Indonesian Journal of Electrical Engineering and Computer Science
Vol. 2, No. 1, April 2016, pp. 115 ~ 131
DOI: 10.11591/ijeecs.v2.i1.pp115-131

New Design of Network on Chip Based on Virtual Routers

Mohamed Fehmi Chatmen*[1], Adel Baganne[2] and Rached Tourki[3]
[1] Microelectronics Laboratory, Faculty of Sciences Monastir, 5000, Tunisia
[2]Laboratory of Science and Technology Information, communication and knowledge, UBS University, BP 92116, 56321 Lorient, France
[3]Microelectronics Laboratory, Faculty of Sciences Monastir, 5000, Tunisia
*Corresponding author, email: mohamed.fehmi.chatmen@gmail.com

Abstract

Network is considered the most convenient way to communicate between different IP integrated into the same chip. Studies have been developed to propose networks with improved performance in terms of latency, power consumption, throughput and quality of service. Most of these networks have been designed based on the 2-dimensional network structure. Recently, with the introduction of the new structure of 3D integrated circuits (3D IC), new works have used this type of circuit to design 3 dimensions on-chip networks. The advantage brought by this new structure is to reduce the average number of hops crossed from the source to the destination, which improves the throughput and the average latency of the network.

Keywords: *chip, latency, on-chip networks, power consumption, throughput, virtual routers*

1. Introduction

3D network differs from 2D network by the ability to integrate more IP (intellectual property) with less latency [1], particularly for routers that are distant from each other.

The main advantage of the 3D structure compared to 2D structure is that all routers are close to each other, making each router better connected: in fact, a router may have 6 connected neighboring routers in a 3D network, while in a 2D network; it may have at most 4 connected neighbors. This gives the 3D network more alternative paths for the transmission of the packet to its destination, which helps us avoid network saturation for low packets loads.

Figure 1 shows the extra possible paths that can be followed by packet to reach its destination in the 3D network compared to the 2D network; in fact, we have at least 2 extra alternative paths (the number of paths depends on the routing algorithm used on the network).

These advantages needed the invocation of much more resources; including ports number used for each router (one router has a maximum of 7 ports instead of 5) and the multiplexing units associated to each extra output port. In addition, the structure of the 3D network itself shows other issues especially regarding the employment of TSV (Through Silicon Via).

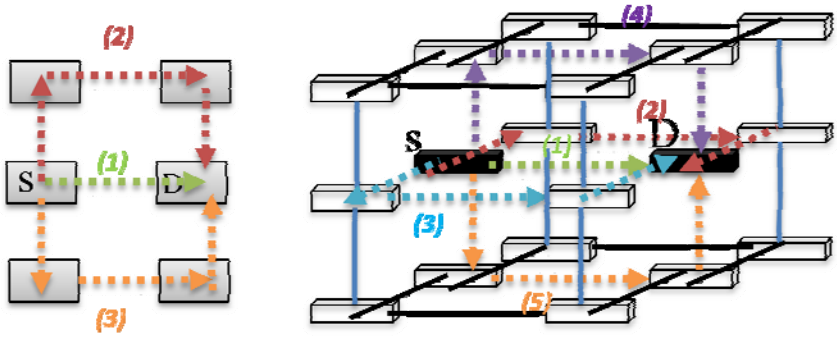

Figure 1. Advantage of 3D network comparing to a 2D network

Received January 19, 2016; Revised March 22, 2016; Accepted March 30, 2016

In this work, we present a new network architecture based on the concept of virtual routers, which can play the same role of a 3D network, but with fewer resources and better performance in terms of latency, in fact, there are two versions of this new architecture:

- The first version is named RVNOC (Reduced Resources Virtual Network on Chip) and is designed to use less resources and have respectful average latency, for that, every RVNOC router is sharing the NI link with few others.
- The second version is named LVNOC (Reduced Latency Virtual Network on Chip) and is designed to have a minimum of latency with an optimized amount of resources, for that, every LVNOC router has its own link to the NI.

Figure 2 shows the proposed architecture (c) with the conventional architectures, (a) for the 2D network and (b) for the 3D network.

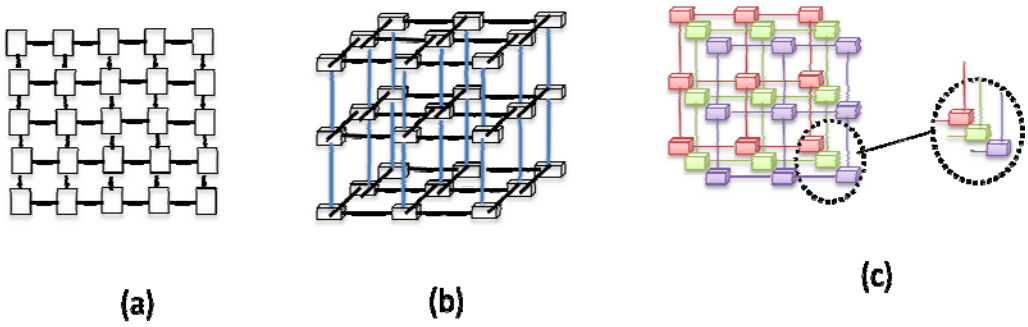

Figure 2. Networks structures

Both the LVNOC and RVNOC networks consist on a number of separated sub networks, which can only be connected to network interfaces; in fact the routers having the same coordinates in their own sub networks have the same network interface. The set of routers having the same network interface form the RG router (Global Router).

Other works on the 3D network have tried to reduce the number of vertical movements (along the z axis: TSV) because the TSV links are complicated, expensive [19, 9] and show considerable power consumption [10].

Our architectures remove these links (we did not need TSV); in fact, our architectures are 2D structures but can support a number of IP equivalent to a 3D network for any dimensions. These architectures were developed and simulated in VHDL using the MODELSIM6.5 tool.

2. Related Works

The reason for migration from 2D network to a 3D network is to increase throughput; for the same number of nodes, we have a lower average number of hops in a 3D network compared to a 2D network [3, 26]. But the 3D network itself shows other challenges especially regarding surface and energy consumption [17, 18] caused by the use of TSV and the difficulties of fabrication that make their integration limited. For this, reducing the number of vertical links (TSV) has been the goal of many works: the work of [7] shows a network structure in which the vertical links are placed irregularly. Bartzas in [8] also has tried to reduce the number of TSV by proposing specific paths for data flow. S. Pasricha in [2] proposes the serialization method as a solution to reduce the number of TSV.

Yan Ghidini in [5] has proposed reducing the number of TSV based on the principle of multiplexing and also adopting the serialization method [4] which is inspired form the work of Pasricha. This principle has been used in a network called LASIO which is the Hermes network expansion.

