PibusBcu Functional Description

This hardware component is a PIBUS controller. The tree basic functionalities are :

- arbitration between simultaneous initiator requests.
- selection of the adressed target, by decoding the MSB address bits.
- time-out, when the selected target does not complete the transaction.

The main features are the following :

- The default master mechanism is not supported. *, The supported PIBUS response codes are PI_ACK_RDY, PI_ACK_WAT, and PI_ACK_ERR.
- The arbitration policy between initiators is round-robin.
- This component uses the mapping table to build the ROM that decodes the address MSB bits and give the target index used to activate the proper p_sel[i] signal.
- The COUNT_REQ[i] register counts the total number of transactions for initiator i.
- The COUNT_WAIT[i] register counts the total number of wait cycles for initiator i.

PibusBcu CABA Implementation

The caba implementation is in

- source:trunk/soclib/systemc/include/caba/interconnect/pibus_bcu.h
- source:trunk/soclib/systemc/src/caba/interconnect/pibus_bcu.cc

Template parameters

template<typename vci_param>

Constructor parameters

```
PiBcu(
     sc_module_name name,   //  Instance Name
     const soclib::common::MappingTable &mp,   // Mapping Table
     size_t nb_master,   // Number of initiators
     size_t nb-slave,   // Number of targets
     uint32_t time_out);   // Time-out (number of cycles)
```

Ports

- sc_in<bool> **p_resetn** : Global system reset
- sc_in<bool> **p_clk** : Global system clock
- sc_out<bool> ***p_gnt** : Pointer to the grant ports array
- sc_in<bool> ***p_req** : Pointer to the request ports array
- sc_out<bool> ***p_sel** : Pointer to the select ports array
- sc_in<sc_uint<32> > **p_a** : Pibus address
- sc_in<bool> **lock** : Pibus lock (used for bursts)
- sc_in<sc_uint<2> > **p_ack** : Pibus response code
- sc_out<bool> **p_tout** : Pibus Time-out