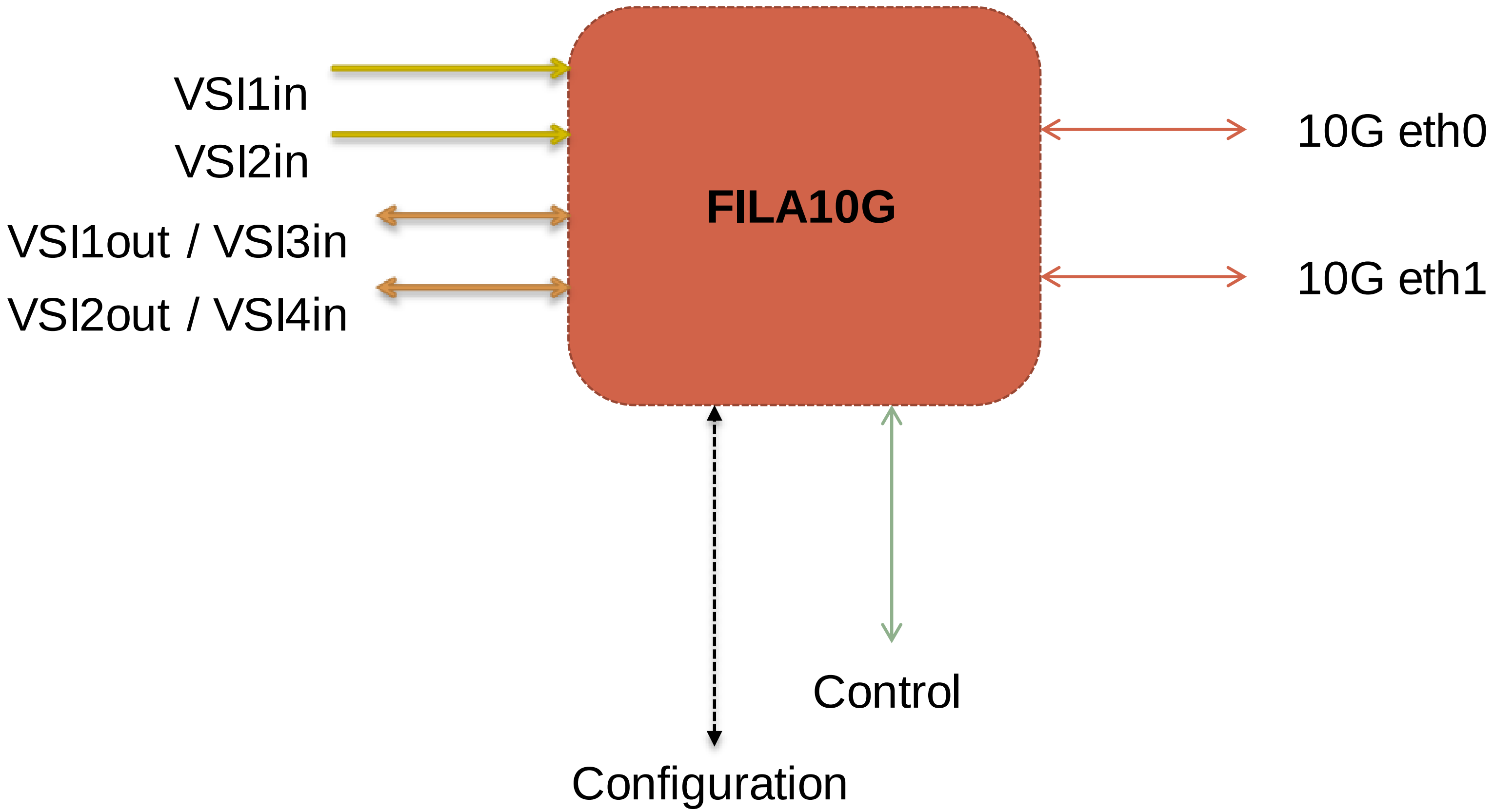
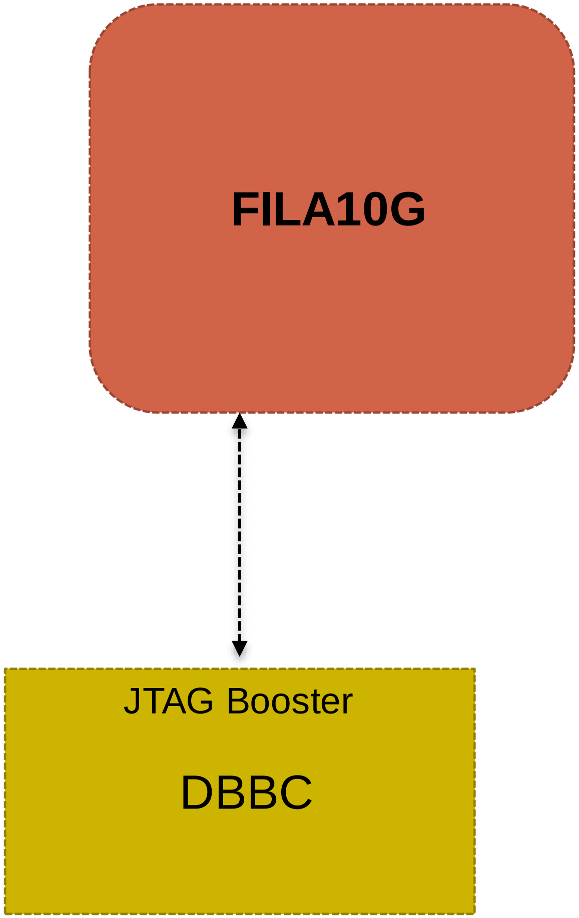
FILA10G – Double Port 10G VLBI-Ethernet Interface

