

downsampling

1) Functional Description

The downsampling operation consists in reducing the number of input (i.e. decreasing the sampling rate in our case). In our system, the downsampling factor is 4. This means that we reduce the number of input symbols by 4. The architecture of the downsampling component is presented in the figure 1. It is composed of a downsampling core and a MWMR wrapper. The wrapper is used to interface the core and the MWMR controller available here [VciMwmrController](#).



2) CABA Implementation

a) Component definition & usage

Component definition

- [source:trunk/soclib/soclib/module/ofdm_chain_components/downsampling/caba/metadata/downsampling.sd?](#)

Usage

Downsampling has a *fifo_depth* parameter, which defines the fifo depth for the input. For example with a FIFO depth equal to 16 :

```
Uses('Downsampling', fifo_depth = 16);
```

b) CABA sources

- interface :
[source:trunk/soclib/soclib/module/ofdm_chain_components/downsampling/caba/source/include/downsampling.h?](#)
- implementation :
[source:trunk/soclib/soclib/module/ofdm_chain_components/downsampling/caba/source/src/downsampling.cpp?](#)

CABA Constructor parameters

```
Downsampling(  
    sc_module_name name,                // Instance name  
    int ncycles) // Number of computation cycles
```

CABA Ports

- `sc_in<bool> p_resetrn` : hardware reset
- `sc_in<bool> p_clk` : clock
- `soclib::caba::FifoOutput<uint32_t> p_to_ctrl` : interface from the downsampling to the MWMR controller
- `soclib::caba::FifoInput<uint32_t> p_from_ctrl` : interface from the MWMR controller to the downsampling

3) TLM-DT Implementation

a) Component definition & usage

Component definition

- [source:trunk/soclib/soclib/module/ofdm_chain_components/downsampling/tlmdt/metadata/downsampling.sd?](#)

b) TLM-DT sources

- interface :
[source:trunk/soclib/soclib/module/ofdm_chain_components/downsampling/tlmdt/source/include/downsampling.h?](#)
- implementation :
[source:trunk/soclib/soclib/module/ofdm_chain_components/downsampling/tlmdt/source/src/downsampling.cpp?](#)

TLM-DT Constructor parameters

```
Downsampling(sc_core::sc_module_name name, // Instance name
             uint32_t id,
             uint32_t read_fifo_depth, // Depth of input buffer
             uint32_t write_fifo_depth, // Depth of output buffer
             uint32_t n_read_channels, // Number of read channels
             uint32_t n_write_channels, // Number of write channels
             uint32_t n_config, // Number of configurations
             uint32_t n_status); // Number of status
```

TLM-DT Ports

- `std::vector<tlm_utils::simple_target_socket_tagged<Downsampling,32,tlm::tlm_base_protocol_types>*>`
p_config: configuration port
- `std::vector<tlm_utils::simple_target_socket_tagged<Downsampling,32,tlm::tlm_base_protocol_types>*>`
p_status: status port
- `std::vector<tlm_utils::simple_initiator_socket_tagged<Downsampling,32,tlm::tlm_base_protocol_types>*>`
p_read_fifo: port from the MWMM controller to the downsampling
- `std::vector<tlm_utils::simple_initiator_socket_tagged<Downsampling,32,tlm::tlm_base_protocol_types>*>`
p_write_fifo: port from the downsampling to the MWMM controller