MIL-STD-1553 related classes. For more detailed information about them, refer to chapter 4.

2.1 PMC1553 BC

PMC1553 is a model of the SμMMIT-based MIL-STD-1553 module from Alphi Technology.

2.2 MS1553 Link/Bus

The **ms1553-link** models a MIL-STD-1553 bus, called link in Simics. It provides interfaces to bus controllers, remote terminal devices and bus monitors. The link keeps track of the protocol state to aid debugging of 1553 device and driver development. It also supports error injection and data inspection.

2.3 MS1553 RT

This **ms1553_rt** is an example model of a remote terminal (RT) according to the MIL-STD-1553 standard. The RT can either connect to a virtual ms1553-bus or real ms1553-bus. The *link* connector connects the RT to a virtual bus. The *bridge* connector connects the RT to a real bus via a *bridge*. The *bridge* controls a host card that emulates the RT. The emulated RT is referred to as the shadow RT, as it is identical to the RT in the Simics world. It is up to the RT in Simics to keep the shadow RT in sync with itself.

2.4 MS1553 Test RT

Simple MIL-STD-1553 Remote Terminal used for testing. This RT can be configured with a number of sub-addresses using the *sub_addresses* attribute. For each sub-address, a static list of 16-bit words should be supplied that specifies what the RT will return on “Transmit” requests. The data may be 1 up to 32 words in length. For “Receive” requests, the RT will report the received words using Simics standard log functions.

2.5 Condor PCI 1553 Host Connection

The **host-condor-pci-1553** class is a layer between RTs in Simics and a host Condor PCI 1553 card. The RTs in Simics are emulated by the host card. It is not possible for the RTs in Simics to control message transfers directly in real time. This is due to hard timing restrictions for the real MIL-STD-1553 bus connected to the host card. The emulated RTs (shadow RTs) are set up by the RTs in Simics to make them respond to messages as they would have done. The RTs in Simics get a copy of the messages handled by the shadow RTs. The **host-condor-pci-1553** class sets up the host card when new instance of the class is created. It creates a emulated RT when a Simics RT connect to it. The class includes the possibility to test RTs in Simics by emulating a simple bus controller. The bus controller is setup using the *bc_test* attribute. This class requires the Condor BusTools 1553-API to be installed on the host to work. Only one instance of this component is allowed in a simulation session.

2.6 PMC1553 Host Controller Connection

The host-pmc-1553 provides a connection from a simulated 1553 Bus Controller to a real Alphi PMC-1553 device, allowing the simulated controller to access devices on a real MIL-STD-1553 bus. The host-pmc-1553 object should be connected directly to the simulated 1553 bus controller device in place of of the ms1553-link.
To compile the host-pmc-1553 module, a driver development kit from Alphi Technology is required. The Alphi Technology PMC 1553 driver, needed to use this module, supports Windows 2000 and has also been tested on Windows Server 2003 SE.

Chapter 3

1553 API

Below is a description of the 1553 interfaces used by devices in Simics. For documentation of the complete Simics API, refer to the *Simics Reference Manual*.

3.1 Interfaces

3.1.1 ms1553_link

Implemented By
host-pmc-1553, ms1553-link

Description

Execution Context
Instruction Context for all methods.

3.1.2 ms1553_terminal

Implemented By
PMC1553, ms1553-recorder-bm, ms1553-test-rt, ms1553_rt

Description

Execution Context
Instruction Context for all methods.

3.1.3 ms1553_bridge_terminal

Implemented By
ms1553_rt

Description

Execution Context
Instruction Context for all methods.

3.1.4 ms1553_bridge_bus

Implemented By
host-condor-pci-1553

Description

Execution Context
Instruction Context for all methods.