How to use it

What to connect



How to configure



FILA10G

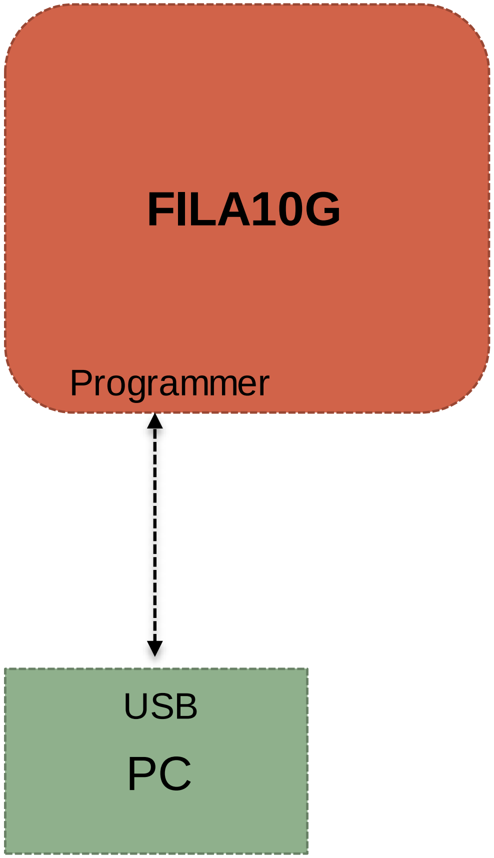
JTAG Booster

DBBC

if the DBBC is nearby

no multiple Xilinx programmers to be handled!

