

Author: Sven Dornbusch dornbusch@mpifr-bonn.mpg.de, Armin Felke, felke@mpifr-bonn.mpg.de

Legend of notation:

[A] → expression A is optional

(A|B) → either expression A or expression B

<x> → a placeholder for a value named "x"

version

Displays the version of the firmware (distinguishing between hardware and software version)

Arguments: –

sysstat

Displays information about the current status of the system and gives an overview of the state of the most important user settings.

Arguments: –

vsi_bitmask

Gets or sets the VSI input bitmask in order to select the channels to be processed

Arguments: [[[...[<bitmask-7>] ... <bitmask-3>] <bitmask-2>] <bitmask-1>|reset]
<bitmask-X> (optional): a 32 bit bitmask each, that specifies the active bits of VSI-X.

Omitted bitmasks are assumed to be zero (0x00000000). The only rule for the values is that in total at least one bit must be active.

All arguments are optional. If the command is called without arguments the current VSI bitmask is displayed.

The eight - up to 32 bit wide - bitmasks specify which bits of a VSI input stream are active and being processed, the others are discarded. This effectively reduces the data amount.

If the command is called with the "reset" argument, all of the bitmasks are reset to their default value (0xffffffff = all active).

In general, the number of active bits in the bitmask should be a power of two (1,2,4,8,16,32,64,128).

If the number of active bits is not a power of two, the highest active bit is replicated in the resulting data word until the the next power of two is reached. This special case is displayed as a warning by the command.

Note: The bitmask is displayed as "(invalid)" if some custom settings (e.g. a custom channel permutation) would lead to an impossible bitmask setup.

time

Displays the current time of the active 1PPS source.

Arguments: –

The displayed time is the (synchronized) VDIF time in UTC format.

timesync

Performs time synchronization to the active 1PPS source.

Arguments: [<YYYY>-<MM>-<DD>T<hh>:<mm>:<ss>[(+|-)<ZZ:ZZ>]]
<YYYY>: the current year in four digit representation (range: 2000..2130).
<MM>: the current month (range: 1..12)
<DD>: the current day (range: 1..31)
<hh>: the current hour (range: 0..23)
<mm>: the current minute (range: 1..59)
<ss>: the current second (range: 1..59)
(+|-)<ZZ:ZZ>: the time zone offset (only full hours are supported)
the offset is optional. if omitted offset 00:00 (=GMT) is the default.
(Example time: 2013-07-09T15:41:33+01:00)

The current UTC can be passed as an argument. If it is omitted, the GPS time will be taken for time synchronisation. For this to work a GPS device must be installed in the system.

A valid 1PPS signal is required for the time synchronization to succeed. If successful, the VDIF time was synchronized by this command.

vdif_leapsecs

Gets/sets the number of UTC leap seconds since VDIF reference epoch.

Arguments: [<leap seconds>]
<leap seconds>: the number of additional leap seconds since VDIF reference epoch (optional)

Generally, the number of leap seconds since the start of the VDIF reference epoch is zero, since UTC leap seconds are inserted at the end of a half-year and the reference epoch is set to the beginning of the current half-year.

This command exists for the case in which a UTC leap second needs to be inserted between the beginning of the reference epoch and the current time. With this command the user can manually insert any number of missing leap seconds. Furthermore, a negative leap second number allows the removal of seconds relative to the start of epoch.

vdif_station

Gets/sets the VDIF station ID.

Arguments: [<VDIF station ID>]
<VDIF station ID> (optional): a two character VDIF station ID (cmp. VDIF standard)

Note: setting this value directly affects the header data of the VDIF data format.

vdif_frame

Gets or sets the properties of VDIF frames.

Arguments: [<channel bit width> <#channels per frame> [<payload size in bytes>]

<channel bit width> (optional): the size of each channel in bits (allowed values: 1,2,4,8,16,32,64)

<#channels per frame> (optional): number of channels per VDIF frame (allowed values: 1,2,4,8,16,32,64,128)

<payload size in bytes> (optional): the total payload size (= frame size without header data) of the VDIF frames

All arguments are optional. If the command is called without arguments, the properties are not modified but only displayed.

If successful, the command displays the resulting number of frames per second and the number of data threads, according to the currently selected input. A warning is displayed instead if the VDIF frame properties do not match the currently selected input. The command fails if the desired frame setup is not supported (the frame setup is not changed in this case).

Note: after having set the VDIF frame properties and when processing has already been started, a reset of the datapath (i.e. "reset" or "reset keepsync") is mandatory.

Following rules apply to the properties:

1. The width of a data sample in a frame (<sample bit width>) equals to
 <#channels per frame> * <channel bit width> bits.
2. <sample bit width> must be 128 or less
3. If one of the properties is set and <payload size in bytes> is omitted, the optimal payload size is computed for the currently selected input.
4. Setting an "illegal" payload size is allowed but with the following restrictions:
 - a) If a <payload size in bytes> is given that exceeds the maximum supported payload size for this configuration, the payload size is automatically capped to the maximum.
 - b) The payload size is ensured to be a multiple of 8 bytes by truncation of the remaining bytes.
 - c) If <payload size in bytes> will result in the last frame of a second interval to be incomplete, this frame is silently discarded.