3.2 Types

3.2.1 ms1553_phase_t

```
typedef enum {
        MS1553_Phase_Idle,
        MS1553_Phase_T_Command,
        MS1553_Phase_R_Command,
        MS1553_Phase_M_T_Command,
        MS1553_Phase_M_R_Command,
        MS1553_Phase_M_N_Command,
        MS1553_Phase_Data,
        MS1553_Phase_Status,
        MS1553_Num_Phases
} ms1553_phase_t;
```

3.2.2 ms1553_error_t

```
typedef enum {
        MS1553_Err_Manchester,
        MS1553_Err_Sync_Field,
        MS1553_Err_Word_Count,
        MS1553_Err_Parity,
        MS1553_Err_Protocol
} ms1553_error_t;
```

3.2.3 ms1553_words_t

```
typedef struct {
        int length;
        uint16 *data;
} ms1553_words_t;
```

3.2.4 ms1553_mode_code_t

```
typedef enum {
        MS1553_Dynamic_Bus_Control = 0,
```

```
        MS1553_Synchronize_T = 1,
        MS1553_Transmit_Status_word = 2,
        MS1553_Initiate_Self_Test = 3,
        MS1553_Transmitter_Shutdown = 4,
        MS1553_Override_Transmitter_Shutdown = 5,
        MS1553_Inhibit_Terminal_Flag_Bit = 6,
        MS1553_Override_Inhibit_Terminal_Flag_Bit = 7,
        MS1553_Reset_Remote_Terminal = 8,
        MS1553_Transmit_Vector_Word = 16,
        MS1553_Synchronize_R = 17,
        MS1553_Transmit_Last_Command_Word = 18,
        MS1553_Transmit_BIT_Vector = 19,
        MS1553_Selected_Transmitter_Shutdown = 20,
        MS1553_Override_Selected_Transmitter_Shutdown = 21
} ms1553_mode_code_t;
```

3.2.5 ms1553_dir_t

```
typedef enum {
        MS1553_Dir_Transmit = 0,
        MS1553_Dir_Receive  = 1
} ms1553_dir_t;
```

3.2.6 ms1553_shadow_word_t

```
typedef enum {
        MS1553_Shadow_Word_Command = 0,
        MS1553_Shadow_Word_Status  = 1,
        MS1553_Shadow_Word_BIT     = 2,
        MS1553_Shadow_Word_Vector  = 3
} ms1553_shadow_word_t;
```

Chapter 4

Class Details

Following is a list of all component and configuration classes used to implement 1553 in Simics. For documentation of other classes, refer to the *Model Builder Reference Manual*.

4.1 Components

4.1.1 pci-pmc1553

Provided By
mil-std-1553-components

Interfaces Implemented
component, component_connector, conf_object, log_object

Description
The “pci-pmc1553” component represents a PMC-1553PCI based MIL-STD-1553 Bus Controller.

Connectors

Name	Type	Direction
pci-bus	pci-bus	up
link-a	ms1553-link	down
link-b	ms1553-link	down

Attributes

instantiated
Optional attribute; **read/write** access; type: `boolean`. Set to TRUE if the component has been instantiated.

object_prefix
Optional attribute; **read/write** access; type: `string` or `nil`. Object prefix string used by the component. The prefix is typically set by the **set-component-prefix** command before the component is created.

top_component
Optional attribute; **read/write** access; type: `object` or `nil`. The top level component. Attribute is not valid until the component has been instantiated.

Class Attributes

basename
Pseudo class attribute; **read-only** access; type: `string`. The basename of the component.

component_icon
Pseudo class attribute; **read-only** access; type: `string` or `nil`. Name of a 24x24 pixels large icon in PNG format used to graphically represent the component in a configuration viewer.

top_level
Pseudo class attribute; **read-only** access; type: `boolean` or `nil`. Set to TRUE for top-level components, i.e. the root of a hierarchy.

Command List

Commands defined by interface conf_object
break-hap, get-attribute-list, get-interface-list, get-interface-port-list, list-attributes, list-interfaces, log, log-group, log-level, log-size, log-type, trace-hap, unbreak-hap, untrace-hap, wait-for-log

Commands

info	print information about the device
status	print status of the device

Command Descriptions

<pci-pmc1553>.info

Synopsis
<pci-pmc1553>.info

Description
Print detailed information about the configuration of the device.

<pci-pmc1553>.status

Synopsis
<pci-pmc1553>.status

Description
Print detailed information about the current status of the device.

4.1.2 std-ms1553-link

Provided By
mil-std-1553-components

Interfaces Implemented
component, component_connector, conf_object, log_object

Description
The "std-ms1553-link" component represents a MIL-STD-1553 bus/link.