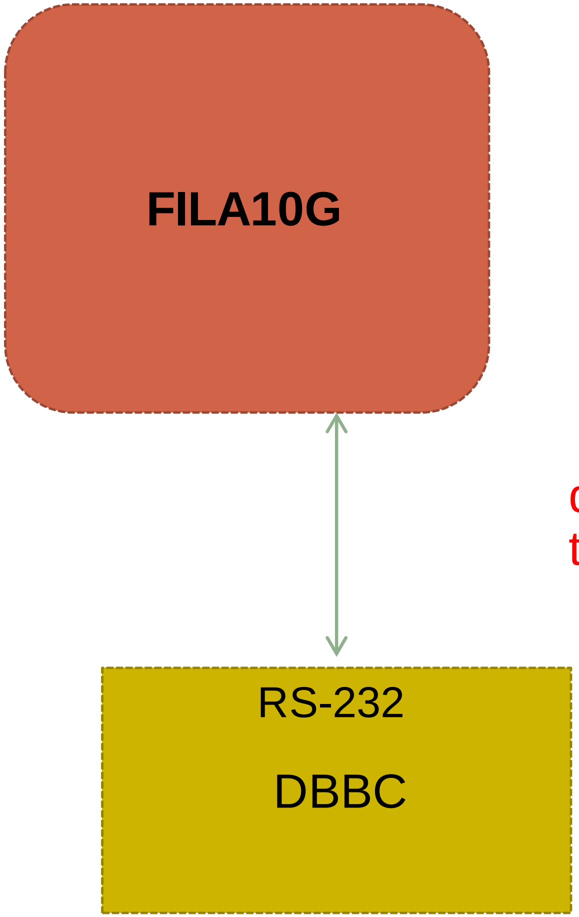
How to configure



if the DBBC is far

no multiple Xilinx programmers to be handled!

How to control



FILA10G

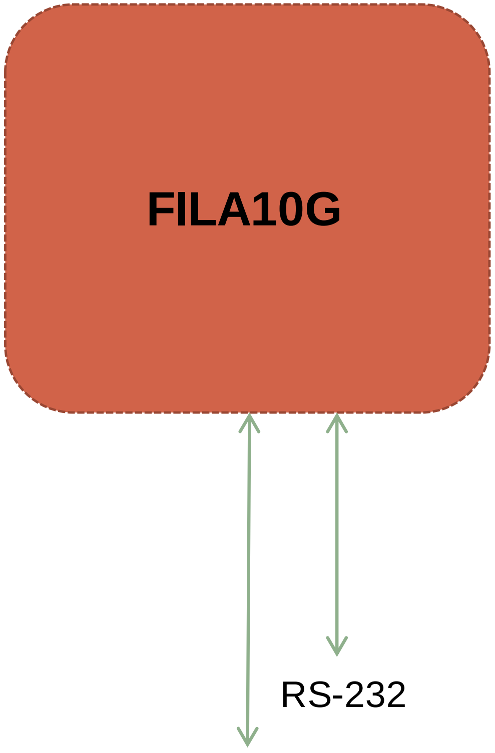
RS-232

DBBC

if the DBBC is nearby

direct access from
the Field System!

