ProNoC

Network-on-Chip Interface
(NI)

NI_master

NI_master is a Wishbone bus (WB)-based interface for the network-on-chip (ProNoC) router. This module has two WB master interfaces, one for sending and another for receiving data packets. The module is controlled using a slave WB. The proposed NI handles each NoC's virtual channel (VC) independently using two multi-channels DMAs (send- and receive-DMAs). The NI uses round robin arbitration between active VCs. An active VC is a VC which has a data to be sent or received. The winner VC is allowed to send/receive data via WB master interface until both of the flowing conditions are met a) The sender's/receiver's resources are available (i.e the sender's FIFO is not full/ the receiver's FIFO is not empty) b) Transfered data size does not reach the maximum burst size. Switching among active VCs, when their resources are not available anymore, relaxes the dependency between VCs. This is because a stuck active VC dose not affect the rests. Setting a burst size can be beneficial in case of sending long packets. It makes sure that a VC does not use the router output bandwidth for a long period of time and instead the bandwidth is shared among all active VCs.

Notice

The NI_master supports WB burst mode. Hence, to avoid performance degradation make sure to enable burst mode when you are calling memory module in processing tile generator.

NI-master Block diagram:

Figure1 depicts the functional block diagram of NI-master module.

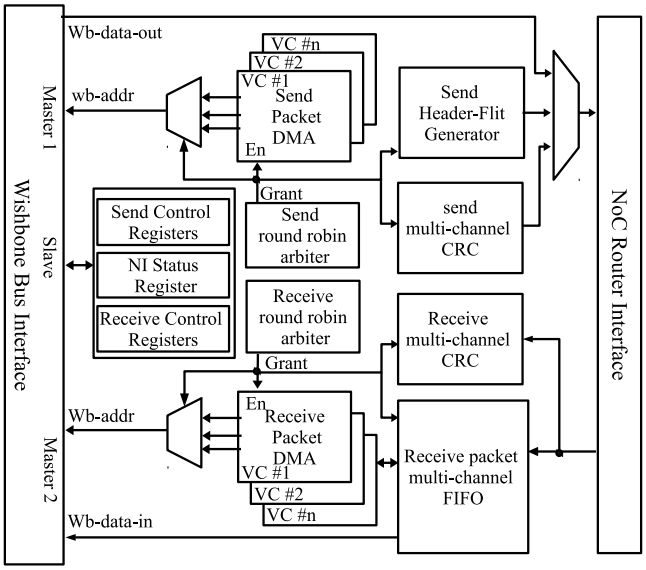

Figure 1: NI_master functional block diagram

Parameters
Description

Table 1: NI_master GUI Parameters.

Name	Description	Permitted values	Default
MAX-TRANSACTION-WIDTH	This parameter defines the maximum possible packet size (in flit) that can be sent via one single packet. The maximum packet size will be 2 power of MAX_TRANSACTION_WIDTH in words.	$n \in \mathbb{N}$, $4 \leqslant n \leqslant 32$	13
MAX-BURST-SIZE	The maximum burst size in words. The NI releases the Wishbone bus master interface each time one burst transaction is completed (or when the sender VC's FIFO becomes full). The bus will be released for one clock cycle. Then in case, there are other active VCs, another active VC will get access to the bus using a round robin arbiter. This process will be continued until there is not any active VC.	$n \in 2^m$, $1 \leqslant m \leqslant 11$	16
Dw	The Wishbone bus data width in bits.	$n \in 4 \times m$, $8 \leqslant m \leqslant 64$	32
CRC_EN	CRC32_Enable: It can be defined as "YES" to enable CRC32 or "NO" to disable it. If CRC is enabled, then two CRC32 generator modules will be added to the NI. One CRC generator for calculating CRC32 of sending packets and another for receiving packets. The CRC32 value of each packet is sent via tail flit and at destination NI, it will be compared with received packet generated CRC32. The matching results can be used for error-detection and can be read via NI slave interface.	"YES", "NO"	"NO"

Slave wishbone bus registers

The NI_master can be controlled using wishbone bus slave interface. Table 2 shows the NI_master internal registers. All NI_master registers are memory mapped.

Table 2: NI_master memory map registers. Note that n in address range is the virtual channel number where $0 \leq n < v$ and v is the number of VCs per router port.

addr[7:0]	Register Name	Description
0	STATUS1-WB-ADDR	NI first status register
1	STATUS2-WB-ADDR	NI second status register
2	BURST-SIZE-WB-ADDR	The DMA burst size in words
$(16n + 3)$	SEND-DATA-SIZE-WB-ADDR	The size of data to be sent in words
$(16n + 4)$	SEND-STRT-WB-ADDR	The start address of data to be sent in bytes
$(16n + 5)$	SEND-DEST-WB-ADDR	The destination router (IP) address
$(16n + 6)$	SEND-CTRL-WB-ADDR	The Send-DMA control register
$(16n + 7)$	RECEIVE-DATA-SIZE-WB-ADDR	The size of received data in words
$(16n + 8)$	RECEIVE-STRT-WB-ADDR	The address of receiver buffer in bytes
$(16n + 9)$	RECEIVE-SRC-WB-ADDR	The address of source router (the router which sent the last received packet).
$(16n + 10)$	RECEIVE-CTRL-WB-ADDR	The receive-DMA control register
$(16n + 11)$	RECEIVE-MAX-BUFF-SIZ	The allocated size for receiver buffer in words.