Connectors

Name	Type	Direction
device	ms1553-link	any

Attributes

instantiated
Optional attribute; **read/write** access; type: `boolean`. Set to TRUE if the component has been instantiated.

object_prefix
Optional attribute; **read/write** access; type: `string` or `nil`. Object prefix string used by the component. The prefix is typically set by the **set-component-prefix** command before the component is created.

top_component
Optional attribute; **read/write** access; type: `object` or `nil`. The top level component. Attribute is not valid until the component has been instantiated.

Class Attributes

basename
Pseudo class attribute; **read-only** access; type: `string`. The basename of the component.

component_icon
Pseudo class attribute; **read-only** access; type: `string` or `nil`. Name of a 24x24 pixels large icon in PNG format used to graphically represent the component in a configuration viewer.

top_level
Pseudo class attribute; **read-only** access; type: `boolean` or `nil`. Set to TRUE for top-level components, i.e. the root of a hierarchy.

Command List

Commands defined by interface conf_object
break-hap, get-attribute-list, get-interface-list, get-interface-port-list, list-attributes, list-interfaces, log, log-group, log-level, log-size, log-type, trace-hap, unbreak-hap, untrace-hap, wait-for-log

Commands

info	print information about the device
status	print status of the device

Command Descriptions

<std-ms1553-link>.info

Synopsis
<std-ms1553-link>.info

Description
Print detailed information about the configuration of the device.

<std-ms1553-link>.status

Synopsis
<std-ms1553-link>.status

Description
Print detailed information about the current status of the device.

4.2 Classes

4.2.1 PMC1553

Provided By
PMC1553

Interfaces Implemented
conf_object, io_memory, log_object, ms1553_terminal, pci_device

Ports
HRESET (signal), SRESET (signal), bc (int_register, io_memory), pci_config (int_register, io_memory), pmc (int_register, io_memory), rt (int_register, io_memory), summit (int_register, io_memory)

Description
PMC1553 is a model of the SμMMIT-based MIL-STD-1553 module from Alphi Technology.

Attributes

AB_STD
Optional attribute; **read/write** access; type: `integer`. Select the standard, 0 for 1553B and 1 for 1553A

AUTOEN
Optional attribute; **read/write** access; type: `boolean`. Enable auto-initialization

EXT_TCLK
Optional attribute; **read/write** access; type: `integer`. Frequency in Hz of external TCLK

LOCK
Optional attribute; **read/write** access; type: `boolean`. Assert LOCK to prevent modification of status word

MSEL
Optional attribute; **read/write** access; type: `integer`. SμMMIT Mode Selection: 0 for BC and 1 for RT.

RTA
Optional attribute; **read/write** access; type: `integer`. Remote Terminal Address

SSYSF
Optional attribute; **read/write** access; type: `boolean`. Subsystem Flag

cmd_info
Optional attribute; **read/write** access; type: `dictionary`. Current command information

config_registers
Pseudo attribute; **read-only** access; type: `[i*]`. The PCI configuration registers, each 32 bits in size.

expansion_rom_size
Optional attribute; **read/write** access; type: `integer`. The size of the expansion ROM mapping.

link
Optional attribute; **read/write** access; type: `[o|[os]|n{2}]`. The MIL-STD-1553 bus/link that the device is connected to.

pci_bus
Required attribute; **read/write** access; type: `[os]` or `object`. The PCI bus this device is connected to, implementing the `pci-bus` interface.

pci_config_command
Optional attribute; **read/write** access; type: `integer`. The PCI command register.

pci_config_device_id
Optional attribute; **read/write** access; type: `integer`. The Device ID of the PCI device

pci_config_vendor_id
Optional attribute; **read/write** access; type: `integer`. The Vendor ID of the PCI device

rom_image
Optional attribute; **read/write** access; type: `[os]`, `object`, or `nil`. Image holding the contents for auto-initialization, the required size of bytes is

- 1088 for RT mode,
- 64 for MT mode,
- 64 + [N x 16] for BC mode, N: number of command block

sram
Required attribute; **read/write** access; type: `[os]` or `object`. Object for dual ported SRAM. The image of this object must be the object specified in the sram_image attribute.

sram_image
Required attribute; **read/write** access; type: `[os]` or `object`. Image holding SRAM contents. The size must be 128KiB. This object must be the same as the image for the object specified in the sram attribute.