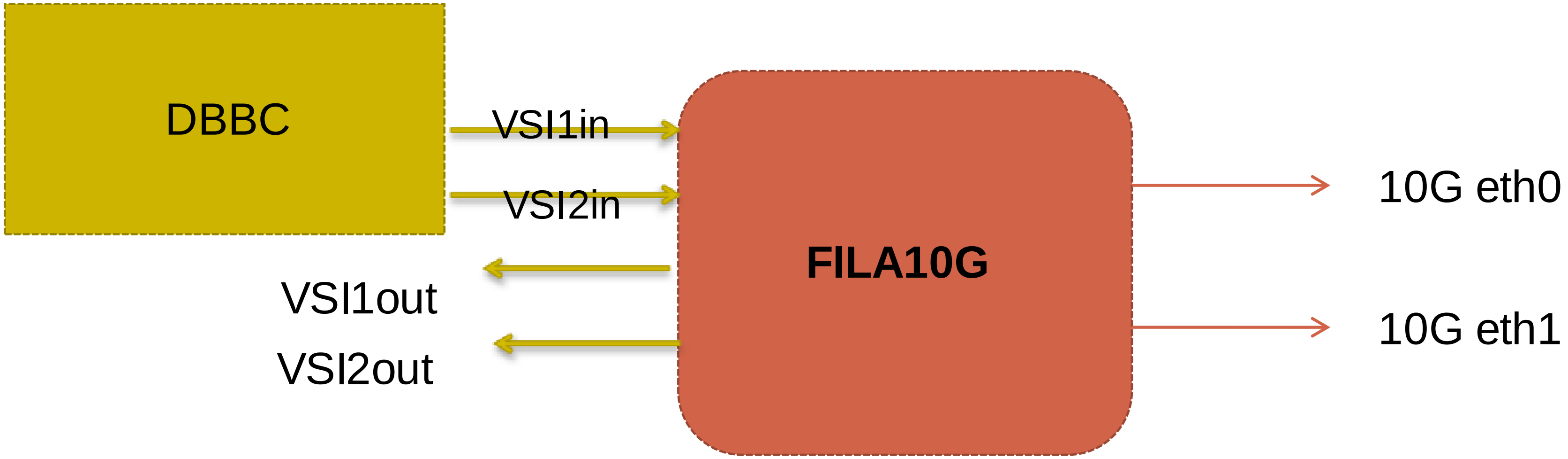
How to control



if the DBBC is far

select one of them

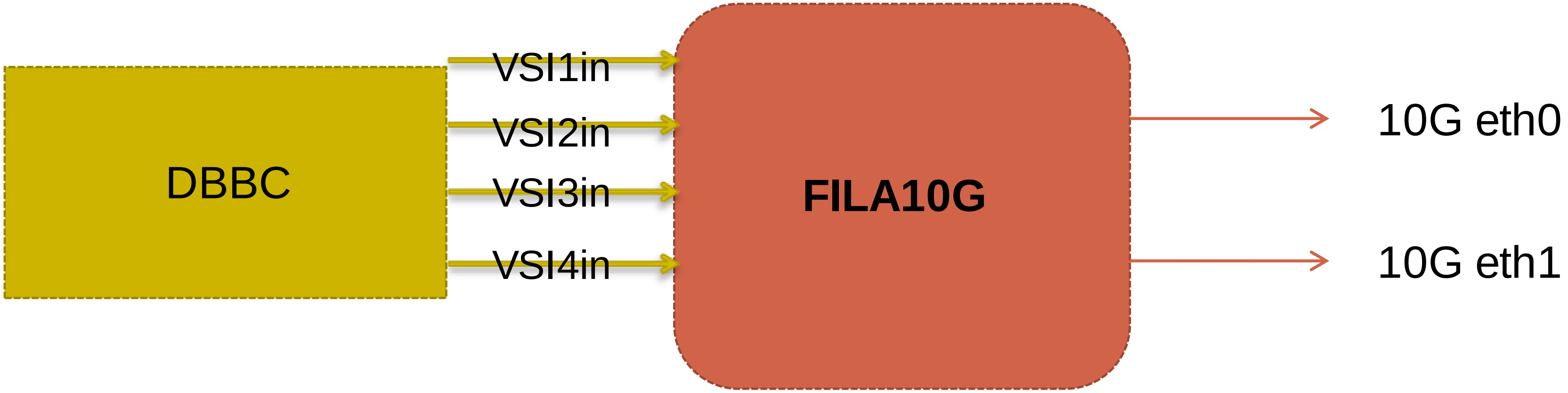
What to transfer



What is sent from the DBBC in VSI1/2 is transferred to 10G eth0/1 and copied on VSI1/2 out

