VirtualDspinNetwork

1) Functional Description

The Virtual Dspin Network aims to connect a two-dimensionnal array of TSAR clusters together.

It virtualizes two virtual channels (usually the primary channel and the memory-coherence channel) on two physical networks VirtualDspinArray of different size (usually one for command paquets and one for response paquets).

The two physical networks and not connected between them.

It implements the optimized and confined broadcast protocol on first network for second channel and the Hypertransport routing on both networks for the first channel.

Input INFSM States

FSM_REQ	Waiting for paquet or sending first flit
FSM_DTN	Allocated to correct output for multicast
FSM_DT_11	Sending second flit to first output in multicast
FSM_REQ_12	Sending first flit to second output in multicast
FSM_DT_12	Sending second flit to second output in multicast
FSM_REQ_21	Sending first flit to third output in multicast
FSM_DT_21	Sending second flit to third output in multicast
FSM_REQ_22	Sending first flit to fourth output in multicast
FSM_DT_22	Sending second flit to fourth output in multicast
FSM_REQ_LOCAL	Sending first flit to fifth (LOCAL) output in multicast
FSM_DT_LOCAL	Sending second flit to fifth (LOCAL) output in multicast

Output INFSM States

LOCAL	Allocated to LOCAL input
NORTH	Allocated to NORTH input
SOUTH	Allocated to SOUTH input
EAST	Allocated to EAST input
WEST	Allocated to WEST input
NOP_LOCAL	Not allocated, and last allocated was LOCAL
NOP_NORTH	Not allocated, and last allocated was NORTH
NOP_SOUTH	Not allocated, and last allocated was SOUTH
NOP_EAST	Not allocated, and last allocated was EAST
NOP_WEST	Not allocated, and last allocated was WEST
NOP_WEST	Not allocated, and last allocated is unkwnow (equivalent of NOP_LOCAL)

2) Component definition & usage

source:trunk/soclib/soclib/module/network_component/virtual_dspin_network/caba/metadata/virtual_dspin_network.sd?

3) CABA implementation

CABA sources

interface	source:trunk/soclib/soclib/module/network_component/virtual_dspin_network/caba/source/include/virtual_d
implementation	source:trunk/soclib/soclib/module/network_component/virtual_dspin_network/caba/source/src/virtual_dspin

CABA Template parameters

int io_mask_size	Size in bits of IO checking
int io_number_size	Size in bits of IO index
int x_addressing_size	Size in bits of first coordinate addressing
int y_addressing_size	Size of second coordinate addressing
int cmd_data_size	Size in bits of command flits
int cmd_io_mask_offset	Emplacement of IO checking in command paquets
int cmd_io_number_offset	Emplacement of IO index in IO table in command paquets
int cmd_x_addressing_offset	Emplacement of target x in first flit in command paquets
int cmd_y_addressing_offset	Emplacement of target y in first flit in command paquets
int cmd_eop_offset	Emplacement of eop checking in command paquets
int cmd_broadcast_offset	Emplacement of broadcast checking in command paquets
int rsp_data_size	Size in bits of response flits
int rsp_io_mask_offset	Emplacement of IO checking in response paquets
int rsp_io_number_offset	Emplacement of IO index in IO table in response paquets
int rsp_x_addressing_offset	Emplacement of target x in first flit in response paquets
int rsp_y_addressing_offset	Emplacement of target y in first flit in response paquets
int rsp_eop_offset	Emplacement of eop checking in response paquets
int in_fifo_size	Size of input fifos
int out_fifo_size	Size of output fifos
int x_min_offset	Emplacement of x_min for broadcast confinement
int x_max_offset	Emplacement of x_max for broadcast confinement
int y_min_offset	Emplacement of y_min for broadcast confinement
int y_max_offset	Emplacement of y_max for broadcast confinement

CABA Constructor parameters

sc_module_name insname	instance name
int size_x	width of network
int size_y	height of network
clusterCoordinates<x_addressing_size, y_addressing_size> * aIO_table	list of IO Clusters

CABA ports

sc_in<bool>	p_clk	Global system clock
sc_in<bool>	p_resetn	Global system reset
DspinOutput<cmd_data_size>	*** p_out_cmd**	For each cluster and each virtual channel : out command
DspinInput<cmd_data_size>	* p_in_cmd	For each cluster and each virtual channel : in command
DspinOutput<rsp_data_size>	*** p_out_rsp**	For each cluster and each virtual channel : out response
DspinInput<rsp_data_size>	* p_in_rsp	For each cluster and each virtual channel : in response

4) TLMT implementation

The TLM-T implementation is not available yet.