Command List

Commands defined by interface conf_object
break-hap, get-attribute-list, get-interface-list, get-interface-port-list, list-attributes, list-interfaces, log, log-group, log-level, log-size, log-type, trace-hap, unbreak-hap, untrace-hap, wait-for-log

Commands

hard-reset	hard reset
info	print information about the device
pci-header	*deprecated* — print PCI device header
print-pci-config-regs	print PCI configuration registers
soft-reset	soft reset
status	print status of the device

Command Descriptions

<PMC1553>.hard-reset

Synopsis
<PMC1553>.hard-reset

Description
hard reset pmc1553 device

<PMC1553>.info

Synopsis
<PMC1553>.info

Description
Print detailed information about the configuration of the device.

<PMC1553>.pci-header — *deprecated*

Synopsis
<PMC1553>.pci-header [-v]

Description
This command is deprecated; use <PMC1553>.print-pci-config-regs instead.

<PMC1553>.print-pci-config-regs

Synopsis
<PMC1553>.print-pci-config-regs [-v]

Description
Print the PCI configuration registers. The *-v* flag turns on verbose mode.

<PMC1553>.soft-reset

Synopsis
<PMC1553>.soft-reset

Description
soft reset pmc1553 device

<PMC1553>.status

Synopsis
<PMC1553>.status

Description
Print detailed information about the current status of the device.

4.2.2 ms1553-link

Provided By
ms1553-link

Interfaces Implemented
conf_object, log_object, ms1553_link

Description
The **ms1553-link** models a MIL-STD-1553 bus, called link in Simics. It provides interfaces to bus controllers, remote terminal devices and bus monitors. The link keeps track of the protocol state to aid debugging of 1553 device and driver development. It also supports error injection and data inspection.

Command List

Commands defined by interface conf_object
break-hap, get-attribute-list, get-interface-list, get-interface-port-list, list-attributes, list-interfaces, log, log-group, log-level, log-size, log-type, trace-hap, unbreak-hap, untrace-hap, wait-for-log

Commands

capture-start	start traffic recorder
capture-stop	stop traffic recorder
info	print information about the device
playback-start	start traffic generator
playback-stop	stop traffic generation
status	print status of the device

Command Descriptions

<ms1553-link>.capture-start

Synopsis
<**ms1553-link**>**.capture-start** *filename*

Description
Starts recording bus traffic to a specified file, in a format that can be played back. If the file exist, it is silently overwritten.

See Also
<ms1553-link>.capture-stop

<ms1553-link>.capture-stop

Synopsis
<**ms1553-link**>**.capture-stop**

Description
Stop recording bus traffic previously started with start-recorder.

See Also
<ms1553-link>.playback-start

<**ms1553-link**>**.info**

Synopsis
<**ms1553-link**>**.info**

Description
Print detailed information about the configuration of the device.

<**ms1553-link**>**.playback-start**

Synopsis
<**ms1553-link**>**.playback-start** *filename*

Description
Starts generating bus traffic from a specified file. The file should be of PASS-3200 dump format.

See Also
<ms1553-link>.capture-start, <ms1553-link>.playback-stop

<**ms1553-link**>**.playback-stop**

Synopsis
<**ms1553-link**>**.playback-stop**

Description
Stop bus traffic generation previously started with playback-start.

See Also
<ms1553-link>.playback-start

<**ms1553-link**>**.status**

Synopsis
<**ms1553-link**>**.status**

Description
Print detailed information about the current status of the device.

4.2.3 ms1553_rt

Provided By
ms1553-rt

Interfaces Implemented
conf_object, log_object, ms1553_bridge_terminal, ms1553_terminal

Description
This **ms1553_rt** is an example model of a remote terminal (RT) according to the MIL-STD-1553 standard. The RT can either connect to a virtual ms1553-bus or real ms1553-bus. The *link* connector connects the RT to a virtual bus. The *bridge* connector connects the RT to a real bus via a *bridge*. The *bridge* controls a host card that emulates the RT. The emulated RT is referred to as the shadow RT, as it is identical to the RT in the Simics world. It is up to the RT in Simics to keep the shadow RT in sync with itself.

