VciMultiTimer Functional Description

This VCI target is a memory mapped peripheral that can control up to 256 software controlled timers. Each timer can optionally generates an independent periodic interrupt. The memory segment allocated to this component must be aligned on 4K bytes boundary. The timer index i is defined by the ADDRESS[12:4] bits. This hardware component cheks for segmentation violation, and can be used as a default target.

Each timer contains 4 memory mapped registers:

- **TIMER_VALUE** : ADDRESS[3:0] = 0x0

This 32 bits register is unconditionnally incremented at each cycle. A read request returns the current time contained in this register. A write request sets a new value in this register.

- **TIMER_RUNNING** : ADDRESS[3:0] = 0x4

When the Boolean value contained in this register is true, the corresponding interrupt is enabled. A write request of a zero gives resets this register. A write request of a non-zero value sets this registe.

- **TIMER_PERIOD** : ADDRESS[3:0] = 0x8

This 32 bits register defines the period between two successive interrupts. A write request writes a new value in this register, and the TIMER_RUNNING register is set to false. A read request returns the current value in this register.

- **TIMER_RESETIRQ** : ADDRESS[3:0] = 0xC

Any write request in this Boolean register will reset the pending IRQ. A read request returns the zero value when there is no pending interrupt, and returns a non zero value if there is a pending interrupt.

VciMultiTimer CABA Implementation

The caba implementation is in

- source:trunk/soclib/systemc/include/caba/target/vci_multi_timer.h
- source:trunk/soclib/systemc/src/caba/target/vci_multi_timer.cc

Template parameters:

- The VCI parameters

Constructor parameters

```
VciMultiTimer(
     sc_module_name name,   //  Component Name
     const soclib::common::IntTab & index,  // Target index
     const soclib::common::MappingTable &mt,   // MappingTable
     size_t nirq);   //  Number of available timers
```

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
- sc_out<bool> ***p_irq** : Pointer on Interrupts ports table