Liu Cheng proposed in [6] to share TSV between adjacent routers to reduce their total number, but to achieve this, an additional port was added to the structure of the routers in each layer forming the 3D network. Efsthios Sotiriou in [13] has proposed a heterogeneous architecture of network on chip formed by routers with 2D and 3D structure to minimize the use

of TSV. Efsthios criticized the excessive use of these links in works offering similar architectures [14, 15, 16] because they neglected the parameters related to the physical implementation.

In [11] the author has reduced the number of TSV showing that with only a few specific locations of TSV, it is possible to achieve a good tradeoff. In [12] the author has tried to reduce the number of TSV using a technique of routing, but this packet transmission technique generates considerable latency for a significant traffic. All these works have tried to improve 3D network performance in terms of surface consumption by acting on the TSV number but assuming that this minimization degrades network performance regarding the average latency. In this paper we show a new architecture of NOC based on the principle of virtual routers. This architecture is defined by a 2D dimension but with the performance of a 3D network and with much less resources.

Our paper is defined on three principal parts, in the first one we explain the structure and the functioning of our low latency router, in the second part we explain how we use the router to establish our new networks structures, and in the last part, we compare the performances of our networks to the others.

3. Architecture of the Router

3.1 Composition and Structure

Our router whose architecture is inspired from [20] is designed to have minimal latency for sending the packet from the source to the destination. In fact, the packet is routed through multiplexing units (CROSSBAR) which operate according to the combinational logic, and only the routing unit operates at the clock edge. This unit reads the header of the packet and directly allocates the destination port in one clock edge.

Figure 3. Structure of the router

Even if we think that our buffered router is the fastest router, it doesn't support quality of service and has no virtual channels. Note that the half counter unit is present only for RVNOC router and not for the LVNOC.

3.2 Principal Functions of the Routing Unit

Our router runs according to the handshake communication protocol. The switching mechanism adopted for our network is the Virtual Cut Through To avoid deadlocks.

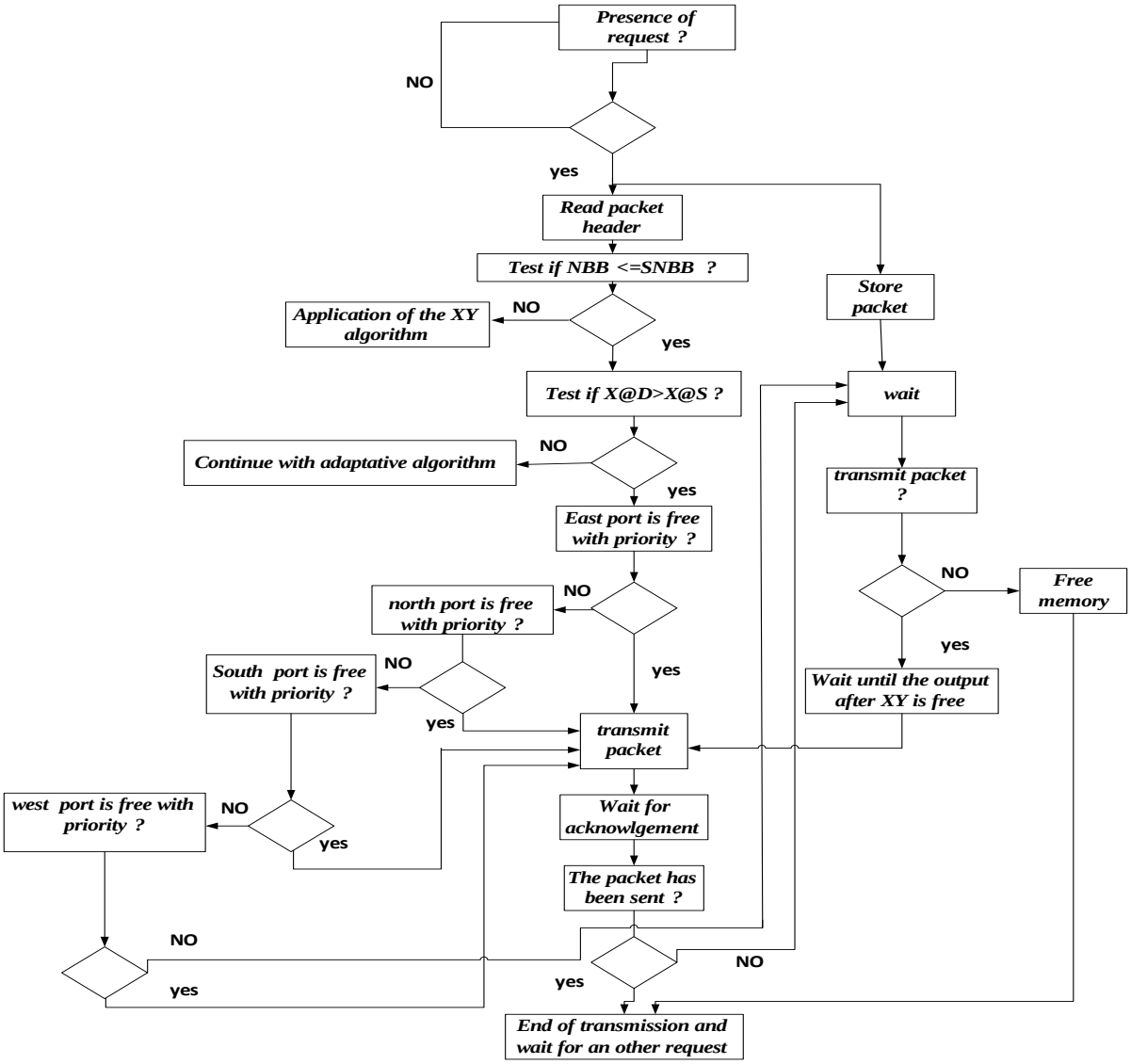

Figure 4. Operating principle of an elementary router

In the presence of request, the routing unit reads the packet header and it starts storing the packet in the storage memory unit (buffer). Meanwhile, the routing unit seeks whether the direction designated by the corresponding header is free (according to the XY algorithm) and that there are no other requests (with higher priority) designating the same output port, then it routes the packet through the multiplexing units (specifying the appropriate values for the input selection of the multiplexing unit) and reports that the port is unavailable to other possible requests, if not (the direction specified by the XY algorithm is unavailable), it seeks the availability of other ports, starting with the ports associated with the shortest path, to route the packet through.

In case all ports are unavailable, the packet already stored in the memory waits for the availability of the port designated by the XY algorithm favorited by a higher priority to the recents requests (designating the same output port). (See Figure 4)

If the sending is completed, the routing unit releases the data stored in the memory and turns the state of the associated output port to available. Note that all ports are initially at the available state.