Attributes

disconnect_bridge
Pseudo attribute; **read/write** access; type: `integer`. Disconnect from bridge.

last_command
Optional attribute; **read/write** access; type: `integer`. The last command word.

status_word
Optional attribute; **read/write** access; type: `integer`. The status word of the RT.

sub_addresses
Required attribute; **read/write** access; type: `[[i[i*][i*]]*]`. The Terminal Address of the RT.

terminal_address
Required attribute; **read/write** access; type: `integer`. The terminal address of the RT.

Command List

Commands defined by interface conf_object
break-hap, get-attribute-list, get-interface-list, get-interface-port-list, list-attributes, list-interfaces, log, log-group, log-level, log-size, log-type, trace-hap, unbreak-hap, untrace-hap, wait-for-log

4.2.4 ms1553-test-rt

Provided By
ms1553-test-rt

Interfaces Implemented
conf_object, log_object, ms1553_terminal

Description

Simple MIL-STD-1553 Remote Terminal used for testing. This RT can be configured with a number of sub-addresses using the *sub_addresses* attribute. For each sub-address, a static list of 16-bit words should be supplied that specifies what the RT will return on "Transmit" requests. The data may be 1 up to 32 words in length. For "Receive" requests, the RT will report the received words using Simics standard log functions.

Attributes

sub_addresses

Required attribute; **read/write** access; type: `[[i[i*]]*]`. A list of sub-addresses that the RT will implement, and for each sub-address there is a non-empty list of up to 32 16-bit words that are returned on reads from that sub-address.

terminal_address

Required attribute; **read/write** access; type: `integer`. The Terminal Address of the RT.

Command List

Commands defined by interface conf_object

break-hap, get-attribute-list, get-interface-list, get-interface-port-list, list-attributes, list-interfaces, log, log-group, log-level, log-size, log-type, trace-hap, unbreak-hap, untrace-hap, wait-for-log

4.2.5 host-condor-pci-1553

Provided By

host-condor-pci-1553

Interfaces Implemented

conf_object, log_object, ms1553_bridge_bus

Description

The **host-condor-pci-1553** class is a layer between RTs in Simics and a host Condor PCI 1553 card. The RTs in Simics are emulated by the host card. It is not possible for the RTs in Simics to control message transfers directly in real time. This is due to hard timing restrictions for the real MIL-STD-1553 bus connected to the host card. The emulated RTs (shadow RTs) are set up by the RTs in Simics to make them respond to messages as they would have done. The RTs in Simics get a copy of the messages handled by the shadow RTs. The **host-condor-pci-1553** class sets up the host card when new instance of the class is created. It creates a emulated RT when a Simics RT connect to it. The class includes the possibility to test RTs in Simics by emulating a simple bus controller. The bus controller is setup using the *bc_test* attribute. This class requires the Condor BusTools 1553-API to be installed on the host to work. Only one instance of this component is allowed in a simulation session.

Command List

Commands defined by interface conf_object

break-hap, get-attribute-list, get-interface-list, get-interface-port-list, list-attributes, list-interfaces, log, log-group, log-level, log-size, log-type, trace-hap, unbreak-hap, untrace-hap, wait-for-log

4.2.6 host-pmc-1553

Provided By

host-pmc-1553

Interfaces Implemented

conf_object, log_object, ms1553_link

Description

The host-pmc-1553 provides a connection from a simulated 1553 Bus Controller to a real Alphi PMC-1553 device, allowing the simulated controller to access devices on a real MIL-STD-1553 bus. The host-pmc-1553 object should be connected directly to the simulated 1553 bus controller device in place of of the ms1553-link.

To compile the host-pmc-1553 module, a driver development kit from Alphi Technology is required. The Alphi Technology PMC 1553 driver, needed to use this module, supports Windows 2000 and has also been tested on Windows Server 2003 SE.

Command List

Commands defined by interface conf_object

break-hap, get-attribute-list, get-interface-list, get-interface-port-list, list-attributes, list-interfaces, log, log-group, log-level, log-size, log-type, trace-hap, unbreak-hap, untrace-hap, wait-for-log

Index

R

S

T