NOC

Introduction:

Interconnection networks are used to interconnect either single or sets of components within a com puter system, or multiple computer systems; the interconnection aspect of computer architecture has gained significant importance in recent years, with regard to the impact of different solutions on the overall performance and cost of the whole system.
One of the main networking domains, and the subject of this project, is Network-on-Chip (NoC), in which the interconnection network is used for interconnecting micro-architecture functional units, reg ister files, caches, compute tiles and processor and IP cores within chips or multi-chip modules; current and near future NoCs support the connection of a few tens to a few hundreds of such devices with a maximum interconnection distance on the order of centimeters.

Specifications of design :-

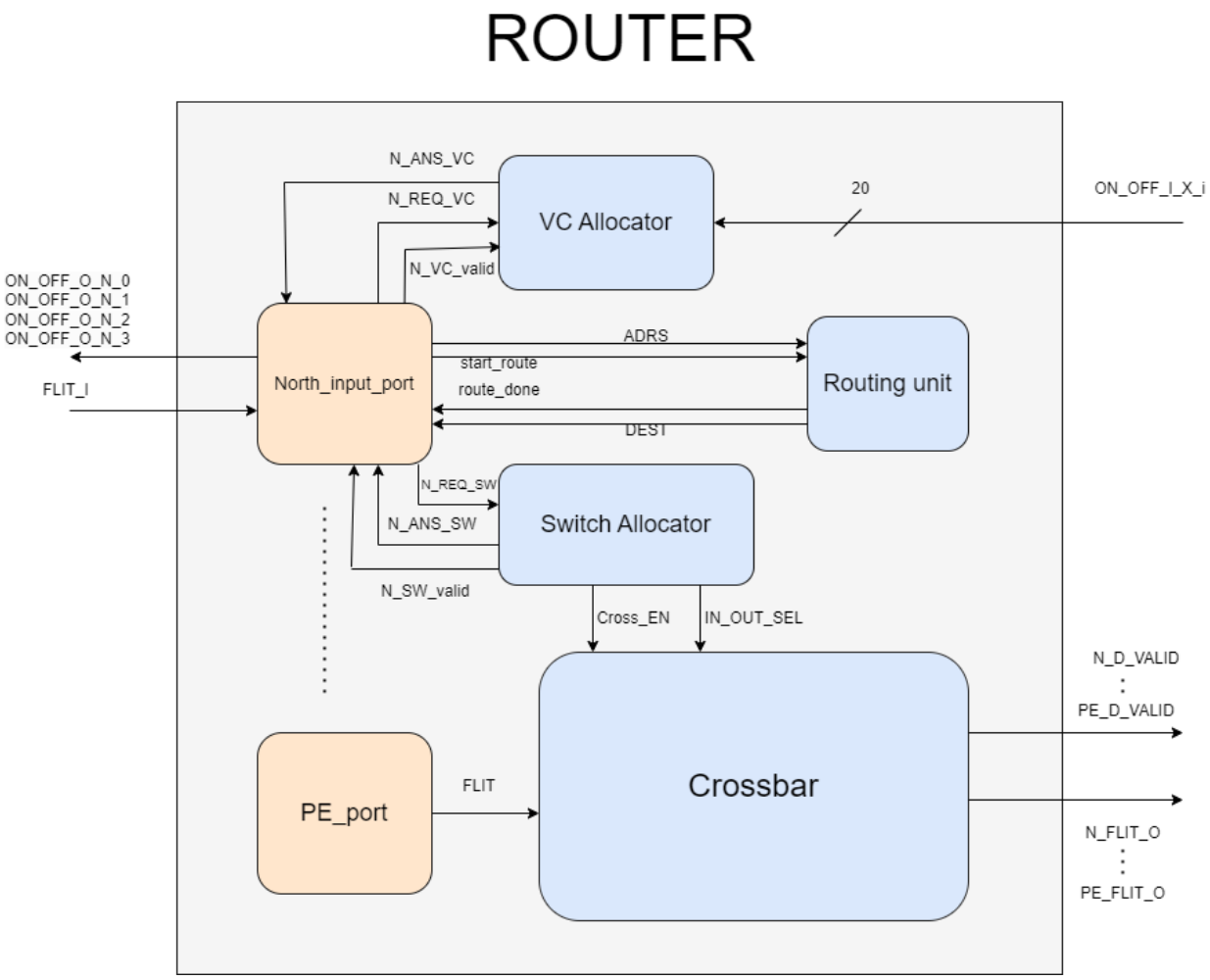
- A Network-on-Chip interconnection module has been developed, with a **2D mesh topology (3X3mesh).**

- The routers allocate data at **flit granularity**, implementing a **wormhole switching architecture** further optimized by the presence of multiple virtual channels per input, avoiding the Head of Line blocking issue and thus allowing a higher average throughput to the network.
- Packet routing is driven by **[Node-based Routing algorithm with West-first Turn Model technique]**, independently computed by each router belonging to the mesh.

- Flow control is implemented **in the switching activity management**, which is controlled by a per-router switch allocation unit, and uses the **On-Off algorithm**, easy to implement but efficient enough for a **medium level of traffic** in the network.

Design and implementation

- Router Block diagram



The router contains 5 blocks:

- INPUT_PORTS module consists of five input ports: N,S,E,W,PE.
- Routing_unit module.
- VC_Allocator module.
- Switch_allocator module.
- Crossbar module.

Packet configurations: -

- Maximum number of packets is 8. [sequence number 3-bit]
- Each packet contains 8 flits.

VC identifier [2-bit]	Type [2-bit]	Sequence number [3-bit]	Data [9-bit]

- Sequence number: number of the packet which the flit belongs to.

- VC identifier: number of the VC.
  - VC0
  - VC1
  - VC2
  - VC3

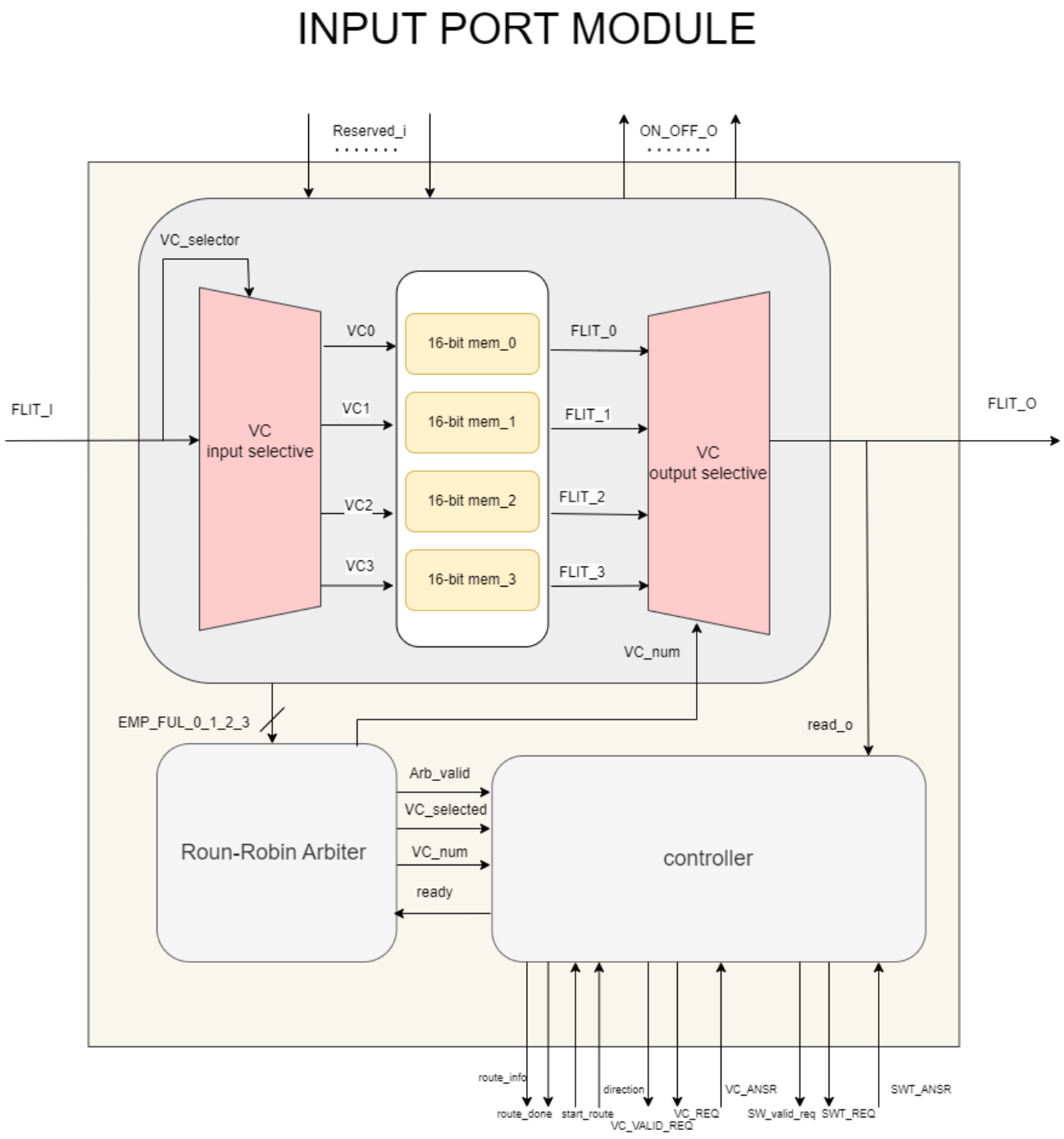
- Type: the type of carried date on the flit {Header, Body, Tail}.
  - Header = 2'b00
  - Body = 2'b01
  - Tail = 2'b10

Input Port module.

