

# VciSimpleRom

## 1) Functional Description

This VCI target is a multi-segments ROM controller.

The VCI DATA width must be 32 or 64 bits. The VCI ADDRESS and the PLEN field must be multiple of 4 bytes.

This component handles one or several independent memory segments. Each segment is defined by a base address and a size (number of bytes).

A READ burst command packet (such a cache line request) contains one single flit. The number of flits in the response packet depends on the PLEN field:

- If VCI DATA width = 32 bits, each flit contains 4 bytes, and the number of flits is PLEN/4.
- If VCI DATA width = 64 bits, and PLEN define an even number of words, each flit contains 8 bytes, and the number of flits is PLEN/8.
- If VCI DATA width = 64 bits, and PLEN define an odd number of words, the last flit contains only 4 bytes (right justified), and the number of flits is PLEN/8 + 1.

The binary code contained in the ROM can be loaded from an ELF binary file by the [Loader](#) defined as a constructor argument.

## 2) Component definition & usage

[source:trunk/soclib/soclib/module/internal\\_component/vci\\_simple\\_rom/caba/metadata/vci\\_simple\\_rom.sd?](source:trunk/soclib/soclib/module/internal_component/vci_simple_rom/caba/metadata/vci_simple_rom.sd?)

```
Uses( 'vci_simple_ram' )
```

## 3) CABA Implementation

### CABA sources

- interface :  
[source:trunk/soclib/soclib/module/internal\\_component/vci\\_simple\\_rom/caba/source/include/vci\\_simple\\_rom.h?](source:trunk/soclib/soclib/module/internal_component/vci_simple_rom/caba/source/include/vci_simple_rom.h?)
- implementation :  
[source:trunk/soclib/soclib/module/internal\\_component/vci\\_simple\\_rom/caba/source/src/vci\\_simple\\_rom.cpp?](source:trunk/soclib/soclib/module/internal_component/vci_simple_rom/caba/source/src/vci_simple_rom.cpp?)

### CABA Constructor parameters

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

### CABA Ports

- **p\_resetn** : hardware reset

- **p\_clk** : clock
- **p\_vci** : The VCI port