Es. 2 x 2 Gbps = 4 Gbps

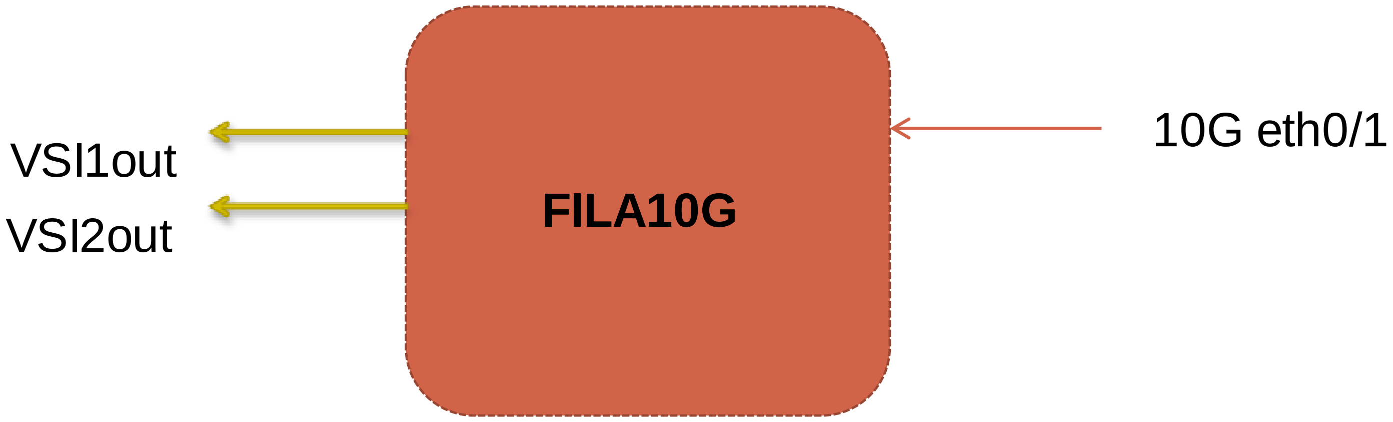
What to transfer



What is sent from the DBBC in VSI1/2/3/4 is transferred to 10G eth0/1

Es. 4 x 2 Gbps = 8 Gbps

What to transfer

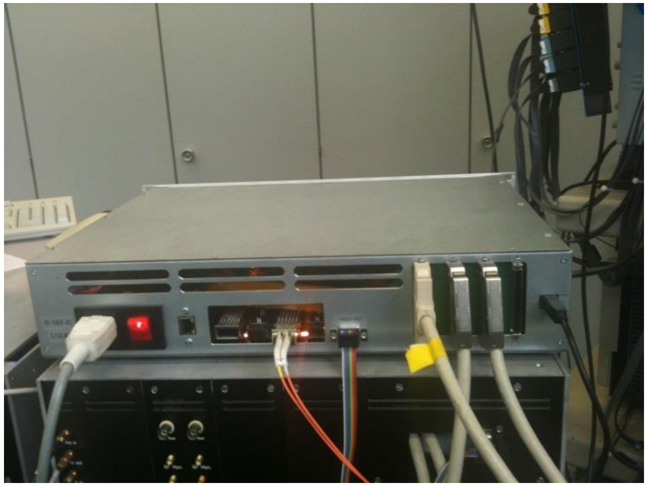


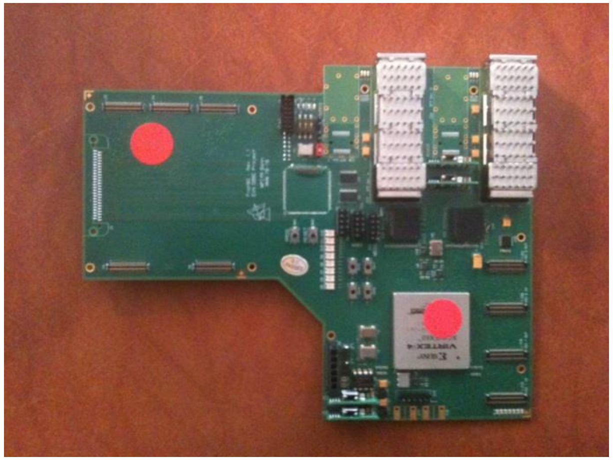
What is sent from the 10G eth0/1 is transferred on VSI1/2 out