- is responsible for storing the flits that comes from its input interface matching the correspondent **virtual channel id.**

- One router has exactly 5x input modules, each input module has 4 **virtual channels.**

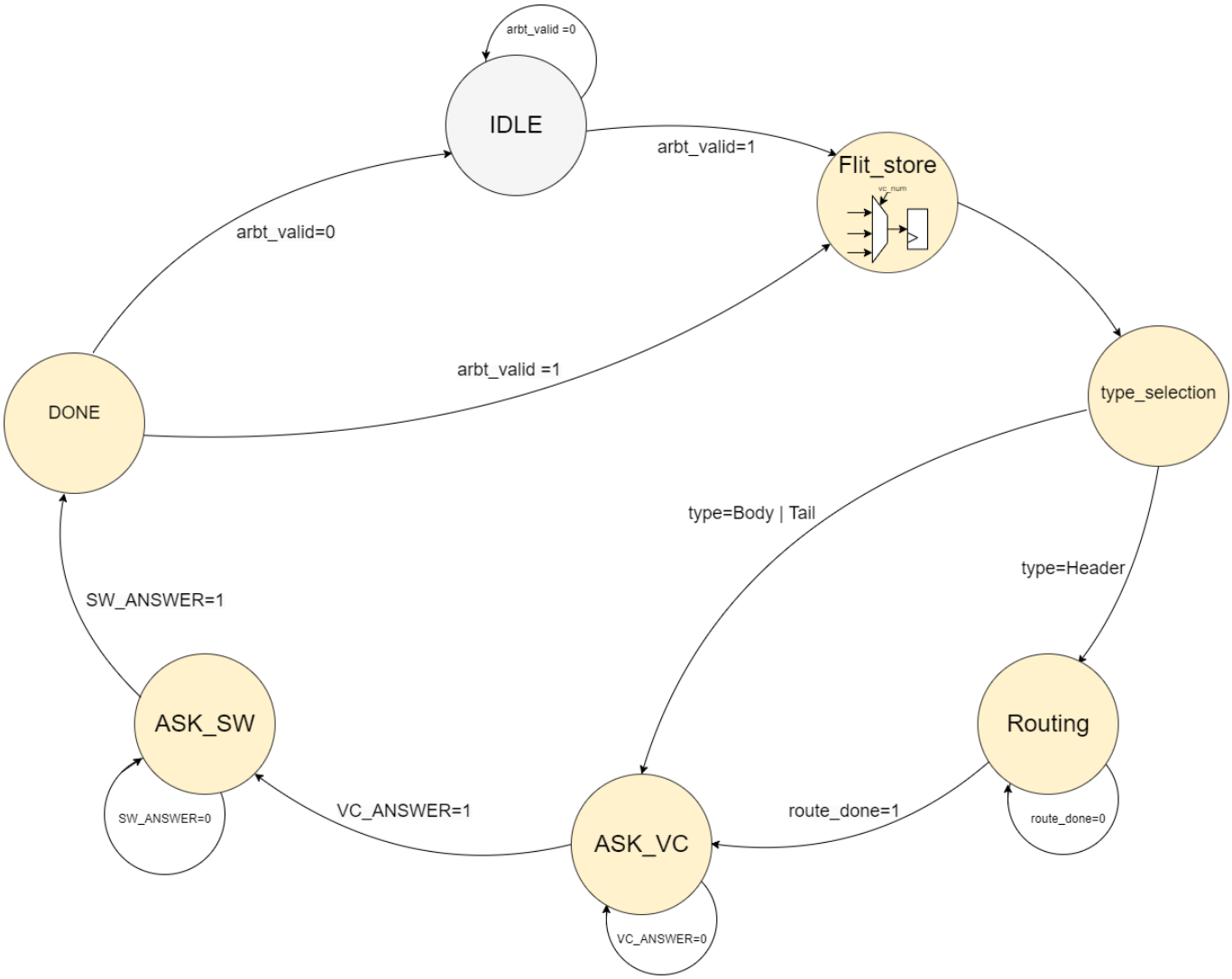
- Each input port has Roun-Robin Arbiter to choose between VCs.



Signals description:-

Pot	width	direction	description
FLIT_I	16-bit	input	The incoming flit from the upstream router.
direction	3-bit	input	The direction which Computed by the routing unit.
VC_ANSR	1-bit	input	The evaluated signal after the VC allocation is done.
SWT_ANSR	1-bit	input	The evaluated signal after the switch allocation is done.
Route_info	4-bit	output	It contains destination node data from the incoming flit going to routing unit to compute the next direction
VC_REQ	5-bit	output	{evaluated output port , VC number} going to VC allocator to check the availability of VCs in the downstream router.
VC_valid_req	1-bit	output	This is a control signal for the VC_ALLOCATOR
SWT_REQ	3-bit	output	the evaluated output port going to switch allocator to check the availability of this output port in the current router.
SW_valid_req	1-bit	output	This is a control signal for the switch_ALLOCATOR
ON_OFF_0_1_2_3	1-bit	output	Control signals for the upstream VC allocator High when there is a valid VC and low when the VC is reserved
Reserved_0_1_2_3	1-bit	input	A control signal from the upstream vc allocator to reserve the valid VC.
FLIT_O	16-bit	output	The output flit from the input port I to the cross bar.

Input controller



IDLE state: it is the default state & reset state of the finite state machine.

Ready	This signal is high at the IDLE state and when there is no operations in the controller	Output to Arbiter

Flit_store state: this state responsible for storing the selected flit into a register.

Type_selection state: this state selects the type-bits then deside,

If (type-bits = header):

- It will send the address-bits stored in the register to the Routing unit.

If (type-bits = body or tail):

-

- It will take the routing information which stored while Routing state, and Sending the request to the vc_allocator.

Route_info	It contains the routing informatin in the flit	Out to Routing unit
Start_route	This signal is high for one clock cycle when sending the routing information and low otherwise	Out to Routing unit
Vc_request	It contains the output port calculated from the routing unit and the vc number {out_port,vc_numb}	Out to VC Allocator
valid_vc_req	This signal is high only for one clock cycle while sending the Vc_request	Out to VC Allocator

Routing state: this state wait for the routing unite output, then store the calculated direction for the flit labeled by it's sequence number.

Vc_request	It contains the calculated output port and the VC number {out_port,vc_numb}	Out to VC Allocator
valid_vc_req	This signal is high only for one clock cycle while sending the Vc_request	Out to VC Allocator

ASK_VC state: it is responsible for waiting the answer of the VC Allocator and sending the request to the switch allocator after the VC Allocator is done.

Vc_request	This signal will be in the same data until the VC answer is positive	Out to VC Allocator
Sw_request	It contains the calculated output port and the VC number {out_port,vc_numb}	Out to switch Allocator
Valid_sw_req	This signal is high only for one clock cycle while sending the sw_request	Out to switch Allocator

