## **VciLocks Functional Description**

#### SOCLIB COMPONENTS INDEX

Ø

This VCI target is a locks controller : In VCI-based systems, it is not anymore possible to "lock the bus" to implement the atomic *test & set* instructions used for software synchronisation. Therefore, this memory mapped hardware peripheral implements a set of binary locks:

- Each binary lock is a single flip-flop, but corresponds to 4 bytes in the address space. The segment allocated to this component must be aligned on a 4 bytes boundary. The number of available locks is defined by segment\_size / 4.
- Any read request is interpreted as a *test & set* operation : the value stored in the addressed flip-flop is returned, and the addressed flip-flop is set to 1.
- All write request are interpreted as *reset* : the addressed flip-flop is reset to 0.

This way, a spin lock is implemented as a simple loop waiting to read 0, and the lock release is a simple write operation. This components checks addresses for segmentation violation, and can be used as default target.

# **VciLocks CABA Implementation**

The caba implementation is in

- source:trunk/soclib/systemc/include/caba/target/vci\_locks.h
- source:trunk/soclib/systemc/src/caba/target/vci\_locks.cc

## **Template parameters:**

• The VCI parameters

## **Constructor parameters**

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

#### **Ports**

- sc\_in<bool> p\_resetn : Global system reset
- sc\_in<bool> p\_clk : Global system clock
- soclib::common::VciiTarget<vci\_param> p\_vci : The VCI port