For the RVNOC router, the routing unit sends packets to the local output port only at a corresponding half-cycles (the RVNOC router operates only for the half cycle specified by the half cycle counter unit);for the other half-cycles, the output port is set to 0 (the reasons will be explained later in section 4).

The figure 4 shows the principle of the routing algorithm using an example of a packet transmission from the local port to the eastern output port, note that the NNB signal is used to switch between the adaptive algorithm and the XY routing algorithm, which will be more detailed later in this paper

3.3 Packet Header and Instantaneous Routing

The packet header contains the information necessary to route the packet from the source to the destination. For our network the packet header is defined on 32 bits.
Packet header:

Y@S	X@S	@IP S	@IP D	P. length	Y@D	X@D

Where we denoted by:

- X@D: the coordinate of destination router following the x-axis defined on 4 bits.
- Y@D : the coordinate of destination router along the y-axis defined on 4 bits.
- P.length: the packet size that is defined on 16 bits.
- @IP D: the IP address relative to NI which is corresponding to the destination router and is defined on 4 bits.
- @IP S: the source IP address relative to NI and is defined on 4 bits.
- X@S: the coordinate of the source router following the x-axis and is defined on 4 bits.
- Y@S: the coordinate of the source router along the y-axis and is defined on 4 bits

As explained in section 3.2, the routing unit selects the output port instantly when reading the header. Indeed, a state signal (STA.P) is assigned to each output port that is set to 0 if the port is free and set to 1 if not. So we have 5 states signals each of which is associated with an output port. The states signals are STA.PW, STA.PE, STA.PN, STA.PS, and STA.PL, and are respectively associated to output ports following the directions WEST, EAST, NORTH, SOUTH, and LOCAL.The state signal value is set to 0 when the port is free (when no packet is being transferred to the associated output port and no already stored packets are waiting for the port availability). Otherwise, the state signal value associated to the output port is set to 1.

Table 1. State signals under different transmissions scenarios

STA.PW	STA.PS	STA.PN	STA.PE	X\{w}->w OR BX\{Bw}->w	X\{S}->S OR BX\{BS}->S	X\{N}->N OR BX\{BN}->N	X\{E}->E OR BX\{BE}->E
0	0	0	0+P	0	0	0	0
0	0	0+P	1	0	0	0	1
0	0	1	0+P	0	0	1	0
0	0+P	1	1	0	0	1	1
0	1	0	0+P	0	1	0	0
0	1	0+P	1	0	1	0	1
0	1	1	0+P	0	1	1	0
0+P	1	1	1	0	1	1	1
1	0	0	0+P	1	0	0	0
1	0+P (y ?)	0+P (y ?)	1	1	0	0	1
1	0	1	0+P	1	0	1	0
1	0+P	1	1	1	0	1	1
1	1	0	0+P	1	1	0	0
1	1	0+P	1	1	1	0	1
1	1	1	0+P	1	1	1	0
1	1	1	1	1	1	1	1

• 0: Port is in the available state
• 1: the port is unavailable
• 0 + P: the port is available and we favorite the transmission in his direction
• 0 + P (y?): Favors the transmission in the direction of the port if this minimizes the distance along the y-axis
• X \ {w}: set of input ports excluding WEST Input Port
• X \ {S}: set of input ports excluding SOUTH Input Port
• X \ {E}: set of input ports excluding the EAST Input port
• X \ {N}: set of input ports excluding the NORTH Input Port
• -> Ds : one of the entry ports is transferring data to the direction of the output port Ds
• BX: all buffers
• BX \ {BDe} -> Ds : one from the set of memories excluding the memory associated with the entry port BDe is transferring data to the output port Ds or waiting for the availability of the output port.

The Table 1 shows the values associated with the different state signals by considering a packet transmission from the local input port and its packet header indicating a X@D coordinate greater than X@R (X@D and X@R are respectively the destination router and the current router coordinates following x-axis) in this case the routing unit facilitates the transmission of the packet to the EAST direction if its state signal is set to 0. If not, we send the packet according to the NORTH or to the SOUTH depending on their availability, but if both are available the routing unit sends the packet to the direction that minimizes the distance between the destination IP and the current router according to the y-axis. In case both signals are set to 1, the packet will be sent to the WEST direction. If all state signals are set to 1 then the already stored packet waits the availability of the EAST output port, this time with a higher priority compared to recent queries.

3.4 Priority Of Input Ports and the Elimination of Live Lock

Our router uses the data stored in memory if there is a transmission error or all output ports are busy, otherwise, we can use any available output port to route the packets which can generate live lock problems (the packet never reaches its destination). And to avoid this problem, a signal called NB is used to indicate to the addressed router the number of times the packet has not followed the path specified by XY algorithm in the already crossed routers. According to the value carried by the NB and compared to the threshold TNB. The routing unit decides which routing algorithm to use: either continues with the adaptive routing, or just uses XY. The threshold depends on the network size (for example for the 3x3 network; the TNB is set to 3 and for the 4x4 network the TNB is set to 4). The choice of the TNB was made after making some performances measurements that we won't consider in this paper, in fact this algorithm which is a novel one, doesn't show better performances compared to the "look ahead" based ones. We only adopted this algorithm because of its implementation simplicity and the performances improvement compared to the deterministic XY. In addition, the establishment of this algorithm is not considered as an aim in this paper.

Our network operates according to FIFO scheduling algorithm, the first input port addressing the output port will be served first. In some cases, multiple input ports address the same output port at the same time (called here instant requests), so the FIFO algorithm is not applicable, and the request having the highest NB number is the one who has the highest priority. In case where both requests has the some NB then an arbitrary priority assignment is considered. This priority is defined as follows: PE > PS > PW > PN > PL with Px is the priority associated with the input port x. Note that the local input port has the lowest priority and this for the simple reason that other packets coming from other directions are older.

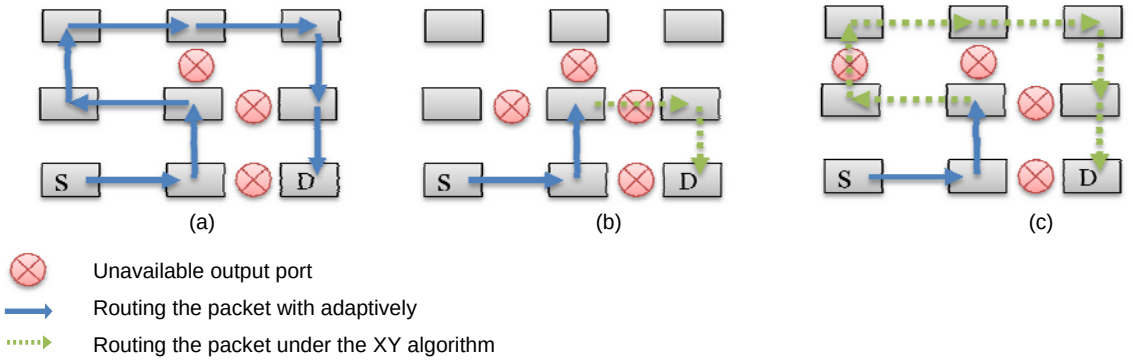

Figure 5. Network behavior to avoid deadlock and live lock

(a) Behavior of the routing algorithm if the packet is carrying a number of two miss-routing compared to XY (NB = 2)
(b) Behavior of the routing algorithm if the packet is in a situation where all output ports are unavailable, in this case the packet already stored in memory waits for the next port availability after the XY direction
(c) Behavior of the routing algorithm if the packet receives a number of three miss-routing(NB = 3) in this case the NB is equal to the TNB then the XY algorithm will be applied even if there are better alternative paths.