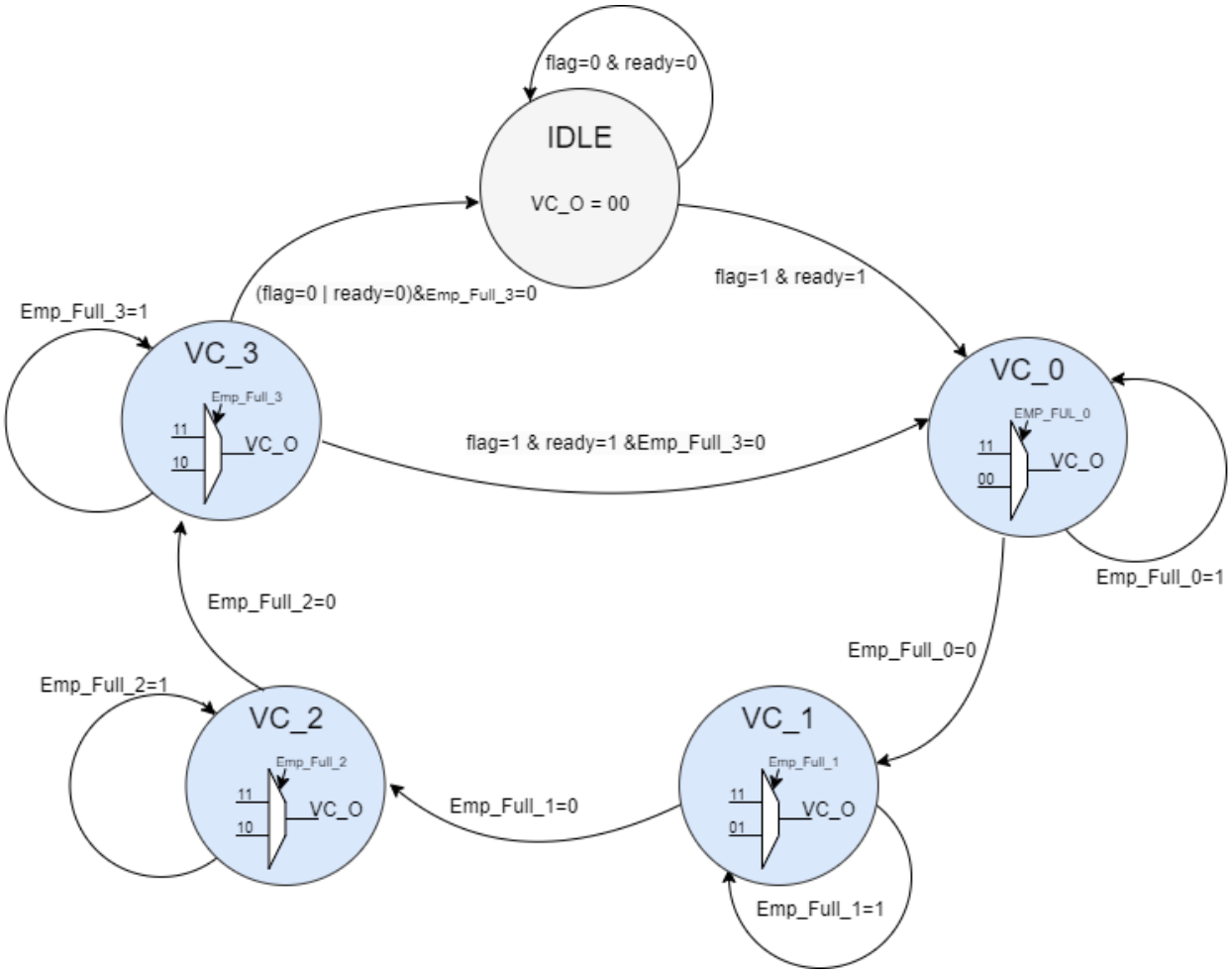
ASK_SW state: this state also waits for the answer of the switch allocator to select the output VC .

Sw_request	It contains the calculated output port and the VC number {out_port,vc_numb}	Out to switch Allocator

DONE state: it is responsible for making the ready signal high and check if the Arbiter ready or not.

Ready	This signal is high at the IDLE state and when there is no operations in the controller	Output to Arbiter
clear	This is high for one clock to clear the vc and deactivate the empty/full signal and activate the on/off signal	Out to input buffer

Round-Robin Arbiter for input Port



- Flag signal consists of {EMP_FULL_0, EMP_FULL_1, EMP_FULL_2, EMP_FULL_3}.
- Ready signal is a control signal coming from the controller.
- VC_O signal carries the VC number to the controller.

IDLE state: the default state & reset state of the finite state machine.

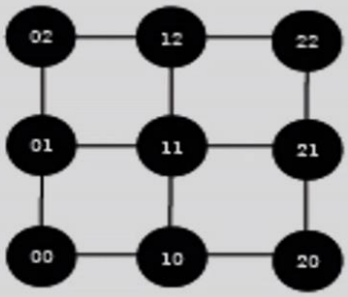
VC_x state: responsible for choosing the VC number x if it's VC is full and still in until the controller done of it.

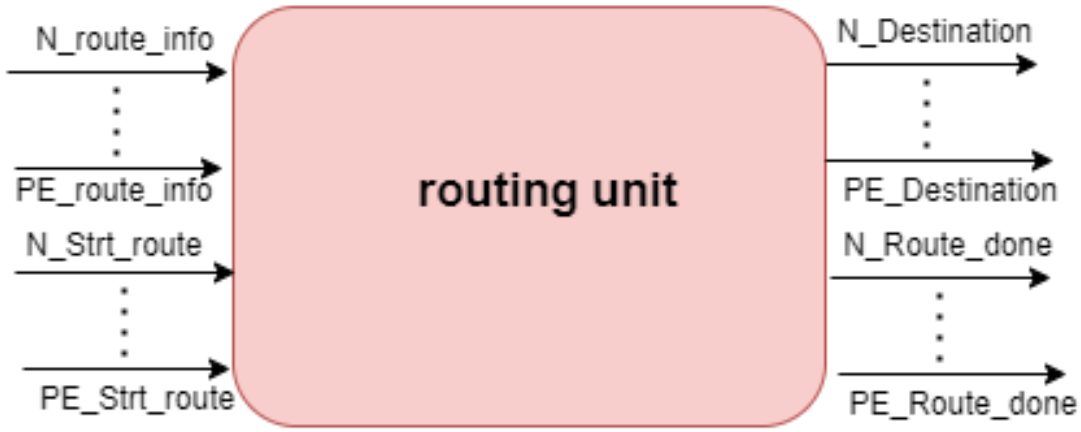
Route computation Unit

- the route computation unit computes the next hop for each packet as soon as the head flit arrives to the router and sends the result to the input port unit.
- the implemented routing algorithm **[Node-based Routing with West-first Turn Model technique]** requires only information about the position of the current router and of the destination router in the 2D mesh. As the table below.

Node-based routing table for a 3 × 3 mesh with West-First turn model

	00	01	02	10	11	12	20	21	22
00	X	N –	N –	E –	E N	E N	E –	E N	E N
01	S –	X	N –	E S	E –	E N	E S	E –	E N
02	S –	S –	X	E S	E S	E –	E S	E S	E –
10	W –	W –	W –	X	N –	N –	E –	E N	E N
11	W –	W –	W –	S –	X	N –	E S	E –	N E
12	W –	W –	W –	S –	S –	X	E S	E S	E –
20	W –	W –	W –	W –	W –	W –	X	N –	N –
21	W –	W –	W –	W –	W –	W –	S –	X	N –
22	W –	W –	W –	W –	W –	W –	S –	S –	X





Signals description:-

Port	width	direction	description
X_route_info	4-bit	input	It contains the destination node data from the incoming flit from the input port (X) to compute the next direction
X_Start_route	1-bit	input	This signal is high for one clock cycle when sending the routing information and low otherwise
X_ROUTE_Done	1-bit	output	This signal is high when finishing the routing computation and low otherwise.
X_Destination	4-bit	output	The direction of the packet.

(X) CORROSPONDING TO THE NAME OF THE INPUT PORT(N,S,W,E,PE)

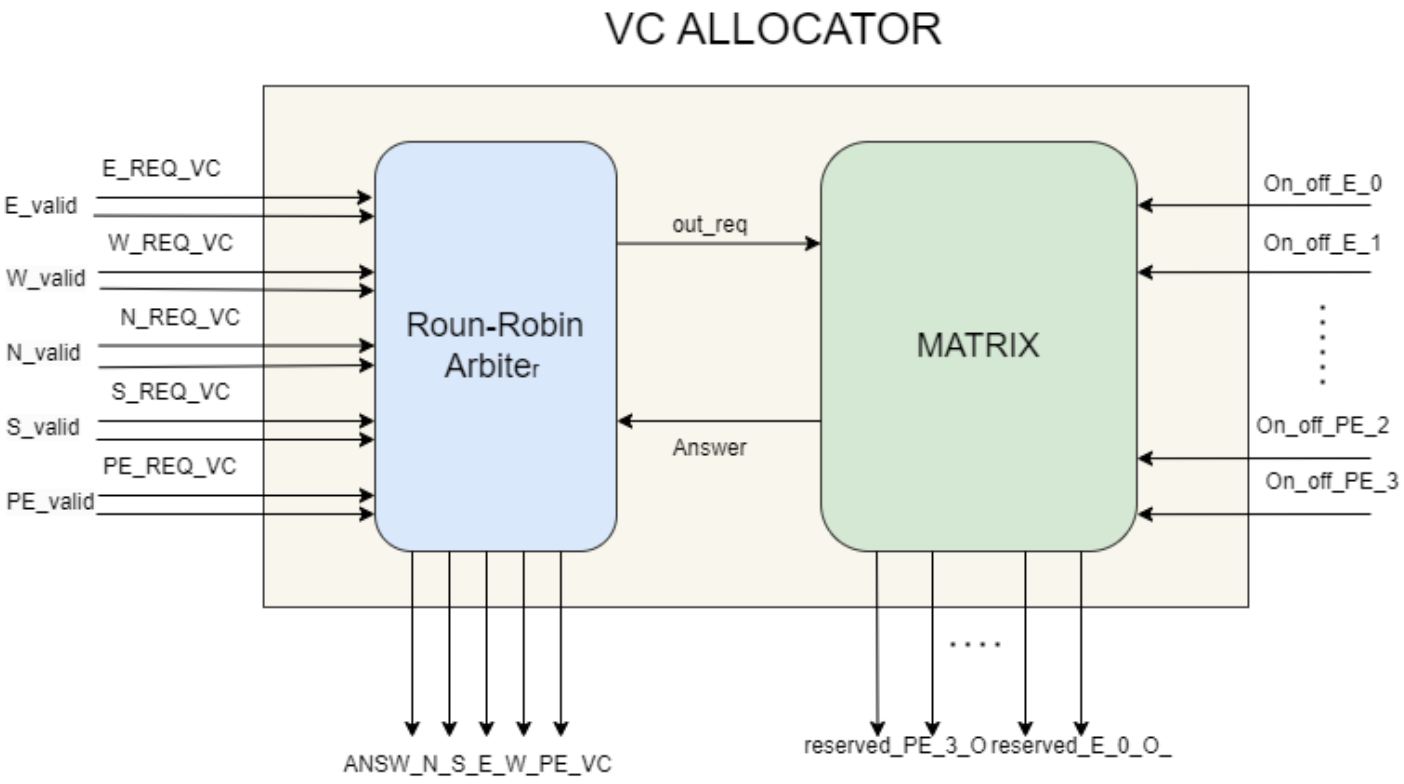
VC Allocator

It responsible for checking the availability of VCs in the downstream router.