STATUS1-WB-ADDR

NI first status register includes four v-bit status variables (v indicates the number of VCs per router port). Each bit represents a VC i.e bit 0 represent VC number 0 and so on.

Bit	$4v$-1 … $3v$	$3v$-1 … $2v$	$2v$-1 … v	v-1 … 0
	send-vc-is-busy	receive-vc-is-busy	receive-vc-packet-is-saved	receive-vc-got-packet
Read/Write	R	R	R	R
Initial Value	0	0	0	0

receive-vc-got-packet: The asserted bit in this register indicates that its respective router's receiver VC (input VC) has a packet in its buffer.

receive-vc-packet-is-saved: The asserted bit in this register indicates that its respective VC's receive-DMA has done saving the packet to the memory (via the WB master interface). The asserted bit is reset by writing any value on its respective VC's RECEIVE-CTRL-WB-ADDR register.

receive-vc-is-busy: The asserted bit in this register indicates that its respective VC's receive-DMA is busy (active) and is receiving a packet. All VC's receive-DMA control registers ($16n$+7$\sim$11) are inaccessible to write when their respective VC's receive-DMA is in busy (active) state.

send-vc-is-busy: The asserted bit in this register indicates that its respective VC's sender DMA is busy (active)and is sending packet. All sender VC's DMA control registers ($16n$+3$\sim$6) are inaccessible to write when their respective VC's send-DMA is busy (active).

Note Several VCs can be in busy (active) state (send- or receive-packets) at the same time. However at each specific time at most two DMAs (one send-DMA and one receive-DMA) can be enabled for sending/receiving packets to/from the network. The enabled VC can be detected by reading second NI register.

STATUS2-WB-ADDR NI second status register consists of 6 status variables (note that v is the number of VCs per router port and vw is the binary size of v which is equal to $\log 2(v)$):

Bit	2	1	0
	got-pck-int-en	save-done-int-en	send-done-int-en
Read/Write	RW	RW	RW
Initial Value	0	0	0

Bit	5	4	3
	got-pck-isr	save-done-isr	send-done-isr
Read/Write	RW	RW	RW
Initial Value	0	0	0

Bit	$2vw$+v+5 vw+v+6	vw+v+5 v+6	v+5 6
	send-enable-binary	receive-enable-binary	crc-miss-match
Read/Write	R	R	R
Initial Value	0	0	0

send-done-int-en: It is a one-bit register which is used by the user to enable the assertion of interrupt signal when any send-VC finishes sending a packet to the NoC. the send-done interrupt is enabled if this register set as one.

save-done-int-en: It is a one-bit register which is used by the user to enable the assertion of interrupt signal when any VC receive-DMA finishes saving a packet to the memory. The save-done interrupt is enabled if this register set as one.

got-pck-int-en It is a one-bit register which is used by the user to enable the assertion of interrupt signal when any input VC receives a packet header flit. The got-packet interrupt is enabled if this register set as one.

send-done-isr Send-done interrupt service routing is a one-bit register that is used for acknowledging the send-done interrupt. This interrupt is acknowledged by writing logic one on this register.

save-done-isr Save-done interrupt service routing is a one-bit register that is used for acknowledging the save-done interrupt. This interrupt is acknowledged by writing logic one on this register.

got-pck-isr Got-packet interrupt service routing is a one-bit register that is used for acknowledging got-pck interrupt. This interrupt is acknowledged by writing logic one on this register.

crc-miss-match Is a v-bit register and indicates if the received packet CRC32 (attached to the tail flit) matches with the generated CRC32 by NI. Note that this register only available if the CRC generation is enabled in NI's HDL code.

receive-enable-binary This register holds the binary number of the VC which is writing the packet data into the memory in the current time.

send-enable-binary This register holds the binary number of the VC which is writing the packet data into the NoC in the current time slot.

BURST-SIZE-WB-ADDR The send-DMA burst-size register. When numbers of continuous sent flits becomes equal to the burst size, the enabled VC's send-DMA is disabled for one clock cycle. The VC will assert a request to the send-round robin arbiter for accessing to the WB interface in next clock cycle if it has not yet complete sending a packet. Disabling this send-DMA, allows the WB master interface to be accessed by other active master interfaces including other VCs which have data to be transferred. Setting smaller values prevents the enabled VC that is sending a long packet (assume there is no contention to block its resources) to hold the control of WB interface for a long time. The size of this register is defined by setting MAX-BURST-SIZE parameter in Verilog file. Setting a new burst size is only allowed when there is not any active VC in the NI.