3.5 Packet Switching Principle

Our router is actually operating in a virtual cut through packet switching type with one modification which is the storage strategy; in fact the router stores immediately the packet in the buffer regardless the state of the addressed port. This does not mean that we must save the entire packet before sending it, but it will be sent as soon as the destination port is available. The instant storage was necessary for two reasons:

The first one is to have two packets addressing the same output port at the same time (the routing unit sees that the output port is free and gives access to both input packets) this will result the loss of data of one of two packets. In such case, the routing unit detects this error immediately (the routing unit always tests if it has given access to an output port to multiple input packets) and stops sending the one with lower priority without risk of losing data because they are already stored in memory. Those two actions operate as combinatorial functions.

The second reason is to have a sending error (receiving an error acknowledgment from the destination router) in this case the routing unit starts to resend the packet as soon as the output port is available.

The communication protocol adopted for our network is the handshake: In presence of a request, the routing unit sends an acknowledgment (ack = 00) to indicate the receipt of the header and then sends (ack = 01) to indicate the beginning of the receipt of the rest of the packet. It sends an acknowledgment (ack=11) to indicate the successful receipt of the entire packet. An acknowledgment (ack=10) is sent in case of transmission error. The treatment of these two communication obstacles were not specified by other on-chip network designers (it is imperative to store the packet for each communication to ensure that no packet will be lost).

4. Network Architecture With Virtual Routers

4.1. The New Networks Structure

Our networks (RVNOK or LVNOC) are composed by global routers RG, where each RG router comprises N elementary routers:

- A base router (R0) is equivalent to the router having the coordinate z=0 for a 3D network.
- N virtual routers: the first one (R1) is equivalent to the router R(X, Y, 1) having the same coordinates (X, Y) as the base router but with the coordinate z=1 and the second is equivalent to the router having the same coordinates (X, Y) as the base router but with the coordinate z=2. The n^{th} router is equivalent to R(X, Y, N).

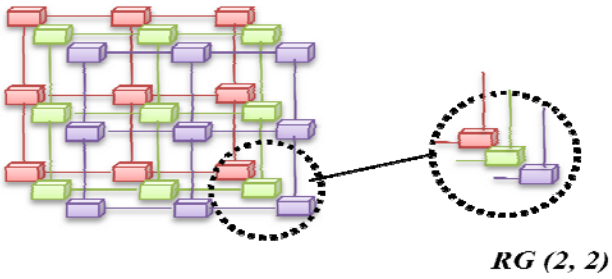

Figure 6. Network architecture with 2 virtual routers

In other words we replace the routers having the same XY coordinates but not the z by a single router known as RG. This allows to eliminate the Z+ and Z- directions as well as the multiplexing entities associated to each router for each level (layer). This leads to a large gain in terms of employed resources. For example, considering RG with only two virtual routers (N=2):

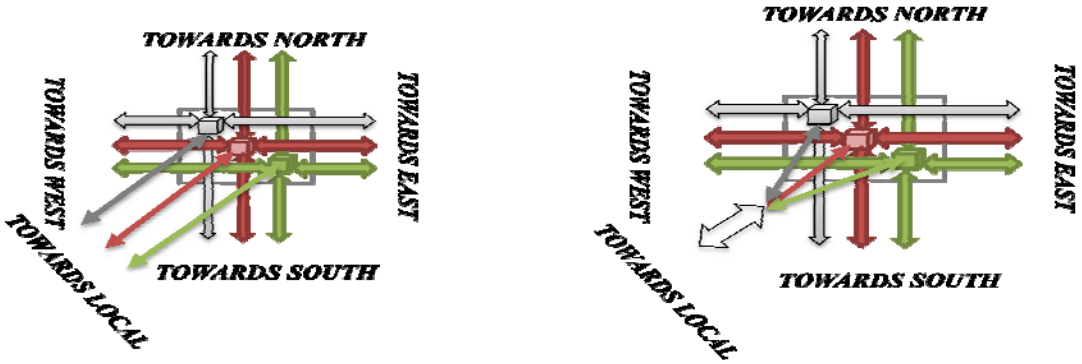

Figure 7. Architecture of a global router with 2 virtual routers

For the first version (RVNOC), the RG router has 12 input/output ports following the directions East, West, North, South equivalent to 24 one-way ports and a single input/output port (local port) equivalent to 2 one-way ports, instead of 3 routers with 14 input/output ports following the directions East, West, North, South, Z+, Z- (Equivalent to 28 ports) and 3 input/output ports to the local node (equivalent to 6 unidirectional ports). Totally, we have 26 ports for global routers instead of 34 ports, this is a reduction of 23%.

For the second version (LVNOC) the RG has the same ports number in all directions but with 2 extra local ports, so we have 30 ports instead of 34 (presented by the 3D router), this is a reduction of 12%.

4.2. Comparison of Our New Network Structure with the 2d Structure

Our architecture can be defined as a set of 2D networks separated from each other, the links between these different networks is done at the network interface (NI). And at each NI, we have a number of connected nodes equal to the number of separate networks. This structure has several advantages compared to 2D network:

4.2.1. Number Of Links

The proposed structure is optimized compared to a 2D network [27]; in fact, the number of links for the conventional 2D network is determined by the equation $Eq.1$:

$$NL = N * M + (N - 1) * M + N * (M - 1) \quad \text{Eq.1}$$

With N: number of network columns
M: number of network lines
For our RVNOC structure, the number of links is determined by the equation $Eq.2$:

$$NLr = Ns * Ms + \big((Ns-1) * Ms + Ns * (Ms-1)\big) * NNE \quad \text{Eq.2}$$

For our LVNOC structure, the number of links is determined by the equation $Eq.3$:

$$NLl = \big(Ns * Ms + (Ns-1) * Ms + Ns * (Ms-1)\big) * NNE \quad \text{Eq.3}$$

With:
Ns: number of columns in a single network.
Ms: number of lines in a single network.
NNE: number of separate networks.
To get an idea of the reduction rate, let us consider a 2D network with size 5x5 (N = M = 5) with possibility to connect 25 cores. After applying $Eq.1$ we have a total of 65 bidirectional links.
The equivalent structure associated with our architecture is defined by a set of three 2 dimensional networks of 3x3 (NEE = N1 = M1 = 3) with possibility to connect 27 cores, after applying $Eq.2$ we get a total of 45 bidirectional links for RVNOC network.
For the LVNOC network, after applying $Eq.3$, we get a 63 bidirectional links. So we can connect much more cores to our proposed architectures with less links.