- A separable input-first allocator is used inside the VC allocator unit.
- It is implementation by Applying a matrix contains all resources with respective to its availability.
- Round-Robin Arbiter to choose between requests.

The available resources matrix:

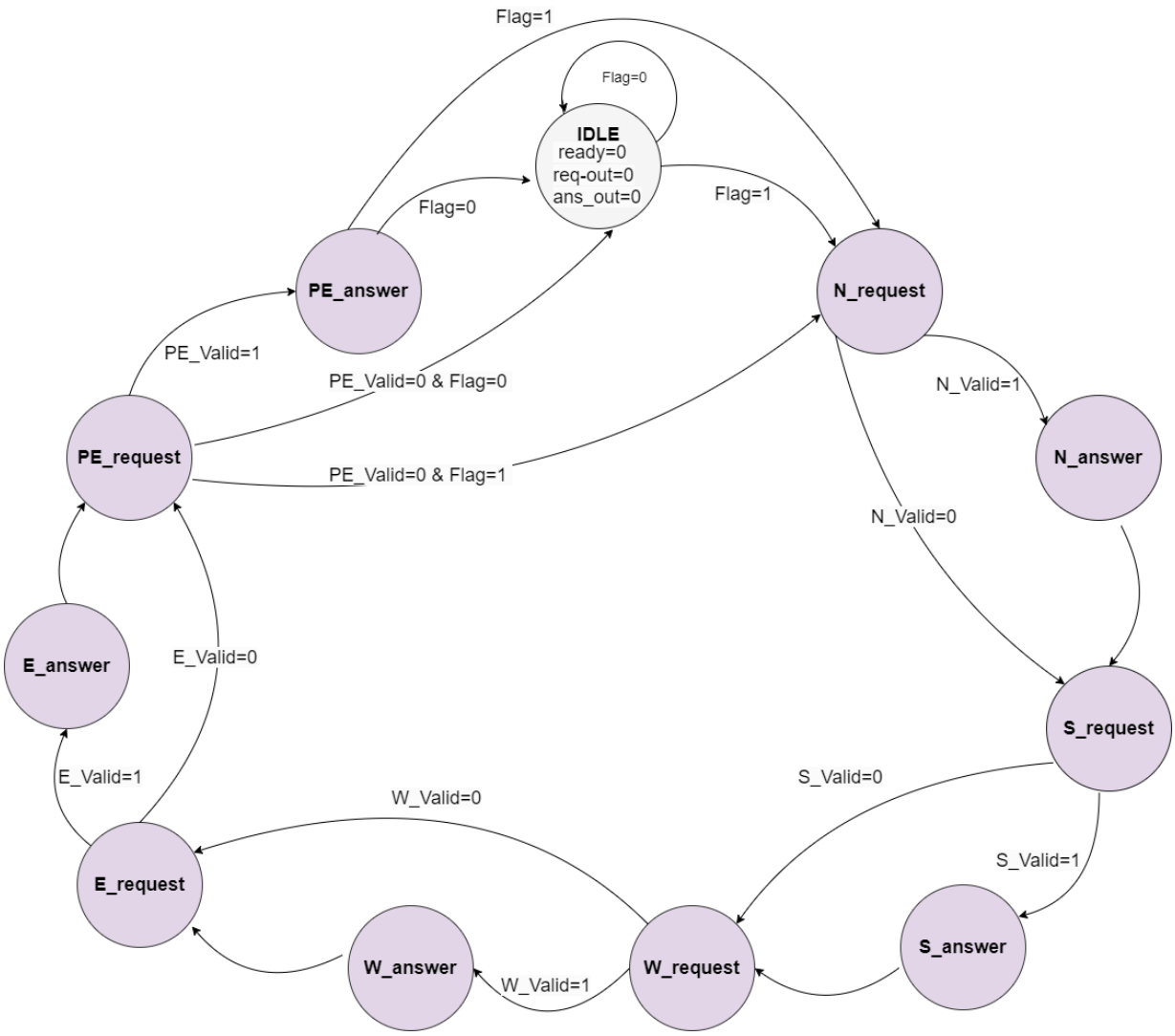
VCi\PORTS	North	South	East	Weast	PE
VC_0	ON_OFF_N_0	ON_OFF_S_0	ON_OFF_E_0	ON_OFF_W_0	ON_OFF_PE_0
VC_1	ON_OFF_N_1	ON_OFF_S_1	ON_OFF_E_1	ON_OFF_W_1	ON_OFF_PE_1
VC_2	ON_OFF_N_2	ON_OFF_S_2	ON_OFF_E_2	ON_OFF_W_2	ON_OFF_PE_2
VC_3	ON_OFF_N_3	ON_OFF_S_3	ON_OFF_E_3	ON_OFF_W_3	ON_OFF_PE_3



Signals description:-

Port	width	direction	description
E_REQ_VC	5-bit	input	East port request consists of {evaluated output port , VC number} coming to check the availability of VCs in the downstream router.
W_REQ_VC	5-bit	input	west port request consists of {evaluated output port , VC number} coming to check the availability of VCs in the downstream router.
N_REQ_VC	5-bit	input	north port request consists of {evaluated output port , VC number} coming to check the availability of VCs in the downstream router.
S_REQ_VC	5-bit	input	south port request consists of {evaluated output port , VC number} coming to check the availability of VCs in the downstream router.
PE_REQ_VC	5-bit	input	Processing Element port request consists of {evaluated output port , VC number} coming to check the availability of VCs in the downstream router.
On_off_x_i : On_off_PE_20	1-bit	input	A single bit signal coming from each VC in each input port of the downstream router, it represents it's VC empty or full if {zero} full if {1} Empty and ready to receive a flit.
ANSW_N_VC	1-bit	output	This signal represents the acceptance of the incoming request from the North input port.
ANSW_S_VC	1-bit	output	This signal represents the acceptance of the incoming request from the South input port.
ANSW_E_VC	1-bit	output	This signal represents the acceptance of the incoming request from the East input port.
ANSW_W_VC	1-bit	output	This signal represents the acceptance of the incoming request from the West input port.
ANSW_PE_VC	1-bit	output	This signal represents the acceptance of the incoming request from the PE input port.
Reserved_x_0 : Reserved_x_20	1-bit	output	This signal goes out to the downstream router to announce that a specific VC is reserved and it's on_off signal must be low

Round-Robin Arbiter for VC Allocator



_ Flag signal: consists of {N_Valid, S_Valid, E_Valid, W_Valid, PE_Valid,}.

_ req_out signal: the signal which carries the chosen request from it's state going out to the matrix.

_ ans_out signal : this signal carries the answer which comes from the matrix to get out of the VC Allocator module.

_ x_reserved_i_o signals: this signal is High if the answer=1 in it's port(x) state

And low other wise.