Bit	log2(MAX-BURST-SIZE)-1 ... 0
	BURST-SIZE-WB-ADDR
Read/Write	W
Initial Value	All valid bits set as one

SEND-DATA-SIZE-WB-ADDR This register holds the size of data to be sent in words. Each VC has its own send-data-size register. The size of this register is defined using MAX-TRANSACTION-WIDTH Verilog parameter. Writing on this register is only permitted when its respective VC's send-DMA is in its ideal state (it is not busy).

Bit	MAX-TRANSACTION-WIDTH-1 ... 0
	SEND-DATA-SIZE-WB-ADDR
Read/Write	W
Initial Value	0

SEND-STRT-WB-ADDR This is the address pointer to the start location of the packet to be sent in the memory. The address must be given in byte (not words). The size of this register is equal to wishbone bus data width (Dw). Writing on this register is only permitted when its respective VC's send-DMA is in its ideal state (it is not busy).

Bit	Dw-1 … 0
	SEND-STRT-WB-ADDR
Read/Write	W
Initial Value	0

SEND-DEST-WB-ADDR This register holds the send-packet's header flit information including the message class number and an 8-bit register which indicates the destination router address. For 2D topologies i.e Mesh and Torus, the IP address is a set of two 4-bit registers (*dest-x* and *dest-y*) which are the x-y coordinates of the destination IP's router. Writing on this register is only permitted when the VC's send-DMA is in its ideal state (it is not busy).

Note that the state of VC's send-DMA changes from ideal to busy as soon as writing on this register. Hence, this register must be the last register to be set before sending a packet.

Bit	8+Cw … 8	7 … 4	3 … 0
	class	dest-y	dest-x
Read/Write	W	W	W
Initial Value	0	0	0

SEND-CTRL-WB-ADDR The send-VC (output VC) control register; it is reserved for enabling future controlling signals. Optionally you can use this register for enabling the sending process instead of using SEND-DEST-WB-ADDR register by editing *ni_vc_wb_slave_regs.v* as follows:

```
SEND_DEST_WB_ADDR: begin
    if (send_fsm_is_ideal) begin
            dest_x_next = s_dat_i[Xw-1     : 0];
        dest_y_next = s_dat_i [(DST_ADR_HDR_WIDTH/2)+Yw-1     :
            DST_ADR_HDR_WIDTH/2];
        pck_class_next= s_dat_i[Cw+DST_ADR_HDR_WIDTH-1    :
            DST_ADR_HDR_WIDTH];
        //send_start = 1'b1; ------ comment this line -----
    end
    end //SEND_DEST_WB_ADDR
SEND_CTRL_WB_ADDR: begin
    if (send_fsm_is_ideal) begin
        send_start = 1'b1; // ----- add it here -----
    end
end    // SEND_CTRL_WB_ADDR
```

RECEIVE-DATA-SIZE-WB-ADDR

This register holds the size of last received packet (for each VC) in words. The read value is only valid after the packet has been saved in buffer.

Bit	MAX-TRANSACTION-WIDTH-1 ... 0
	RECEIVE-DATA-SIZE-WB-ADDR
Read/Write	R
Initial Value	0

RECEIVE-STRT-WB-ADDR

This is the address pointer to the start location of the memory where the newly arrived packet must be stored by NI. The address must be decoded in byte (not words). The size of this register is equal to wishbone bus data width (Dw). Writing on this register is only permitted when the VC's receive-DMA is in its ideal state (it is not busy).

Bit	Dw-1 ... 0
	RECEIVE-STRT-WB-ADDR
Read/Write	W
Initial Value	0

RECEIVE-SRC-WB-ADDR

This register consists of received packet message class number and its source router address. The source router address is an 8-bit register which indicates the address of the router that had sent the received VC's packet. For 2D topologies i.e Mesh and Torus, this address consists of two 4-bit registers (*dest-x* and *dest-y*) which are the x-y coordinates of the source IP's router.

Bit	$8+Cw$ ... 8	7 ... 4	3 ... 0
	class	source-y	source-x
Read/Write	W	W	W
Initial Value	0	0	0

RECEIVE-CTRL-WB-ADDR

The receive-VC (input VC) control register. Writing on this register is only permitted when the VC's receive-DMA is in its ideal state (it is not busy). By Writing any value on this register the receive-vc-packet-is-saved is reset and one of the following two conditions will happen:

1. If a packet exists in a router VC's input buffer, the VC's receive-DMA starts saving that packet into the memory.
2. In case that the router VC's buffer is empty (no packet has arrived yet) it configures the VC's receive-DMA to auto-save the next incoming packet.