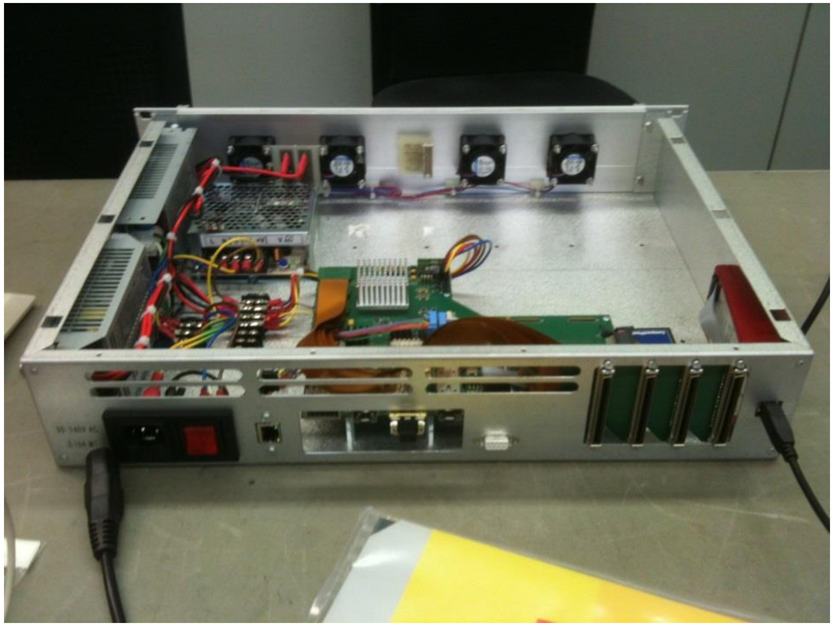
Which format

- MK5B up to 4 Gbps (native is 2Gbps)
- VDIF Single Thread
- VDIF Multiple Threads
- RAW (no headers are added)
- Threads can be fed by a selection of data channels eventually corner turned
- The two 10G Ethernet ports are independent in the destination address
- The two 10G Ethernet ports in multi-thread mode support an independent block of destination address coupled with the thread content selection

How it look like







What to set to control it

- Physical data input (VSI, 10G)
- Selection of data from the physical input
- Decimation of data input
- Physical data output (VSI, 10G)
- Operational status set (start, stop, etc.) and Data format
- Parameters in the data format
- Time synchronization
- Ethernet settings (local and destination IP, gateway IP address, port, etc.)
- Status monitoring
- Testing mode

Physical data input selection

inputselect

Selects one of the available input data sources.

Arguments: <input source>

<input source>: valid values are: (tvg|vsi1|vsi2|vsi1-2|vsi1-2-3-4)

FiLa10G's physical input width is 128 bit wide.

vsi_samplerate

Gets or sets the VSI input sample rate (samples per second).

Arguments: [<#samples per second> [<decimation divisor>]]

<#samples per second> (optional): lets the system know the sample rate of the VSI input (samples per second)

<decimation divisor> (optional): a divisor d in the range 1..255.

d=1 is the default, i.e. no decimation. d>1 decimates the input such that the resulting sample rate is 1/d of the original rate.

All arguments are optional. If the command is called without arguments the current VSI input sample rate is displayed.

Selection of data from the physical input

vsi_bitmask

Gets or sets the VSI input bitmask.
Arguments: [[[[<bitmask-4>] <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 up to four 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.

Decimation of data input

vsi_inputwidth

Gets or sets the effective bit width of the VSI input, that is to be processed by FiLa10G.
Arguments: [<input width in bits>]
<input width in bits> (optional): allowed input widths: 1,2,4,8,16,32,64,128
If the command is called without arguments the current input width is displayed. Otherwise the input width is set to the new value.
FiLa10G has a 128 bit wide input width, such that up to 128 parallel input bit streams are supported.
Only the <input width in bits> bit streams with the lowest index are fed to the processing pipeline.
The internal data path width of FiLa10G is 64 bit. Input widths lower than 64 result in the input data being packed to the internal data path width. An input width of 128 will result in the input data being serialized to two consecutive 64 bit words.
When a VSI input is selected ("inputselect"), the input width is automatically set to the actual size of the selected input.
Note: In general, the input width should be equal to or lower than the width of the actual VSI input source. Higher input widths will likely waste memory on the recorder.
Note: multi-channel VDIF format is only supported for the input widths 32 and 64.
Note: the "VSI input width" is also modified by the command "vsi_bitmask". Thus, using a custom bitmask and a custom input width at the same time may lead to unexpected results.