4.2.2. Algorithmic Complexity

Algorithmic complexity necessarily depends on the network size (especially regarding the issue of live lock) and since the size of our network is much less than the 2D ones because it is divided by NNE, we can say that our architecture offers a significant reduction on the computational complexity

4.2.3. Latency

It is clear that with increasing the size of the network, the average latency increases, our new architecture always shows a reduced size compared to a 2D network, even with a larger number of connected cores, so certainly our architecture presents a reduced latency (considering the low injection rate for the RVNOC structure) compared to its equivalent in the 2D networks.

We're talking about a smaller network size because we consider for each communication a separate network (which always has a reduced size compared to a 2D network).

Figure 8 shows a comparison between the latency of a 2D network of size 5x5 with both RVNOC 3x3 and LVNOC 3x3:

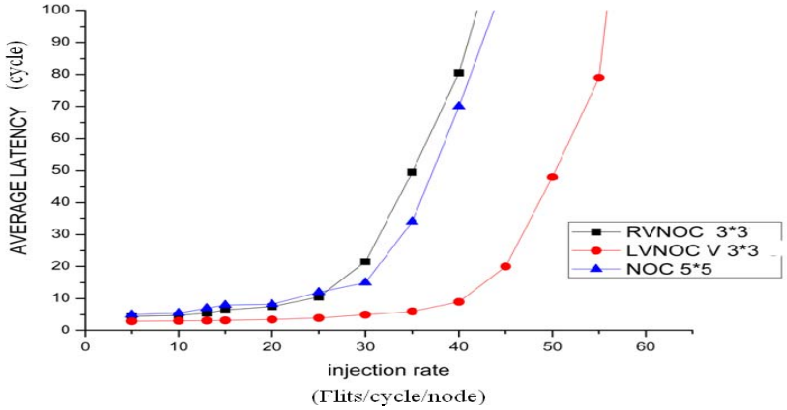

Figure 8. The average latency of a 2D network of size 5x5 compared with both RVNOC 3x3 and LVNOC 3x3

Figure 8 shows that the performance of the RVNOC is better than the 2D NOC only for low injection rate. The LVNOC always shows reduced latency compared to other networks. (Reasons will be explained later in this paper).

4.3. The Time Multiplexing Principle

In addition to its role of delivering different packages to different destinations, the routing unit also allows to synchronize the different levels of the network (here we talk about the RVNOC). In fact, the routers belonging to the same separate network operate at the same period. Which is not the same for other routers having a different index; Base routers are also functioning on their own period (half-cycle): the half-cycle counting unit counts the half-cycle modulo N-1, the number of cycles introduced to the routing unit (between 1 and N-1) indicates which type of routers is operating. As an example, let's consider N as the number of half-cycle and a network with two virtual routers:

- n = 1: the base routers (equivalent to z = 0 in 3D NOC) are in operating state
- n = 2: virtual routers VR1 (equivalent to routers having z = 1 in 3D NOC) are in operating state
- n = 0: virtual routers VR2 (equivalent to routers having z = 2 in 3D NOC) are in operating state

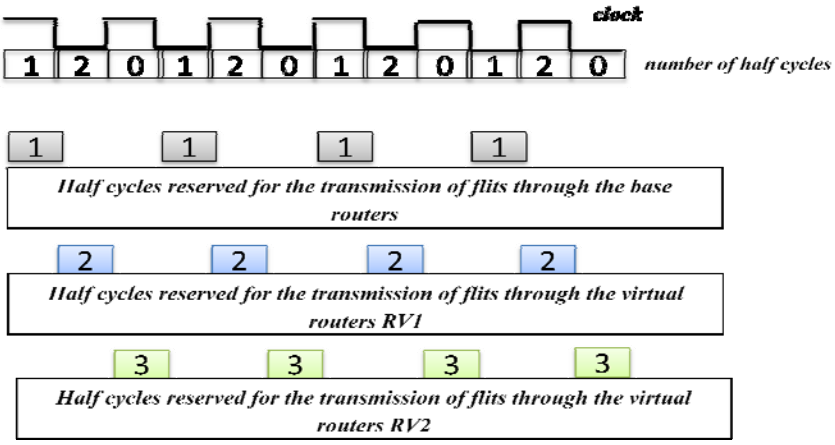

Figure 9. Time division multiplexing for three elementary routers

The RG router allows transmitting two flits at the end of one cycle (1 flit for the first half cycle of the first cycle and second flit in the second half cycle of the second cycle) for each router, averaging a latency of 1 cycle for the transmission of a single flit between two neighbors RG routers.

This function (The time multiplexing) concerns only the RVNOC version while the LVNOC all routers work simultaneously at the same clock edge.

4.4. Structure of the Local Port for Rg Router

4.4.1. Case of the Rvnoc

Beside the deletion of displacement along the z-axis (z +, z-), we reduced the number of local ports while maintaining the same number of cores that can be connected in a 3D network. Actually, a set of N cores can be connected to a single global RG router using one input port and one output port. The output port named DATA_OUT_LOCAL_RG is multiplexed (shared) between N output ports from N elementary routers. This multiplexing is done using the multiplexing unit that has a single function which is the OR gate between the N output signals.

In fact, each signal carries the appropriate data within the appropriate half cycle, otherwise, it is set to 0 (see section 3.2). Therefore, at each half cycle, a single signal has appropriate data (and is not set to zero) so that the result of the OR gate operations with other

signals has no influence. Thus, at each half-cycle, we have a data transfer associated with a single virtual level (separate network).