The request states: responsible for checking if there is a valid request and send it to the matrix.

req_out	6-bit	Carries the request bits	Out to the matrix

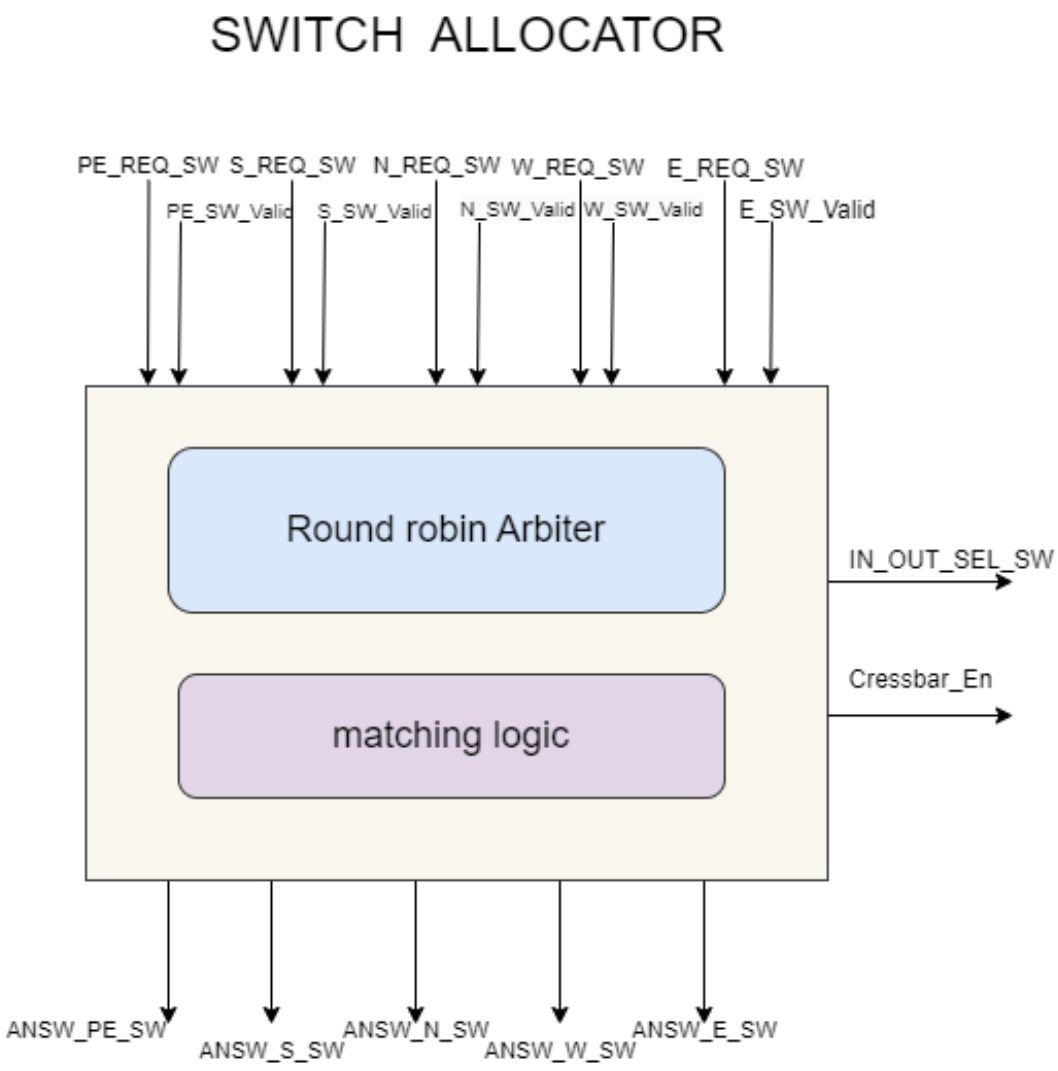
The answer states: responsible for capturing the answer from the matrix and make the ready signal low.

Ans_out	1-bit	Carries the answer captured from the matrix	Out to the INPUT Module
x_reserved_i_o	1-bit	It represents if the on_off signal of a specific VC is reserved or not.	Out to downstream input port

Switch Allocator

The switch allocator module solves the contention between input buffers for the access to the crossbar switch.

- By implementing a separable input-first allocator and thus sending control signals to the cross5 bar module and to the input buffers.
- And Round-Robin Arbiter to choose between the available output ports.

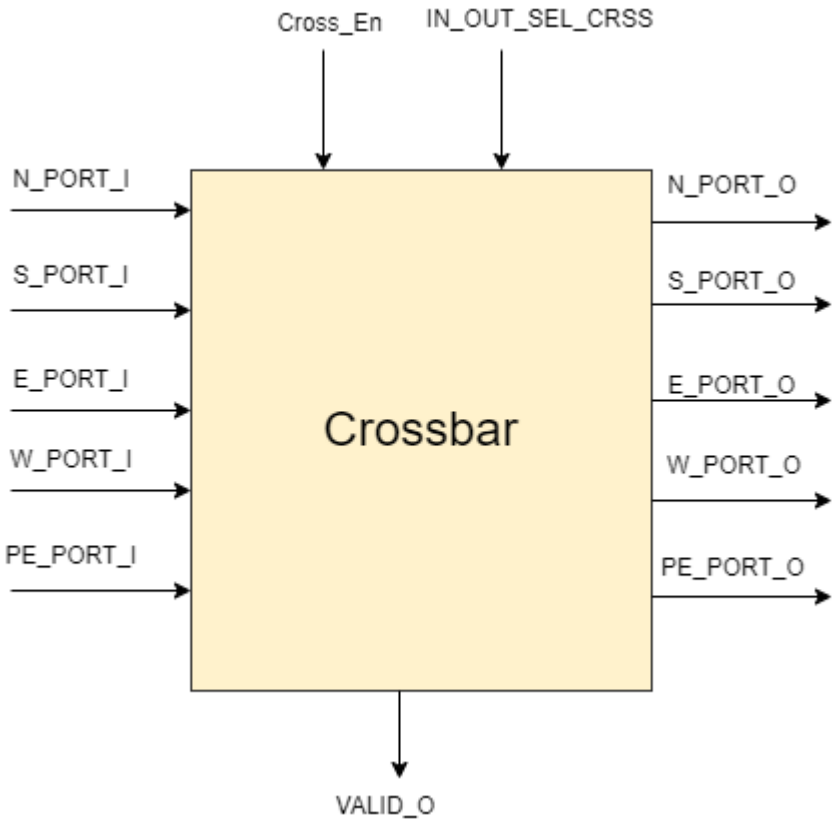


Signals description:-

Port	width	direction	description
E_REQ_SW	3-bits	input	East port request consists of {evaluated output port} coming to check the availability of this output port in the current router.
W_REQ_SW	3-bits	input	west port request consists of {evaluated output port} coming to check the availability of this output port in the current router.
N_REQ_SW	3-bits	input	north port request consists of {evaluated output port} coming to check the availability of this output port in the current router.
S_REQ_SW	3-bits	input	south port request consists of {evaluated output port} coming to check the availability of this output port in the current router.
PE_REQ_SW	3-bits	input	PE port request consists of {evaluated output port} coming to check the availability of this output port in the current router.
X_SW_Valid ⋮ X_SW_Valid	1-bit	input	The valid signal is high when the request ready for the switch and low when the answer of the switch is high.
ANSW_E_SW	1-bit	output	This signal represents the acceptance of the incoming request from the East input port.
ANSW_W_SW	1-bit	output	This signal represents the acceptance of the incoming request from the West input port.
ANSW_N_SW	1-bit	output	This signal represents the acceptance of the incoming request from the North input port.
ANSW_S_SW	1-bit	output	This signal represents the acceptance of the incoming request from the South input port.
ANSW_PE_SW	1-bit	output	This signal represents the acceptance of the incoming request from the PE input port.
IN_OUT_SEL_SW	6-bit	output	The signal which control the cross bar it's consists of {input port , output port }
Crossbar_En	1-bit	output	This enable signal is high when IN_OUT_SEL_SW signal is ready and low otherwise

Crossbar

Cross bar module allows moving flits from input Block to output ports allowing only one input to propagate at a time. And input and output selection are done through switch allocator.

