

# VciDma Functional Description

This VCI component is both a target and an initiator.

- Addressing as a target allows to configure it for a transfer.
- Initiator will do the transfer

There is only one DMA context handled at a time

An IRQ is optionally asserted when transfer is finished.

This hardware component checks for segmentation violation, and can be used as a default target.

## Memory region layout

- DMA\_SRC: ADDRESS[3:0] = 0x0

The source of memory copy.

- DMA\_DST: ADDRESS[3:0] = 0x4

The destination of memory copy

- DMA\_LEN: ADDRESS[3:0] = 0x8

Length of transfer, in bytes. Writing to this register initiates the transfer, you should write to it after src and dst. This register gets back to 0 when transfer is finished.

- DMA\_IRQ\_ENABLED: ADDRESS[3:0] = 0xC
  - ◆ Written to: A boolean enabling the IRQ line (0 is disabling)
  - ◆ Read from: A boolean indicating the completion of the transfer, (0 is completed)

## Component usage

For extensibility issues, you should access the DMA using globally-defined offsets.

You should include file source:trunk/soclib/include/soclib/dma.h from your software, it defines DMA\_SRC, DMA\_DST, DMA\_LEN, DMA\_IRQ\_ENABLED.

Sample code:

```
#include "soclib/dma.h"

static const volatile void* dma_address = 0xc0000000;

void * memcpy(void *dst, const void *src, const size_t len)
{
    volatile int *dma = ((int*)dma_address);

    dma[DMA_DST] = (uint32_t)dst;
    dma[DMA_SRC] = (uint32_t)src;
    dma[DMA_LEN] = len;
    dma[DMA_IRQ_ENABLED] = 1;
}
```

```

        dma[DMA_LEN] = (uint32_t)len;
        while ( dma[DMA_LEN] )
            ;
        return dst;
    }

```

(add -I/path/to/soclib/include to your compilation command-line)

## Component definition

Available in source:trunk/soclib/desc/soclib/vci\_dma.sd

## Usage

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

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

## VciDma CABA Implementation

The caba implementation is in

- source:trunk/soclib/systemc/include/caba/target/vci\_dma.h
- source:trunk/soclib/systemc/src/caba/target/vci\_dma.cc

## Template parameters:

- The VCI parameters

## Constructor parameters

```

VciDma (
    sc_module_name name,    // Component Name
    const soclib::common::IntTab & index, // Target index
    const soclib::common::MappingTable &mt, // MappingTable
    const size_t burst_size ); // Number of bytes transfered in a burst

```

## Ports

- sc\_in<bool> **p\_resetrn** : Global system reset
- sc\_in<bool> **p\_clk** : Global system clock
- soclib::caba::VciTarget<vci\_param> **p\_vci\_target** : The VCI target port
- soclib::caba::VciInitiator<vci\_param> **p\_vci\_initiator** : The VCI initiator port
- sc\_out<bool> **p\_irq** : Interrupt port