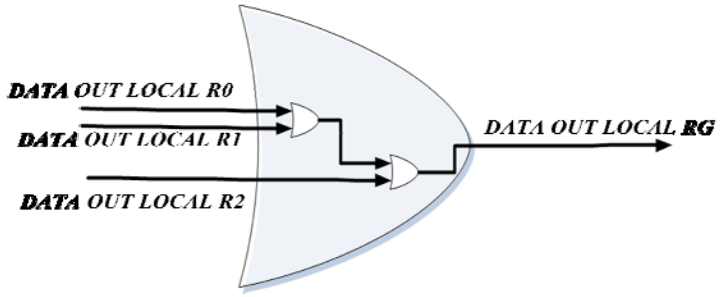

Figure 10. Output local port multiplexer (case 3RVNOC)

The input port DATA_IN_LOCAL_RG is multiplexed between the N ports incoming to NI (network interface) from the N cores (IP). The NI uses a multiplexing unit similar to the one of the RG router. This time, setting signals to 0 at the corresponding half-cycles is done by the network interface. (See Figure 11)

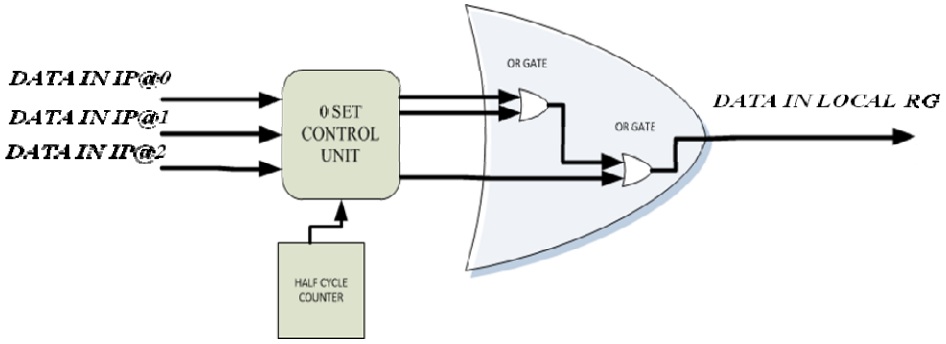

Figure 11. Input local port multiplexer (case 3RVNOC)

Note: 3RVNOC means RVNOC network with 3x3 dimensions
The input port DATA_IN_LOCAL_RG is connected directly to all elementary routers forming the RG router and the demultiplexing is done implicitly within each elementary router (reads the data of the port only at the corresponding half-cycle). There is no need then to use additional demultiplexing unit.

4.4.2. Case of the Lvnoc

The RG router in this case has N input/output local ports where each input/output port is connected to one of N elementary routers.

4.5. Structure of the Network Interface

For both LVNOC and RVNOC structure, the network interface allows different IP to communicate with the network. Actually, if an IP with address @S transmits a packet to a destination with address @D, the packet must be routed through all the routers belonging to the same separate network (which corresponds to the virtual level of the destination IP). If two IPs sharing the same NI, address two locations with the same virtual level, then the packet from one of the them will be forwarded through another available virtual level specified by the NI (same thing in case when more than one IP are addressing the same @D since we have as many virtual levels as IPs connected to the same NI). So, to route a packet to its destination, we can use any virtual level that guarantees the package delivery to the corresponding RG.

The NI, according to the address specified by the IP header, will deliver the packet to its destination @D. If packets from different virtual levels are addressing the same IP destination, NI transmits one package and stores the others in buffers and retransmits them when the path is available.

For the RVNOC NI, The memory buffer stores the packet at the level associated to the half cycle defined by the corresponding virtual level and retransmit it at the half cycle defined by the destination IP level. For that, we attribute N-1 buffer for each IP destination. For N = 3 we have two buffers for each IP, see Figure 12.

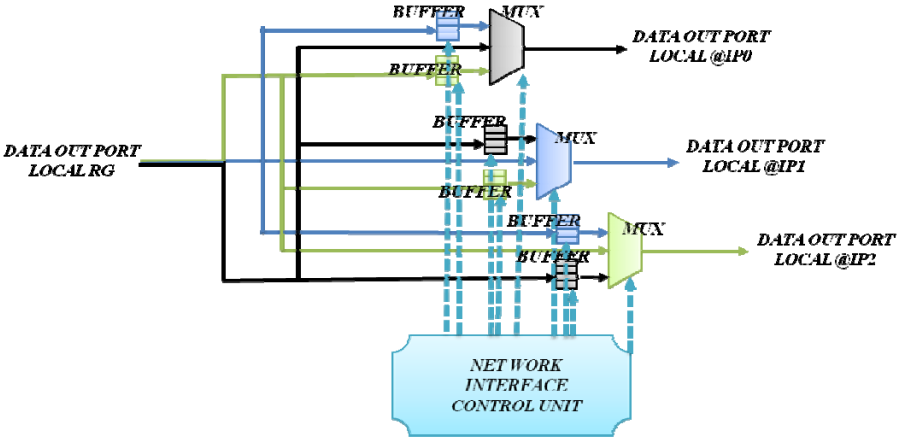

Figure 12. structure of the RVNOC network interface

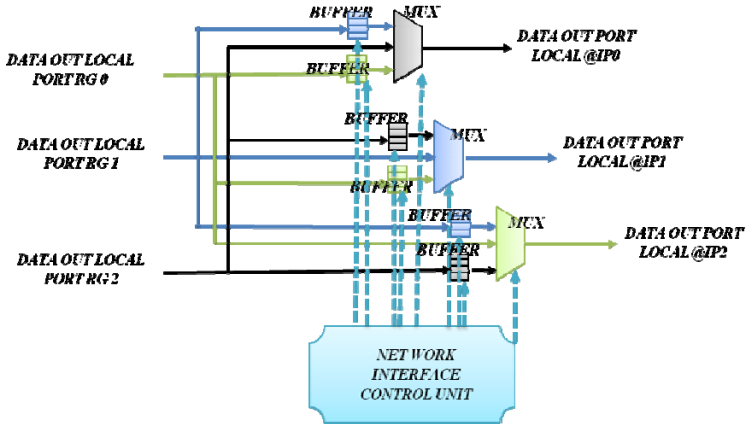

Figure 13. Structure of the LVNOC network interface

Fitting the period of the addressed IP with the period of the virtual level inside the buffer is done by counting half cycles: The flit is written in the memory at the half cycle reserved for the associated virtual level designed by the source IP, and is output at the half cycle associated with the virtual level of the destination IP.

For the LVNOC NI, there is no need to count half cycles, in fact all NI components work simultaneously at the same clock edge.

To transmit data in the network, the LVNOC NI assigns the input to one of the different outputs depending on their availability, and as it is already mentioned, NI does not take into consideration the virtual level used for transmission.

