

Synchronization

1) Functional Description

The synchronization operation consists in determining the beginning of the pilot sequence and to estimate and compensate the sampling clock difference between the transmitter and the receiver. The architecture of the synchronization component is presented in the figure 1. It is composed of a synchronization 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/synchronization/caba/metadata/synchronization.sd?](#)

Usage

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

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

b) CABA sources

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

CABA Constructor parameters

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

CABA Ports

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

3) TLM-DT Implementation

a) Component definition & usage

Component definition

- source:trunk/soclib/soclib/module/ofdm_chain_components/synchronization/tlmdt/metadata/synchronization.sd?

b) TLM-DT sources

- interface :
source:trunk/soclib/soclib/module/ofdm_chain_components/synchronization/tlmdt/source/include/synchronization.h?
- implementation :
source:trunk/soclib/soclib/module/ofdm_chain_components/synchronization/tlmdt/source/src/synchronization.cpp?

TLM-DT Constructor parameters

```
Synchronization(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<Synchronization,32,tlm::tlm_base_protocol_types>*>`
p_config: configuration port
- `std::vector<tlm_utils::simple_target_socket_tagged<Synchronization,32,tlm::tlm_base_protocol_types>*>`
p_status: status port
- `std::vector<tlm_utils::simple_initiator_socket_tagged<Synchronization,32,tlm::tlm_base_protocol_types>*>`
p_read_fifo: port from the MWMR controller to the synchronization
- `std::vector<tlm_utils::simple_initiator_socket_tagged<Synchronization,32,tlm::tlm_base_protocol_types>*>`
p_write_fifo: port from the synchronization to the MWMR controller