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 six mapped register: 5 accepting only writes, and one accepting only reads.

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 raise the interrupt signal SIGINT
- Make the simulator pause (i.e. wait for a keyboard event)
- Make the simulator return the number of cycles of simulation

Registers accepting writes 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
- `SIMHELPER_SIGINT` (at offset 0x14): Raise interrupt signal (SIGINT)

Register accepting reads is:

- `SIMHELPER_CYCLES` (at offset 0x10): Returns the number of cycles

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 instanciation:

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

CABA Ports

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

4) TLM-DT Implementation

TLM-DT sources

- interface : source:trunk/soclib/soclib/module/verification_component/vci_simhelper/tlmdt/source/include/vci_simhelper.h?
- implementation : source:trunk/soclib/soclib/module/verification_component/vci_simhelper/tlmdt/source/src/vci_simhelper.cpp?

TLM-DT Constructor parameters

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

Example instanciation:

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

TLM-DT Ports

- **p_vci** : The VCI target port