RECEIVE-MAX-BUFF-SIZ

The allocated receive-memory buffer size in words. Setting this register before saving a packet is required to avoid memory crashes due to overflow. In case that the size of received packet is larger than the allocated memory size, the rest of that packet will be discarded.

Bit	MAX-TRANSACTION-WIDTH-1 ... 0
	RECEIVE-MAX-BUFF-SIZ
Read/Write	W
Initial Value	0

FPGA Implementation Results

Table 3 shows the NI_master hardware implementation results on Altera Cyclone IVE EP4CE115F29C7 FPGA for different number of VCs and with/without CRC32 enable. All other NI parameters were set as follows: Payload size of 32-bit, buffer size of 4-flit per VC, burst size of 32-flit and MAX-TRANSACTION-WIDTH of 13-bit.

Table 3: Hardware implementation results on Altera Cyclone IVE EP4CE115F29C7 FPGA device

# VC	CRC_EN	LUC	Memory(bit)
-	"NO"	349 / 114,480 (<1 %)	136 / 3,981,312 (<1 %)
2	"NO"	704 / 114,480 (<1 %)	272 / 3,981,312 (<1 %)
4	"NO"	1,332 / 114,480 (1 %)	544 / 3,981,312 (<1 %)
-	"YES"	527 / 114,480 (<1 %)	136 / 3,981,312 (<1 %)
2	"YES"	909 / 114,480 (<1 %)	272 / 3,981,312 (<1 %)
4	"YES"	1,906 / 114,480 (2 %)	544 / 3,981,312 (<1 %)

The send-packets simulation waveform

Figure 2 shows how the NI output bandwidth is shared between different number of active VCs. This result is obtained for NoC router with four VCs per port and 32-bit payload size. Each VC sent long packets (512 bytes) to the NoC. The bit 35^{th}-32^{nd} of output flit is the VC ID represented in one-hot code. Hence, the asserted bit in this range shows which VC is enabled and is sending its flits to the NoC. As shown in this Figure the NI output bandwidth is equally shared between all active send-VCs.

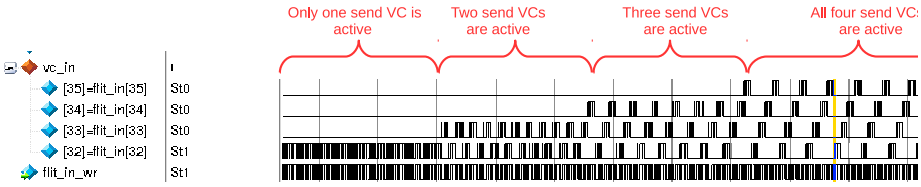

Figure 2: NI-master packet sending simulation waveform. The Burst size has been set as 32 and there were only one active IP sending packets to the network.

NI_slave

NI_slave is a Wishbone bus (WB)-based interface for the network-on-chip (ProNoC) router. This module is an extension of NI_master module connected to two input and output buffers. There are three WB slave interfaces in this module, one for writing on output buffer, one for reading input buffer and one for controlling the NI.

NI-slave Block diagram:

Figure3 depicts the functional block diagram of NI-slave module.

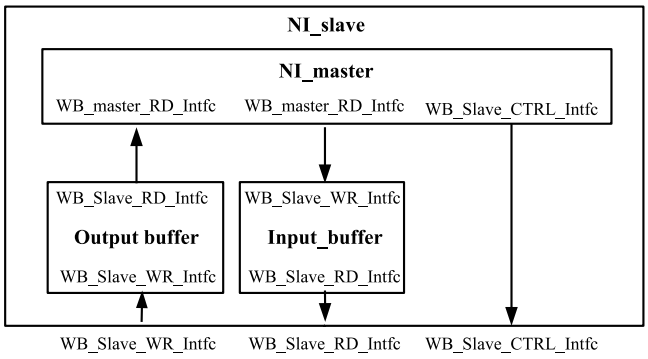

Figure 3: NI_master functional block diagram

Parameters Description

The NI_slave share all NI_master parameters described in Table 1 as well as two additional parameters for setting IO buffers' width:

Table 4: NI_master GUI Parameters.

Name	Description	Permitted values	Default
INPUT_MEM_Aw	This parameter defines the input buffer memory address width in bits. The actual memory size is $2^{INPUT\_MEM\_Aw}$ in words	$n \in \mathbb{N}$, $4 \leqslant n \leqslant 32$	10
OUTPUT_MEM_Aw	This parameter defines the output buffer memory address width in bits. The actual memory size is $2^{OUTPUT\_MEM\_Aw}$ in words	$n \in \mathbb{N}$, $4 \leqslant n \leqslant 32$	10