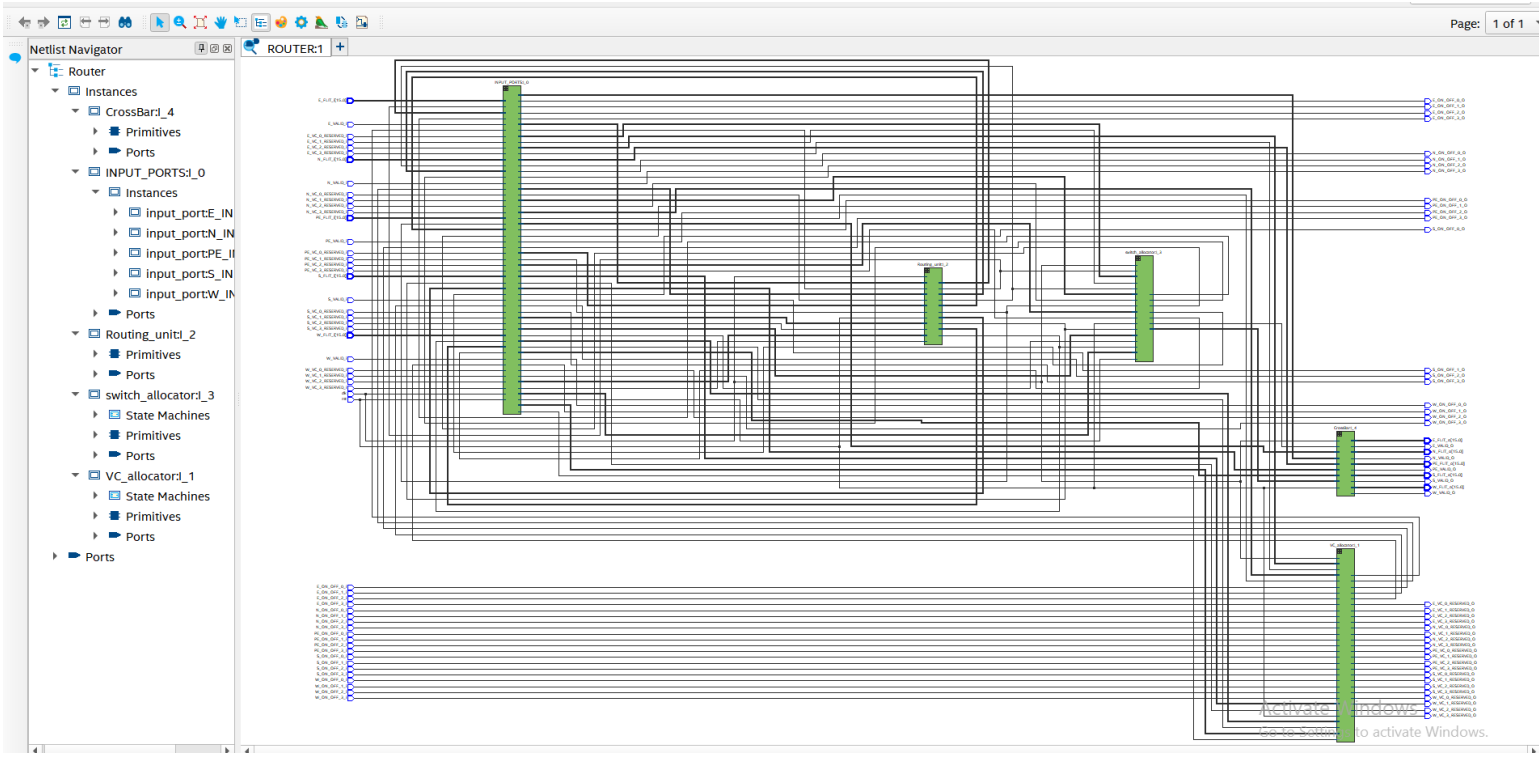


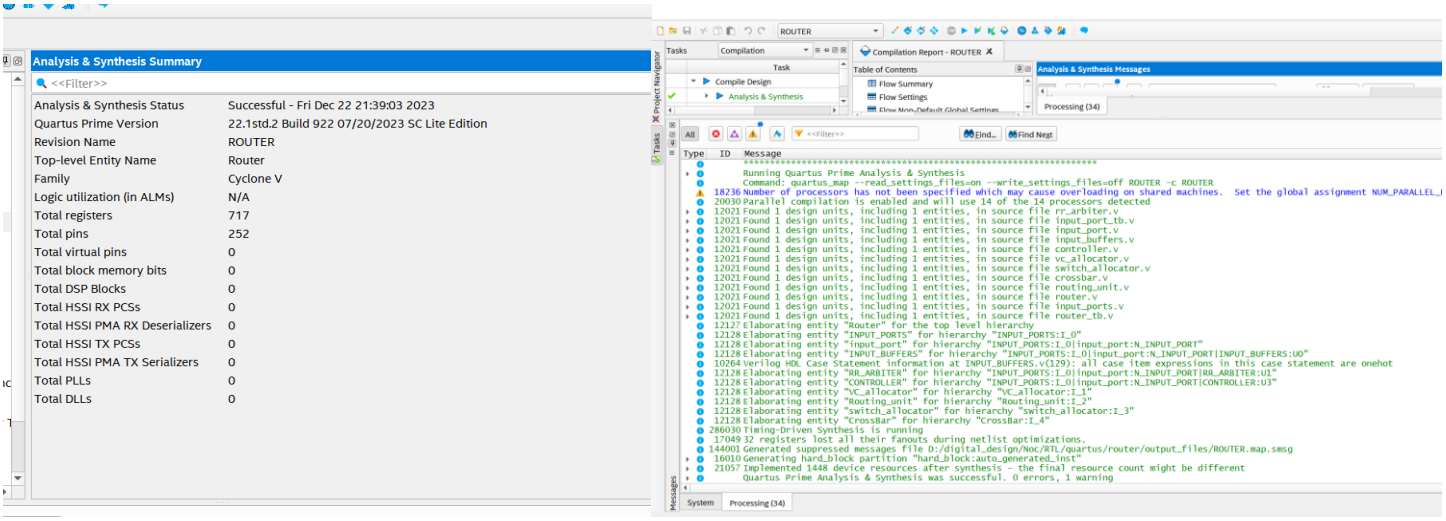
Signals description:-

Port	width	direction	description
IN_OUT_SEL_CRSS	8-bit	input	The signal which control the cross bar it's consists of {input port , output port }
N_PORT_I	16-bit	input	Input ports
S_PORT_I	16-bit	input	
E_PORT_I	16-bit	input	
W_PORT_I	16-bit	input	
PE_PORT_I	16-bit	input	

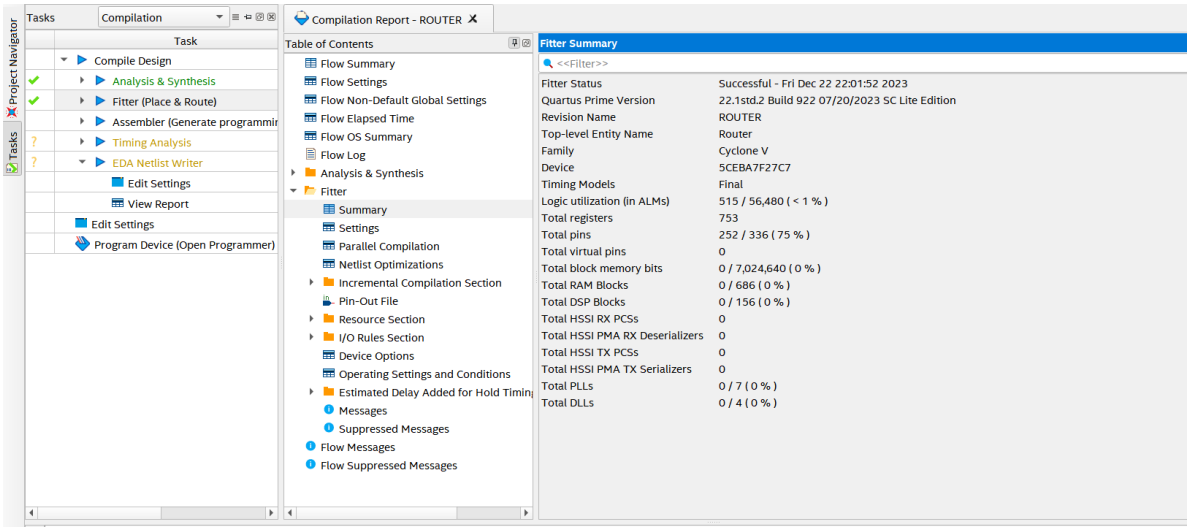
Cross_En	1-bit	input	A control signal for the crossbar from the switch allocator to take the data from input.
N_PORT_O	16-bit	output	Output ports
S_PORT_O	16-bit	output	
E_PORT_O	16-bit	output	
W_PORT_O	16-bit	output	
PE_PORT_I	16-bit	output	
Valid_O	1-bit	output	This signal is high when the data ready at the crossbar to go to the downstream router

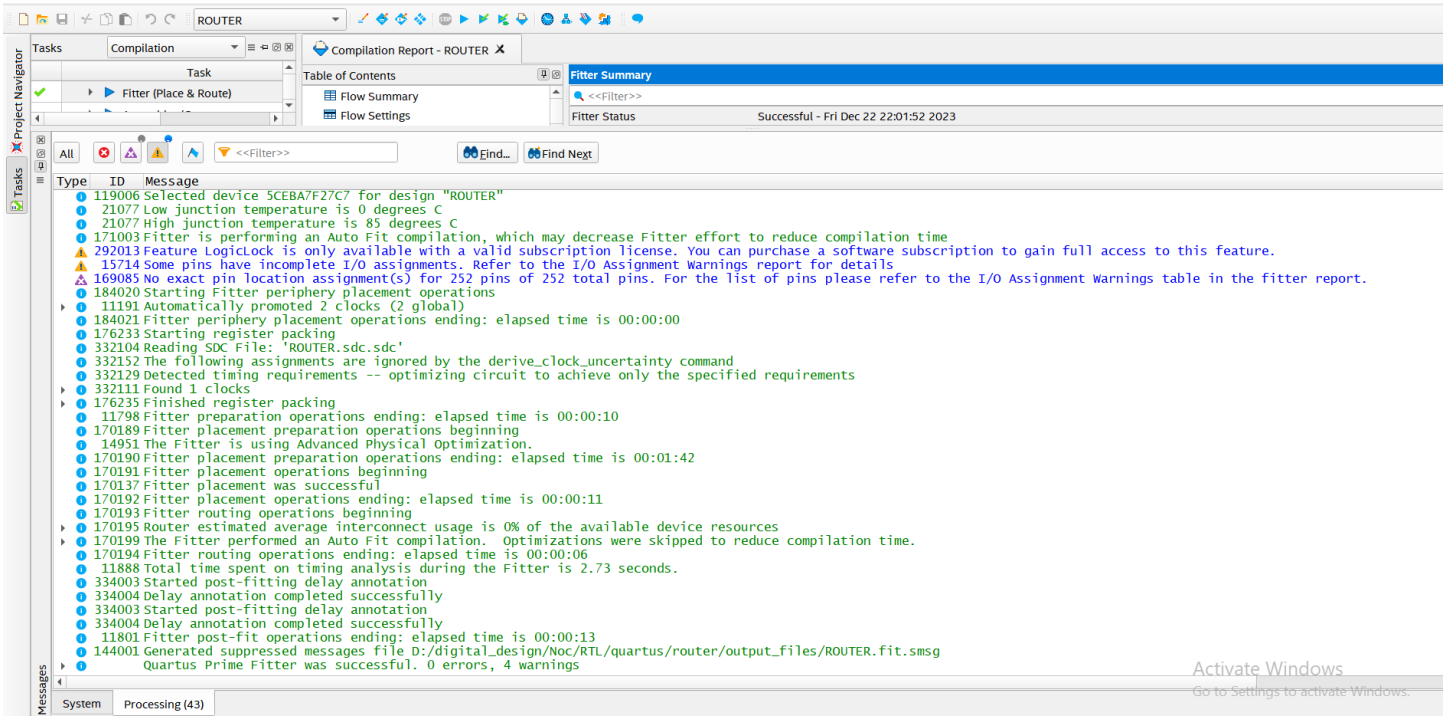
Synthesis & RTL Netlist for Router





Filter (place&route)