Operational status set and Data format

reboot

Reboots the system.
FiLa10G'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 FiLa10G system boot again.
Warning: all previously done settings and states are lost when rebooting.

reset

Resets FiLa10G's datapath and erases synchronized time.
Arguments: [keepsync]
If "reset" is called without arguments, the complete data path is reset and time synchronization is lost.
If "reset" is called with argument "keepsync", FiLa10G tries to maintain the current time synchronization. For this to work the input stage of the data path and the timers are not reset.

start

Starts/restarts sending of (formatted) output data.
Arguments [(vdif|mk5b|raw) [force]]
The used output data format is either VDIF, Mark5B or raw format. Raw format requires no time synchronization. The VDIF and Mark5B formats require the respective timer to be set.

stop

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

Physical data output

vsi_outputselect

Selects the VSI output data source.

Arguments: [<output data source>]

<output data source> (optional): (vsi1-2|vsi<X>-<Y>|output<X>|eth<X>|eth<X>-data)

If the command is called without arguments the current VSI pack mode is displayed.

The data that appears at the VSI outputs can be configured by the user. By default the VSI-1 input is connected to VSI-1 output and VSI-2 input is connected to VSI-2 output.

The following sources are available:

- **vsi1-2**: the default output
- **vsi1-1**: both VSI outputs are connected to VSI-1
- **vsi2-2**: both VSI outputs are connected to VSI-2
- **vsi2-1**: like the default, but having VSI-1 and VSI-2 flipped
- **output0**: VSI-1 output is connected to the lower 32 bits of the data stream at output 0 of the FiLa10G core. VSI-2 output is connected to the respective higher 32 bits.
- **output1**: VSI-1 output is connected to the lower 32 bits of the data stream at output 1 of the FiLa10G core. VSI-2 output is connected to the respective higher 32 bits.
- **eth0**: VSI-1 output is connected to the lower 32 bits of the data stream received from Ethernet core "eth0". VSI-2 output is connected to the respective higher 32 bits.
- **eth1**: VSI-1 output is connected to the lower 32 bits of the data stream received from Ethernet core "eth1". VSI-2 output is connected to the respective higher 32 bits.

Physical data output (continue)

……vsi_outputselect

- **eth0-data**: same as "eth0" with the difference that all VDIF and Mark5B header data is cut from
the incoming data stream.
- **eth1-data**: same as "eth0" with the difference that all VDIF and Mark5B header data is cut from the incoming data stream.

The PPS signal can be reconstructed correctly from the packet stream being received over Ethernet, if the packet format can be recognized and the packets are well-formed and arrive in sequential order.

The stream received from an Ethernet source is recognized as Mark5B data if the incoming packet size is exactly 5008 bytes and the used sync word equals 0xabaddeed. It is interpreted as Raw data if the packet size is exactly 8192 bytes, which disables any further header analysis and especially the PPS signal reconstruction feature. It is interpreted as VDIF data in any other case.

Applies to VDIF and Mark5B format only: The PPS signal is reconstructed and attached to the output stream whenever the first packet in a second interval is detected (see respective standard for details).

Note: the VSI output uses the same clock as the selected source. This is either the current system clock or the clock from the VSI input ports.

Parameters in the 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.

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.

Parameters in the data format (continue)

…….vdif_frame

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) If <payload size in bytes> will result in the last frame of a second interval to be incomplete, this frame is silently discarded.

vdif_interval

Gets/sets the active frame ID interval within a 1PPS period.
Arguments: [<minimum frame ID> <maximum frame ID>]
<minimum frame ID> (optional): the minimum frame ID that is transmitted to the output.
allowed range: 0...4294967294
<maximum frame ID> (optional): the maximum frame ID that is transmitted to the output.
allowed range: 0...4294967295
"*" is a shortcut for maximum frame ID.
Frames are filtered by their ID, such that only frame with an ID in the range <minimum frame ID> to <maximum frame ID> appear in the output stream. The others are completely discarded from the stream.
The frame index starts with 0 for the first VDIF frame in the second interval. The frame index increases with each following VDIF frame until it restarts again with 0 for the first frame of the next second interval.
Execute "vdif_interval 0 *" to reset the interval setting to its standard value.
Note: the frame ID interval setting applies to the VDIF data format only.

