Tc4200

1) Functional Description

This VCI target corresponds to a IEEE802.16e LDPC decoder. It embeds an internal hardware Cycle Accurate Bit Accurate model of the ?TurboConcept's IEEE802.16e WiMAX LDPC decoder ?tc4200.

TurboConcept's TC4200-WiMAX Core is a high speed Low Density Parity Check code (LDPC) decoder optimized for WiMAX (IEEE 802.16e) specifications. A patented decoding architecture allows meeting high throughputs within small devices, and still offering close-to-ideal Bit Error Rate (BER) performances.

Figure 1 presents the general core structure. The Tc4200 is made of a VCI wrapper and an internal hardware decoder model which communicates using proprietary FIFO-like protocols.

2) CABA Implementation

a) Component definition & usage

- source:trunk/soclib/soclib/module/streaming_component/tc4200/caba/metadata/tc4200.sd
- source:trunk/soclib/binary/module/streaming_component/tc4200/caba/doc

b) CABA sources

- interface : source:trunk/soclib/soclib/module/streaming_component/tc4200/caba/source/include/tc4200.h
- implementation : source:trunk/soclib/soclib/module/streaming_component/tc4200/caba/source/src/tc4200.cpp
- internal component interface : source:trunk/soclib/binary/module/streaming_component/tc4200/caba/include/tc_tc4200.h
- internal component library : source:trunk/soclib/binary/module/streaming_component/tc4200/caba/lib

CABA Constructor parameters

- IEEE802.16e LDPC decoder

```
Tc4200(
     sc_module_name name,                     // Instance name
     const soclib::common::IntTab &index,     //  Target index
     const soclib::common::MappingTable &mt)  // Mapping Table
```

CABA Addressable registers

- Read only registers
  - `TC4200_D_OUT` Data output register
  - `TC4200_MONITOR` Monitoring interface. See Figure 2

- Write only registers
  - ♦ `TC4200_CONFIG` Configuration interface. See Figure 3
  - ♦ `TC4200_D_IN_FIRST` Register for the First data corresponding to a new frame. See Figure 4.
  - ♦ `TC4200_D_IN` Register for any other input frame data. See Figure 5.

CABA Ports

- sc_in<bool> **p_resetn** : hardware reset
- sc_in<bool> **p_clk** : clock
- soclib::common::VciTarget<vci_param> **p_vci** : The VCI port

3) TLM-DT Implementation

The TLM-DT implementation assumes the instantiation of the MWMR component.

a) Component definition & usage

- source:trunk/soclib/soclib/module/streaming_component/tc4200/tlmdt/metadata/tc4200.sd
- source:trunk/soclib/binary/module/streaming_component/tc4200/tlmdt/doc

b) TLM-DT sources

- interface : source:trunk/soclib/soclib/module/streaming_component/tc4200/tlmdt/source/include/tc4200.h
- implementation : source:trunk/soclib/soclib/module/streaming_component/tc4200/tlmdt/source/src/tc4200.cpp
- internal component interface : source:trunk/soclib/binary/module/streaming_component/tc4200/tlmdt/include/tc_tc4200.h
- internal component library : source:trunk/soclib/binary/module/streaming_component/tc4200/tlmdt/lib

TLM-DT Constructor parameters

```
Tc4200(
     sc_module_name name,                    // Instance name
     uint32_t       id,
     uint32_t       MWMR2core_fifo_depth,
     uint32_t       core2MWMR_fifo_depth)
```

TLM-DT Ports

- std::vector<tlm_utils::simple_target_socket_tagged<Tc1700,32,tlm::tlm_base_protocol_types> *> **p_config**: configuration port
- std::vector<tlm_utils::simple_target_socket_tagged<Tc1700,32,tlm::tlm_base_protocol_types> *> **p_status**: status port
- std::vector<tlm_utils::simple_initiator_socket_tagged<Tc1700,32,tlm::tlm_base_protocol_types> *> **p_MWMR2core_fifo**: port from the MWMR controller to the Tc4200
- std::vector<tlm_utils::simple_initiator_socket_tagged<Tc1700,32,tlm::tlm_base_protocol_types> *> **p_core2MWMR_fifo**: port from the Tc4200 to the MWMR controller

4) Limitation

This model has the following two limitations:

- stopping criterion is disable;
- fixed number of performed iterations.

The number of performed iterations is fixed to a reasonable value still offering close-to-ideal Bit Error Rate (BER) performances. Please contact ?TurboConcept for information about these limitations.

5) License

The VCI wrapper is licensed under the SoCLib, GNU LGPLv2.1 license.

The VCI wrapper instantiates an internal hardware decoder. This internal hardware decoder is licensed under BSD-like license.

This internal hardware decoder is distributed in a binary form.

6) RTL model

Please contact ?TurboConcept for information about purchasing a fully functional RTL model of the internal hardware decoder.