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 avalaible here VciMwmrController.

2) 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);
```

3) CABA Implementation

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

4) TLM-T Implementation

The TLM-T implementation is not yet available.