

VciLogConsole

1) Functional Description

This VCI spy is a component reading everything that goes through a VCI port and dumps every command and its associated response. Moreover, this component assert VCI protocol rules.

It uses a special VciMonitor port type, which is all-input.

Sample output:

```
logger WRITE 1 cells @0x716000cc plen = 1 const = 0 (r)srcid = 0 be = 1
716000cc: 00XXXXXX
Response: OK 1 cells

logger READ 1 cells @0x601094e0 plen = 32 const = 0 (r)srcid = 0 be = 15
Response: OK 8 cells
601094e0: 00004490 21180501 0100a524 fbffc514 000064a0 0800e003 21104001 1a250408
```

2) Component definition & usage

source:trunk/soclib/soclib/module/test_control_component/vci_logger/caba/metadata/vci_logger.sd?

See [SoclibCc/VciParameters](#)

```
Uses( 'caba:vci_logger', **vci_parameters )
```

It can be connected anywhere on a VCI signal:

```
caba::VciSignals<vci_param> signal_vci_m[4];

...

soclib::caba::VciLogger<vci_param> logger("logger", maptab);
logger.p_clk(signal_clk);
logger.p_resetr(signal_resetr);
logger.p_vci(signal_vci_m[0]);
```

3) CABA Implementation

CABA sources

- interface :
source:trunk/soclib/soclib/module/test_control_component/vci_logger/caba/source/include/vci_logger.h?
- implementation :
source:trunk/soclib/soclib/module/test_control_component/vci_logger/caba/source/src/vci_logger.cpp?

CABA Constructor parameters

```
VciLogConsole(
    sc_module_name name, // Instance name
    const soclib::common::MappingTable &mt) // Mapping Table
```

Example instantiation:

```
VciLogger logger("logger", mapping_table );
```

CABA Ports

- `sc_in<bool> p_resetn` : Global system reset
- `sc_in<bool> p_clk` : Global system clock
- `soclib::common::VciMonitor<vci_param> p_vci` : The VCI port