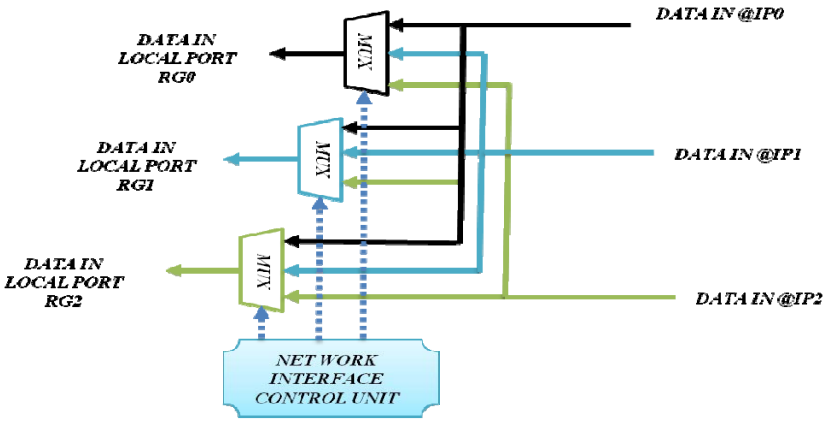

Figure 14. Structure of the LVNOC network interface.

In fact, every NI output port is multiplexed from all NI input ports, this multiplexing depends on the virtual level defined by the @D and the availability of an output port (if we can't transmit data through the output port related to the virtual level defined by @D, the NI affects the input data to another available output port).

5. Our Latency Reduction Compared to Conventional 3d Network

To show the importance of the latency reduction of our network compared to the 3D conventional network, we will use the following example: consider the 3D network of size 3x3x3 and the router with address (0, 2, 0) sends a packet to the router designated by the address (1, 1, 2), and the latency to transmit a flit from an input port to an output port is 1 cycle, then we have four routers to cross from the source to the destination which induces a latency of 4 cycles.

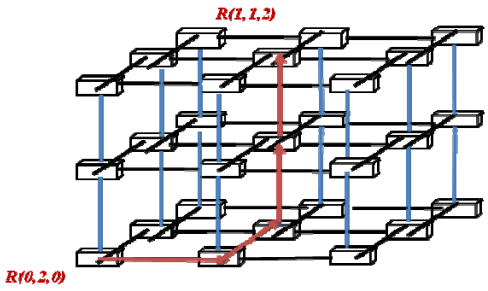

Figure 15. Example of a communication within a classic 3D network

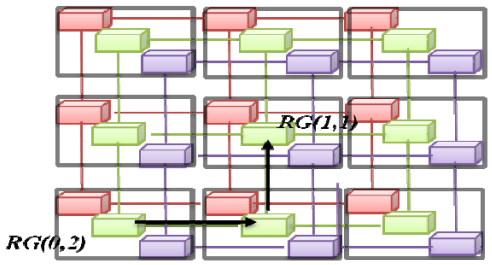

Figure 16. Example of a communication within a network with virtual routers

This communication is similar to transmitting the packet from RG router with coordinates (0,2) to the RG router with coordinates (1,1) within a network with virtual routers with a dimension of 3x3. Then we have a path with two hops which means a latency of 3 cycles for a RVNOC router and only 2 cycles for the LVNOC router.

The transmission speed of the router (one clock edge is sufficient to transmit a flit from the input port to the output port) and the latency removal within the axis Z, have allowed us to have more efficient networks comparing to most conventional 3D networks. In fact, the figures below show the networks average latency according to the different injection rates under a uniform traffic for both LVNOC and RVNOC. The calculation of the average latency is based on the tests approach inspired from [21] and using the ModelSim tool:

Figure 17. The average latency under uniform traffic for networks RVNOC 3 x 3, RVNOC 4x4 and RVNOC 5x5

Figure 18. the average latency under uniform traffic for networks LVNOC 3 x 3, LVNOC 4x4 and LVNOC 5x5.

The simulation shows that for the RVNOC latency of our network is close to that of a 2D network for low injection rate and this is explained by the temporal multiplexing principle. We note that the RVNOC network get saturated at poor injection rates because the different virtual levels (separate networks) operate in pipeline which generates extra delay. In fact, this delay is proportional to the size of packets and the dimension of the designed network. This explains the significant increase of latency as the network gets a small increase of size as shown in Figure 17.

The LVNOC network shows a higher performance compared to the RVNOC, this is explained by the fact that each network interface has an additional number of ports equivalent to the number of the separated networks (NNE) which improves the data throughput, and by the fact that for this network version we don't have an overhead time for data transmission caused by the pipelined behavior as in the RVNOC network. The gap of performance between these two networks gets bigger for higher injection rates.

Figure 19 shows a comparison between the performances of our networks and the network presented by [7] in terms of latency. This comparison is made to a 3D network of size 5x5x5, which corresponds to a network with size 5x5 for our both networks structures:

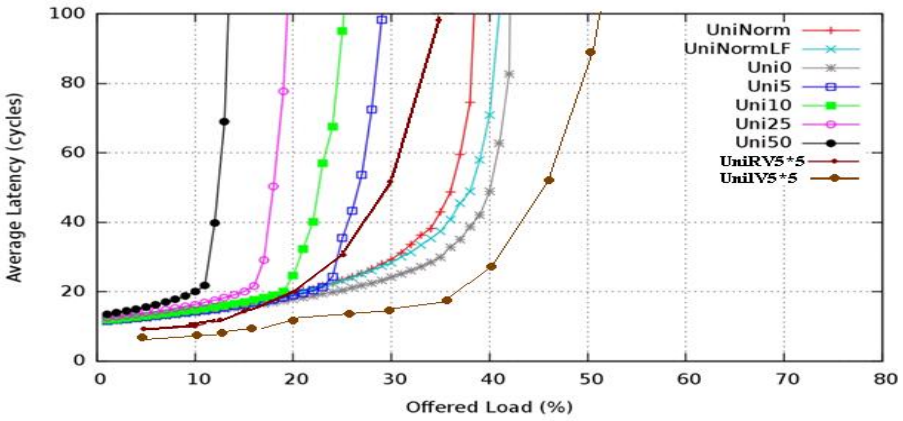

Figure 19. Latency comparison between networks with virtual routers and a classic 3D network

For Figure 19 we have:

- "UniNorm": presents the average latency of a fully connected 3D-Mesh network when using the ordinary XYZ routing algorithm.
- "Uni0": presents the average latency using the Elevator-First routing algorithm for a fully connected 3D-Mesh network.
- "UniNormLF": latency of "UniNorm" using lager FIFO.
- The curves "Uni5,""Uni10,""Uni25," and "Uni50" are the results when 5, 10, 25, and 50 percent of vertical links are removed.
- UNIRV5*5: presents the average latency of the RVNOC network with size 5x5 under a uniform traffic.
- UNILV5*5: presents the average latency of the LVNOC network with size 5x5 under a uniform traffic.

Figure 19 shows that before the saturation, the RVNOC is more efficient to all other structures proposed by [7] even in the case of the fully connected network. But in the saturation phase, our network becomes less efficient to the completely connected structures but still more efficient to the structures in which the number of TSV is reduced. The LVNOC shows greater performances compared to all other structures before and after saturation.

