

Upsampling

1) Functional Description

The upsampling operation consists in adding zeros between the inputs (i.e. the OFDM symbols in our case). In our system, the upsampling factor is 4. This means that we add 3 zeros between each input. The architecture of the upsampling component is presented in the figure 1. It is composed of a upsampling core and a MWMR wrapper. The wrapper is used to interface the core and the MWMR controller available here [VciMwmrController](#).



2) Component definition & usage

Component definition

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

Usage

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

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

3) CABA Implementation

CABA sources

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

CABA Constructor parameters

```
Upsampling(  
    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 upsampling to the MWMR controller
- `soclib::caba::FifoInput<uint32_t> p_from_ctrl` : interface from the MWMR controller to the upsampling

4) TLM-T Implementation

The TLM-T implementation is not yet available.