Parameters in the data format (continue)

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_legacy

Enables (on) or disables (off) VDIF legacy headers.
Arguments: (on|off)
Note: setting this value directly affects the header data of the VDIF data format.

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.

Time synchronization

mk5b_timesync

Performs Mark5B time synchronization to the active 1PPS source.
Arguments: <years since 2000> <modified Julian day> <seconds>
<years since 2000>: the number of years since 2000 (=currentYear-2000)
<modified Julian day>: the current modified Juilian day (only the last three digits are required)
<seconds>: the past seconds within the current modified Julian day.
A valid PPS signal is required for the time synchronization to succeed. If successful, the Mark5B time is synchronized with this command.

vdif_timesync

Performs VDIF time synchronization to the active 1PPS source.
Arguments: <half years since 2000> <seconds>
<half years since 2000>: the number of past half years since 2000 (only complete half years are counted)
<seconds>: the number of seconds since the beginning of the current half year
A valid PPS signal is required for the time synchronization to succeed. If successful, the VDIF time is synchronized with this command.

Time synchronization (continue)

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)
A valid PPS signal is required for the time synchronization to succeed. If successful, both VDIF time and Mark5B time are synchronized by this command.

Ethernet settings

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

destination
Gets/sets the output destination.
Arguments: <output index> [(<IPv4 address>[:<port>]|none) [<thread ID>]]
<output index>: the index of the output for which the destination is to be got or set.
allowed values: 0, or 1
<IPv4 address>(optional): IP address, format x.x.x.x
<port>(optional): IP port number
<thread ID>(optional): specifies the ID of the data thread for which the destination is to be set
Execute the command in the form "destination <output index>" to get a list of all current destination settings for the respective output.

tengbarp
Sets one ARP entry in a 10Gb Ethernet device.
Arguments: <device name> <ARP table index> <MAC address>
<device name>: "eth0", or "eth1"
<ARP table index>: index of ARP table entry to be modified
<MAC address>: MAC address to be set

Ethernet settings (continue)

tengbcfg
Sets the parameters of a 10Gb Ethernet device.
Arguments: <device name> <tengbcfg parameters>
<device name>: "eth0", or "eth1"
<tengbcfg parameters>: the paramters to be set

tengbinfo
Retrieves the current parameters of a 10Gb Ethernet device.
Arguments: <device name>
<device name>: "eth0", or "eth1"

Status monitoring

output

Displays output debug information.
Arguments: [<output index> [<frame ID> [<duration>]]]
<output index>: 0 or 1 (in current implementation both outputs send same data)
<frame ID>: Index of frame within current second starting with 0.

sysstat

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

tick

Enables (on) or disables (off) continuous 1PPS tick display on the console.
Arguments: (on|off)
Note: for compatibility to legacy batch scripts "tick" may be called without arguments which toggles the state between "on" and "off".

time

Displays the current time of the active 1PPS source.
Arguments: –
The displayed time is the (synchronized) VDIF time in UTC format.

version

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

Testing mode

tvg_mode

Gets or sets the test vector generator mode.
Arguments: [<tvg mode>]
<tvg mode> (optional): valid values are: (all-0|all-1|vsi-h|cnt)
If the command is called without arguments the current TVG mode is displayed.

tvg_samplerate

Gets or sets the TVG sample rate (samples per second).
Arguments: [<#samples per second>]
<#samples per second> (optional): sets the sample rate (samples per second) of the TVG input.
Valid range is: 1...64000000
Note: the TVG is operated with a 64MHz clock.

Other commands…..

A number of hidden commands are avilable for inner functionality and board testing

Commands that you do not find in the previous list are still available as legacy and hidden commands, so if you could still use them for some time

For more details or missing parts see the manual (it is improving!)

Example of commands sequence

```
...

inputselect vsi1-2

reset

vsi_samplerate 64000000

timesync 2013-05-22T15:30:00+01:00

vdif_station EF

vdif_frame 2 1

destination 0 192.169.68.10:46227

destination 1 192.169.68.11:46227

start vdif
...
stop
...
start
...
```