Timing analysis

_ clock period 7.2n

_ No setup violation at all corners

Slow 1100mV 0C Model Setup Summary

<<Filter>>

	Clock	Slack	End Point TNS
1	clk	0.465	0.000

Fast 1100mV 0C Model Setup Summary

<<Filter>>

	Clock	Slack	End Point TNS
1	clk	4.168	0.000

Fast 1100mV 85C Model Setup Summary

<<Filter>>

	Clock	Slack	End Point TNS
1	clk	3.885	0.000

__ No Hold violation at all corners

Slow 1100mV 85C Model Hold Summary

<<Filter>>

	Clock	Slack	End Point TNS
1	clk	0.361	0.000

Fast 1100mV 0C Model Hold Summary

<<Filter>>

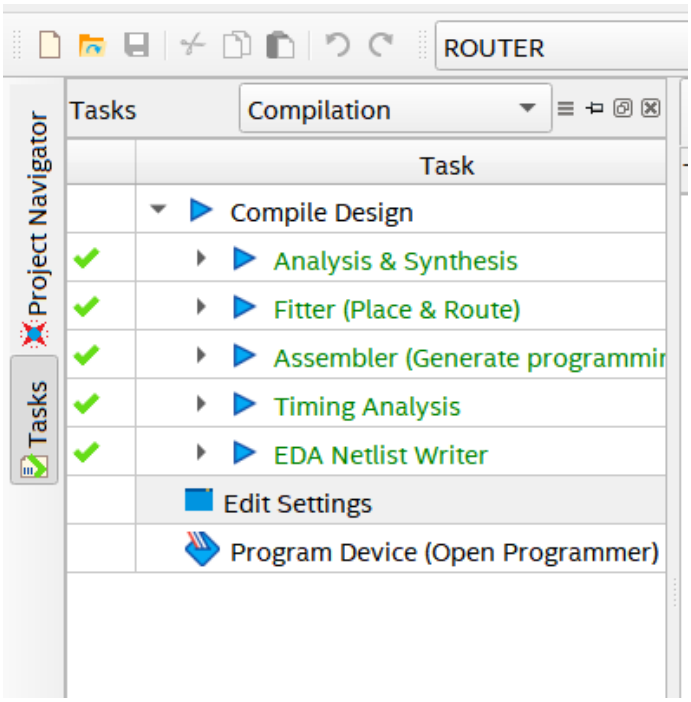
	Clock	Slack	End Point TNS
1	clk	0.162	0.000

Multicorner Timing Analysis Summary

<<Filter>>

	Clock	Setup	Hold	Recovery	Removal	Minimum Pulse Width
1	Worst-case Slack	0.465	0.162	N/A	N/A	2.837
1	clk	0.465	0.162	N/A	N/A	2.837
2	Design-wide TNS	0.0	0.0	0.0	0.0	0.0
1	clk	0.000	0.000	N/A	N/A	0.000

Router Flow is completed



Router simulation

Router simulation

1- When 5 input for 5 different input ports to five different output ports:

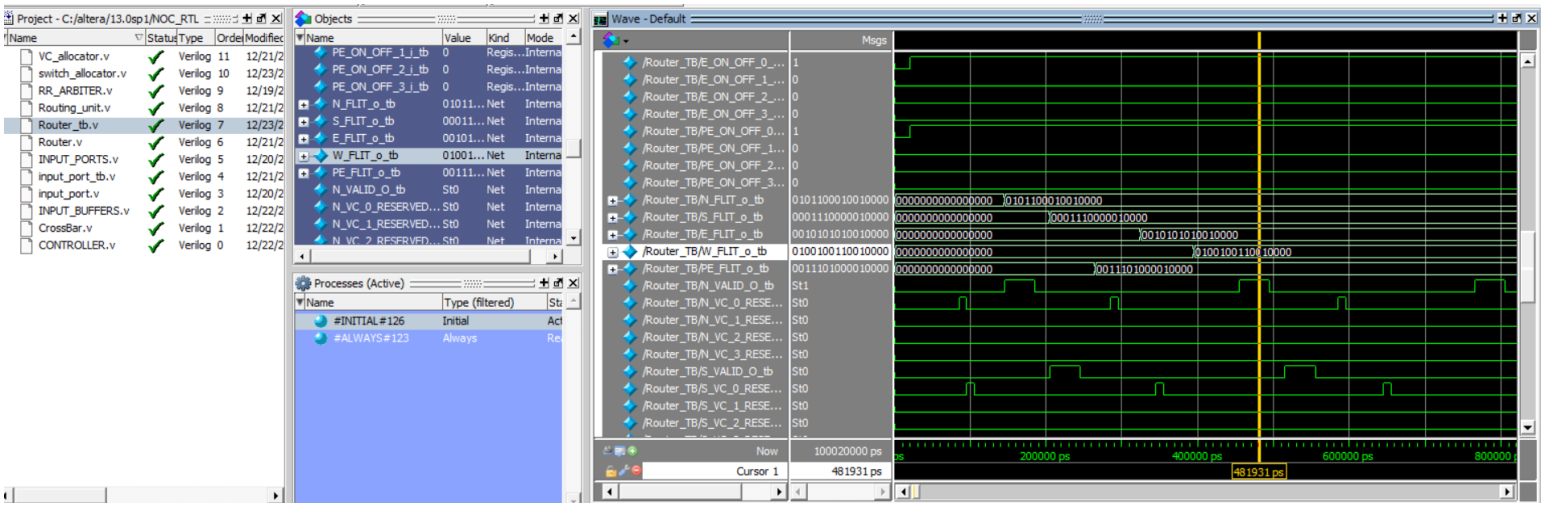


Fig. five inputs to different output ports.

2- When 2 different flits on 2 different input ports to same output ports:

