

VciMultiRam Functional Description

This VCI target is an embedded SRAM controller. This hardware component handles independent memory segments. Each segment is defined by a base address and a size (number of bytes). Both the base and the size parameters must be multiple of 4. The segments allocated to a given instance of this component are defined in the [Mapping Table](#).

Each segment is implemented as a dynamically allocated array in the constructor.

A MultiRam will initialize its segments from a binary if an ElfLoader? is attached to it.

VciMultiRam CABA Implementation

The caba implementation is in

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

Template parameters:

- The VCI parameters

Constructor parameters

- Elf-initialized MultiRam:

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

- Uninitialized MultiRam:

```
VciMultiRam(  
    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::VciTarget<vci_param> **p_vci** : The VCI port

Usage

```
// You may create a loader, loading "a.out"  
soclib::common::ElfLoader loader( "a.out" );
```

```
// You may create a MultiRam, using the loaded object
soclib::caba::VciMultiRam ram0( "ram0", soclib::common::IntTab(2), mapping_table, loader );
```

On reset, any loadable segment in ELF file will be loaded at matched location.