# VciLocks

## 1) Functional Description

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.

## 2) Component definition & usage

source:trunk/soclib/soclib/module/internal_component/vci_locks/caba/metadata/vci_locks.sd?

See SoclibCc/VciParameters

```
Uses( 'vci_locks', **vci_parameters )
```

## 3) CABA Implementation

### CABA sources

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

### CABA Constructor parameters

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

### CABA Ports

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

# 4) TLM-T implementation

The TLM-T implementation is not yet available.