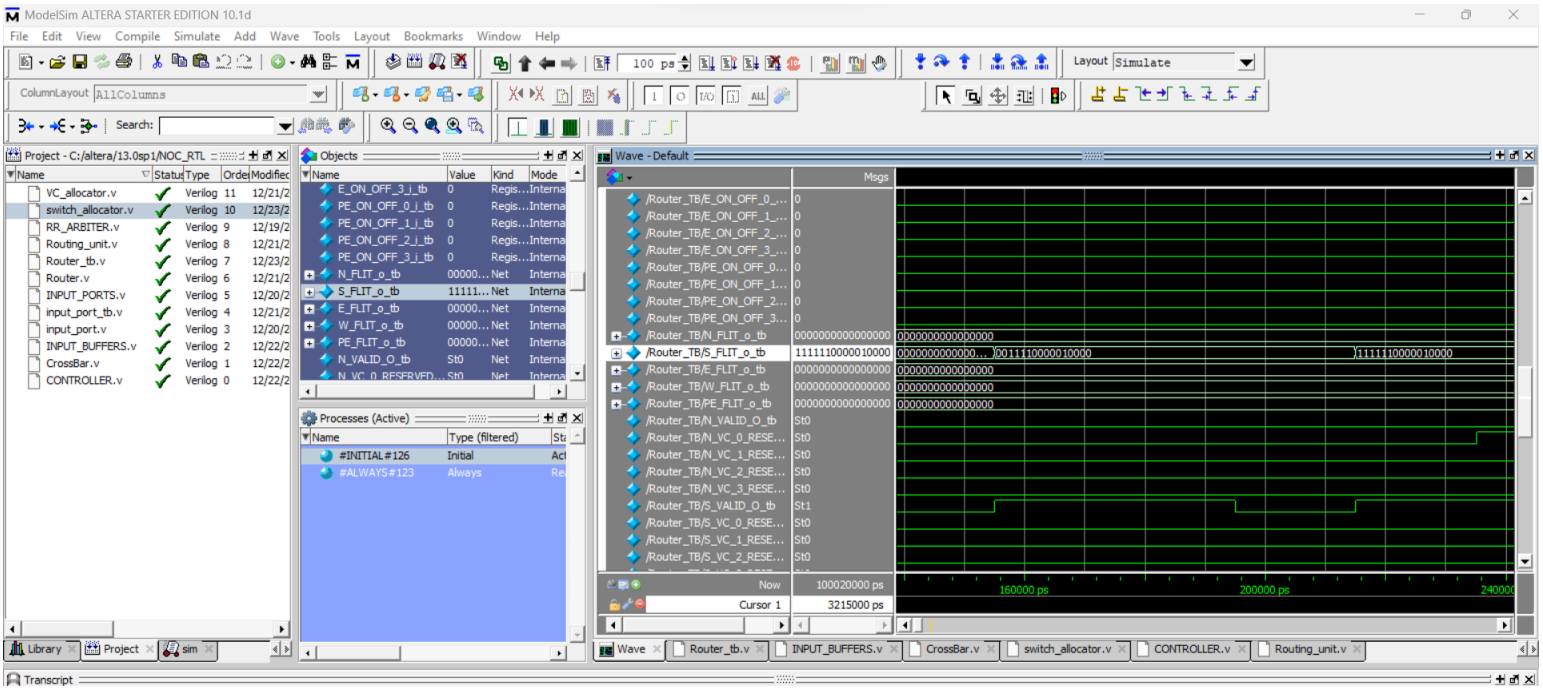
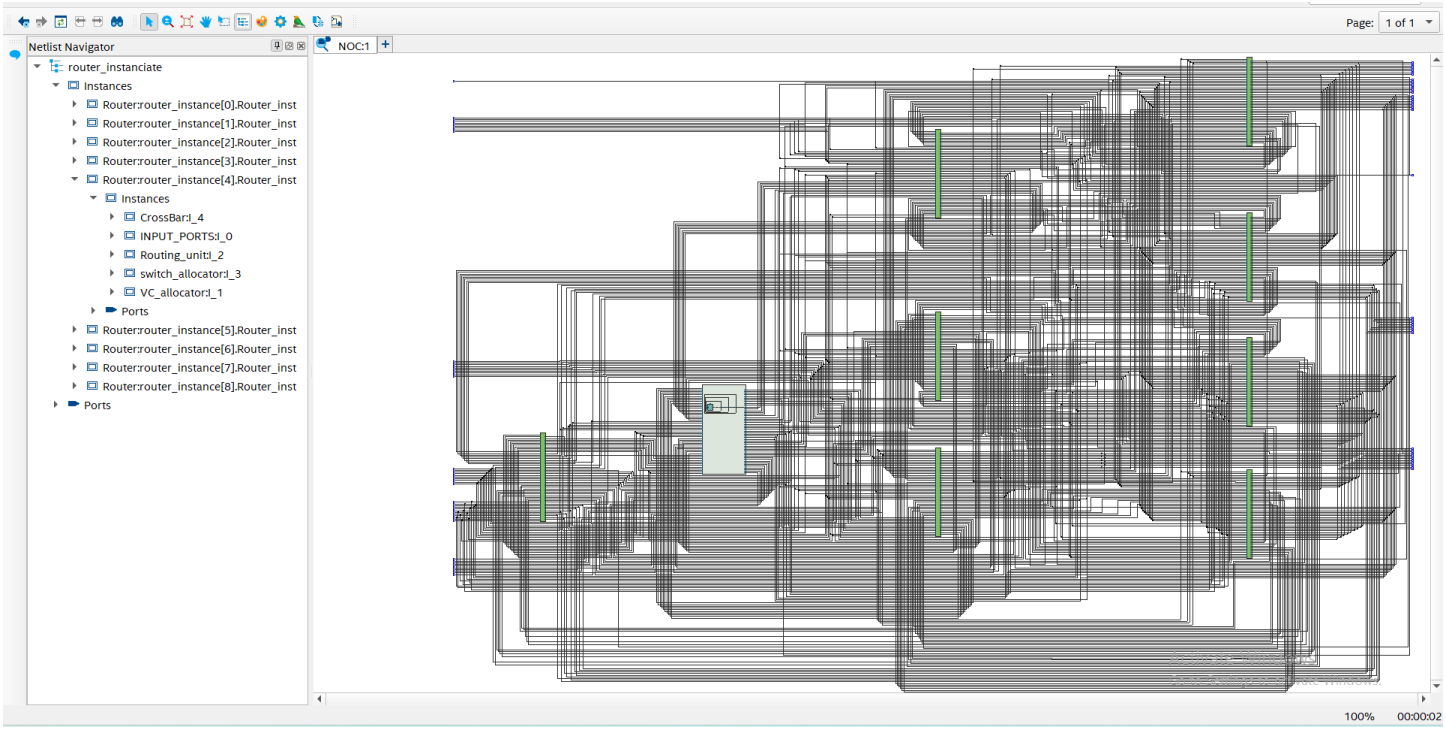


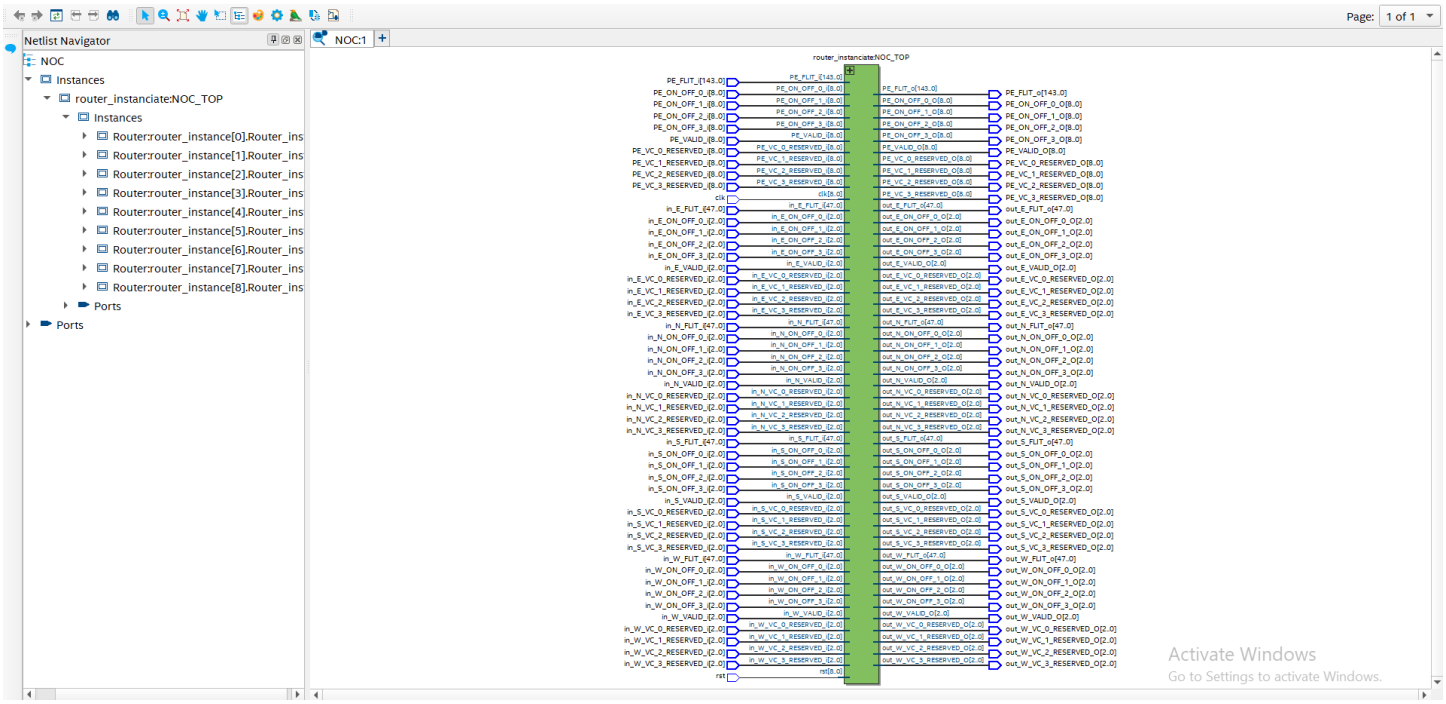
Fig. 2 different flits on 2 different input ports to same output ports

Netlist of connecting routers with each other

- We used for generate to generate 9 routers and instantiate them to the main router, then connected them to each other to generate the network.



NOC Netlist


Netlist Navigator
NOC
Instances
router_instanciate:NOC_TOP
Instances
Router:router_instance[0].Router_ins
Router:router_instance[1].Router_ins
Router:router_instance[2].Router_ins
Router:router_instance[3].Router_ins
Router:router_instance[4].Router_ins
Router:router_instance[5].Router_ins
Router:router_instance[6].Router_ins
Router:router_instance[7].Router_ins
Router:router_instance[8].Router_ins
Ports
Ports
NOC:1
router_instanciate:NOC_TOP
Page: 1 of 1
Activate Windows
Go to Settings to activate Windows.