For more information about VDIF frame formats refer to the VDIF standard.

Note: a channel bit width of 64 bit simple data is not directly supported by the VDIF standard, such that in this case complex data is assumed and the complex flag is asserted in the header.

Note: this command exists for user convenience. As an alternative to this command, all of the VDIF frame properties can be directly set by editing FiLa10G's VDIF header registers.

vdif_userdata

Gets or sets the current user data fields that are embedded in the VDIF frame header.

Arguments: [<d0> [<d1> [<d2> [<d3>]]]]

<d0> (optional): the value of the first user data field (unsigned integer in hex or decimal format)

<d1> (optional): the value of the second user data field (unsigned integer in hex or decimal format)

<d2> (optional): the value of the third user data field (unsigned integer in hex or decimal format)

<d3> (optional): the value of the fourth user data field (unsigned integer in hex or decimal format)

All arguments are optional. If the command is called without arguments, the values of the user data fields are not modified but only displayed.

For each argument that is specified, the given value will be written to the respective user data field.

Note: setting a user data field by the help of this command is equivalent to writing directly into the respective `vdif_header` registers, but more comfortable.

destination

Gets/sets the output destination.

Arguments: <output index> [(<IPv4 address>[:<port>])|none]

<output index>: the index of the output for which the destination is to be get or set.

allowed values: 0, or 1

<IPv4 address>(optional): IP address, format x.x.x.x

<port>(optional): IP port number

Execute the command in the form "destination <output index>" to get a list of all current destination settings for the respective output.

Executing the command in the form "destination <output index> <IPv4 address>:<port>" sets the destination for the respective output. All frames coming from this output are sent to the specified destination. This overwrites all other destination settings previously made for this output.

Executing the command in the form "destination <output index> none" effectively disables the respective output and no frames will be sent at all. This overwrites all other destination settings previously made for this output.

tengbinfo

Retrieves the current parameters of a 10Gb Ethernet device.

Arguments: <device name>

<device name>: "eth0", "eth1", "eth2" or "eth3"

tengbcfg

Sets the parameters of a 10Gb Ethernet device.

Arguments: <device name> <tengbcfg parameters>

<device name>: "eth0", "eth1", "eth2" or "eth3"

<tengbcfg parameters>: the parameters to be set, example would be:

tengbcfg eth0 ip=192.168.1.28 gateway=192.168.1.1 nm=24

for setting ip-address, gateway and netmask for eth0

See TenGBE core documentation for further information.

tengbarp

Sets one ARP entry in a 10Gb Ethernet device.

Arguments: <device name> <ARP table index> <MAC address>

<device name>: "eth0", "eth1", "eth2" or "eth3"

<ARP table index>: index of ARP table entry to be modified

<MAC address>: MAC address to be set

arp

Enables (on) or disables (off) ARP queries on both Ethernet cores (eth0 and eth1).

Arguments: [(on|off)]

If arguments are omitted, the current arp status is displayed on the console.

start

Starts/restarts sending of VDIF output data.

Arguments vdif [force]

This command requires the respective timer to be synchronized (see "timesync" command).

For fast testing: append argument "force" to the command to automatically synchronize timer to "zero" time (= "2000-01-01T00:00:00"). Provided that a valid 1PPS signal is available, this will always be successful.

stop

Stops sending of output data (the opposite of "start").

Arguments: –

tick

Enables continuous 1PPS tick display on the console.

Arguments: –

WARNING: This command can only be called from a direct serial connection, and not by the control software!

The VDIF time must be synchronized before using this command.

When enabled the current time is displayed every second (synchronized with 1PPS source). Press <Space> or <Enter> key (or almost any other key) to leave the tick mode.

By pressing “+” or “-” the time can be adjusted by +1s or -1s, respectively. The total adjustment range is -60 to +60 seconds.

reset

Resets Core3H's datapath and erases synchronized time.

Arguments: [keepsync]

If "reset" is called without arguments, the complete datapath is reset and time synchronization is lost.

If "reset" is called with argument "keepsync", the Core3H tries to maintain the current time synchronization. For this to work the input stage of the data path and the timers are not reset. Time synchronization will not be correct anymore in the rare but possible case that a data sample with 1PPS flag is lost during the reset process.

reboot

Reboots the system.

Arguments: –

The Core3H's hardware and software is reset to its initial state, i.e. as it was directly after the programming of the FPGA, and lets the Core3H system boot again.

Warning: all previously configured settings and states are lost when rebooting!

core3_init

Initializes FiLa10G for the CORE3 system.

Arguments: –

core3_bstat

Displays the CORE3 bit statistics for the output of the selected filter.

Arguments: (0|1)

core3_power

Displays the power statistics of the filters

Arguments: –

core3_corr

Displays the correlation results between samplers 0-1, 1-2, 2-3

Arguments: --

These values are used to calibrate and validate the correct phase relation between the samplers of the ADB3L board.

sampler_offset

Displays the offset statistics for each of the four samplers of the ADB3L sampler board.

Arguments: --

These values are used to calibrate and validate the offset of the four samplers of the ADB3L sampler board.

sampler_power

Displays the power levels for each of the four samplers of the ADB3L sampler board.

Arguments: --

These values are used to calibrate and validate the gain of the four samplers of the ADB3L sampler board.