

VciSimHelper

1) Functional Description

This VCI target is a synthetic component that can be used to instrument the simulation process. It can make the simulation stop, exit or crash different ways.

It has four mapped register accepting only writes.

Possible actions are:

- Make the simulator call `sc_stop()`
- Make the simulator call `exit( val )`, with a given `val`
- Make the simulator throw an exception with a given value in message
- Make the simulator pause (i.e. wait for a keyboard event)

Registers are:

- `SIMHELPER_SC_STOP` (at offset 0x0): Call `sc_stop()`
- `SIMHELPER_END_WITH_RETVAL` (at offset 0x4): Call `exit( val )`
- `SIMHELPER_EXCEPT_WITH_VAL` (at offset 0x8): Create a `soclib::exception::RunTimeError("Simulation yielded error level "+val)`
- `SIMHELPER_PAUSE_SIM` (at offset 0xc): Pause simulation

2) Component definition & usage

source:trunk/soclib/soclib/module/verification_component/vci_simhelper/metadata/vci_simhelper.sd

See [SoclibCc/VciParameters](#)

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

3) CABA Implementation

CABA sources

- interface :
[source:trunk/soclib/soclib/module/verification_component/vci_simhelper/caba/source/include/vci_simhelper.h?](#)
- implementation :
[source:trunk/soclib/soclib/module/verification_component/vci_simhelper/caba/source/src/vci_simhelper.cpp?](#)

CABA Constructor parameters

```
VciSimhelper(  
    sc_module_name name,    // Instance name  
    const soclib::common::IntTab &index,    // Target index  
    const soclib::common::MappingTable &mt)    // Mapping Table
```

Example instantiation:

```
VciSimhelper simhelper("simhelper",  
    IntTab(2,3),  
    mapping_table );
```

CABA Ports

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