

FIR128

1) Functional Description

The FIR (Finite Impulse Response) is a basic building block in any Digital Processing (DSP) system. The frequency response of the filter depends on the value of its coefficients, or taps. In our case, the tap number of the filter is equal to 128. The architecture of the FIR is presented in the figure 1. It is composed of a FIR128 core and a MWMM wrapper. The wrapper is used to interface the core and the MWMM controller available here [VciMwmmController](#).



2) CABA Implementation

a) Component definition & usage

Component definition

- [source:trunk/soclib/soclib/module/ofdm_chain_components/fir128/caba/metadata/fir128.sd?](#)

Usage

FIR128 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('Fir128', fifo_depth = 16);
```

b) CABA sources

- interface : [source:trunk/soclib/soclib/module/ofdm_chain_components/fir128/caba/source/include/fir128.h?](#)
- implementation :
[source:trunk/soclib/soclib/module/ofdm_chain_components/fir128/caba/source/src/fir128.cpp?](#)

CABA Constructor parameters

```
Fir128(  
    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 fir128 to the MWMM controller
- `soclib::caba::FifoInput<uint32_t> p_from_ctrl` : interface from the MWMM controller to the fir128

3) TLM-DT Implementation

a) Component definition & usage

Component definition

- source:trunk/soclib/soclib/module/ofdm_chain_components/fir128/tlmdt/metadata/fir128.sd?

b) TLM-DT sources

- interface :
source:trunk/soclib/soclib/module/ofdm_chain_components/fir128/tlmdt/source/include/fir128.h?
- implementation :
source:trunk/soclib/soclib/module/ofdm_chain_components/fir128/tlmdt/source/src/fir128.cpp?

TLM-DT Constructor parameters

```
Fir128(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<Fir128,32,tlm::tlm_base_protocol_types>*>`
p_config: configuration port
- `std::vector<tlm_utils::simple_target_socket_tagged<Fir128,32,tlm::tlm_base_protocol_types>*>`
p_status: status port
- `std::vector<tlm_utils::simple_initiator_socket_tagged<Fir128,32,tlm::tlm_base_protocol_types>*>`
p_read_fifo: port from the MWMMR controller to the fir128
- `std::vector<tlm_utils::simple_initiator_socket_tagged<Fir128,32,tlm::tlm_base_protocol_types>*>`
p_write_fifo: port from the fir128 to the MWMMR controller