

VciLocks 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.

Component definition

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

Usage

VciLocks has no other parameter than VCI ones, it may be used like others, see [SoclibCc/VciParameters](#)

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

VciLocks CABA Implementation

The caba implementation is in

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

Template parameters

template<typename vci_param>

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::caba::VciTarget<vci_param> p_vci` : The VCI port