6. Implementation of the Designed Routers Using FPGA Board

In order to highlight the area of our network, we have implemented the various elementary routers on an FPGA board VIRTEX 6 XC6VLX760 using the tool XILINX ISE 14.1 (we considered that LVNOC and RVNOC have the approximately the same router).

Table 2. Area and frequency of our different routers

Type of elementary router	3 PORTS ROUTER	4 PORTS ROUTER	5 PORTS ROUTER
frequency (MHZ)	362	41	23
area	2986	6570	25821
FPGA DEVICE	VIRTEX 6	VIRTEX 6	VIRTEX 6

The implementation of different routers shows that the number of ports has major effects on routers performance in terms of surface consumption and maximum execution frequency.

The table presented below shows our results corresponding to an elementary router with 5 ports compared to other works whose networks architectures are implemented on FPGA board.

Table 3. Area and frequency of our router compared to others implementations.

Design	[22]	[23]	[25]	[24]	[25]	VNOC router
Number of ports	7	7	7	7	7	5
Frequency (MHz)	327	353	195	250	65	23
Area	1391	1273	7847	11550	3557	25821

Table 3 shows that our router has 5 ports instead of 7 since our elementary router has no ports along the Z+ and Z- directions. We also note that our router consumes more surface compared to others, this is explained by the use of 16 FLIT sized buffer at each input port and by the employment of multiplexing units to achieve the pipelined operating. Most of the surface is allocated for the logic (the number of logic LUTS is equal to 25681 which is about 99% of the consumed surface). This is obvious because our routing unit doesn't work in a round robin method which is not the case for the work presented by [22, 23, 24 and 25], so that we can say that our router is greedy in terms of surface consumption but works smarter.

7. Conclusion

In this paper we have introduced new networks architectures that can be considered as alternatives to the 3D network but with less resources (in terms of number of interconnect links and all associated Crossbar). For the first proposed network structure (the RVNOC), the performance is significant for small-sized networks and for transmission of medium sized packets but will be degraded in a major way when extending the network (see Figure 19). The LVNOC structure shows a better performance in terms of latency compared to the RVNOC structure and other studied 3D architectures; we believe actually that our LVNOC network is the best network solution which can replace the 3D architecture and resolve the issues of the TSV employment.

References

[1] MH Jabbar and D Houzet, O Hammami. *3D Multiprocessor with 3D NOC Architecture Based on Tezzaron Technology*. in: IEEE International 3D System Integration Conference (3DIC), Osaka. 2012.
[2] S Pasricha. *Exploring Serial Vertical Interconnects for 3D ICs*. in: Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE .
[3] A Vino Vilmet Rose, R Seshasayanan. Optimization of vertical links in a three-dimensional NOC based multicore crypto processor for cloud computing. in: *Journal of Engineering, Computing and Architecture*.
[4] Y Ghidini, M Moreira, L Brahm, T Webber, N Calazans, C Marcon. "*Lasio 3D NoC vertical links serialization: Evaluation of latency and buffer occupancy*". in: 26th Symposium on Integrated Circuits and Systems Design (SBCCI). 2013: 1 – 6.
[5] Y Ghidini, M Moreira, L Brahm, T Webber, N Calazans, C Marcon. "TSV Multiplexing: A 3D NoC Occupancy Analysis".
[6] C Liu, L Zhang, Y Han, and X Li. *Vertical interconnects squeezing in symmetric 3D mesh Network-on-Chip*. in: 16th Asia and South Pacific Design Automation Conference (ASP-DAC). 2011: 357 – 362.
[7] F Dubois, A Sheibanyrad, F Petrot and M Bahmani. *Elevator-First: A Deadlock-Free Distributed Routing Algorithm for Vertically Partially Connected 3D-NoCs*. in: IEEE Transactions on Computers. 2013; 62(3): 609-615.
[8] A Bartzas, K Siozios, D Soudris. *Three-dimensional Networks-on-Chip architectures*. in: NETWORKS-ON-CHIPS. 2009: 1.
[9] D Velenis, M Stucchi, E Marinissen, B Swinnen and E Beyne. *Impact of 3D design choices on manufacturing cost*. in: IEEE International Conference on 3D System Integration. 2009.
[10] T Xu, P Liljeberg and H Tenhunen. '*A study of Through Silicon Via impact to 3D Network-on-Chip design'*. in: International Conference on Electronics and Information Engineering. 2010.
[11] T Xu, P Liljeberg and H Tenhunen. *Optimal Number and Placement of Through Silicon Vias in 3D Network-on-Chip*. in: IEEE 14th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS). 2011: 105 –110.
[12] C Rusu, L Anghel and D Avresky. 'Message routing in 3D networks-on-chip'. *NORCHIP*. 2009.
[13] E Sotiriou-Xanthopoulos, D Diamantopoulos, K Siozios, G Economakos and D Soudris. *A framework for rapid evaluation of heterogeneous 3-D NoC architectures*. in: Microprocessors and Microsystems. 2014; 38(4): 292-303.

[14] V Pavlidis and E Friedman. *Three-dimensional integrated circuit design*. Amsterdam: Morgan Kaufmann. 2009.
[15] I Anagnostopoulos, A Bartzas and D Soudris. *Application-Specific Temperature Reduction Systematic Methodology for 2D and 3D Networks-on-Chip*. in: Lecture Notes in Computer Science. 2010: 86-95.
[16] A Richard, D Milojevic, F Robert, A Bartzas, A Papanikolaou, K Siozios, and D Soudris. *Fast Design Space Exploration Environment Applied on NoC's for 3D-Stacked MPSoC's*. in: Proc. Architecture of Computing Systems. 2010: 16.
[17] WR Davis, et al. *Demystifying 3D ICs: The Pros and Cons of Going Vertical*. In Design & Test of Computers, IEEE, 2005; 22(6): 496-510.
[18] K Bernstein, P Andry, J Cann, P Emma, D Greenberg, W Haensch, M Ignatowski, S Koester, J Magerlein, R Puri, and A Young. *Interconnects in the third dimension: Design challenges for 3-D ICs*. in Proc. 44th ACM/IEEE Des. Autom. Conf. 2007: 562–567.
[19] JH Lau. *TSV Manufacturing Yield and Hidden Costs for 3D IC Integration*, in: Proceedings of Electronic Components and Technology Conference 2010, Las Vegas, USA. 2010: 1031-1042.
[20] R Ben-Tekaya, A Baganne, K Torki, R Tourki. *Performance evaluation of MIC@R NoC for real-time applications*. in : International Journal of Computer Aided Engineering and Technology. 2010; 2(3): 274-293.
[21] GM Chiu. *The Odd-Even Turn Model for Adaptive Routing*. in: IEEE Trans. Parallel Distrib. Syst. 2000; 11(7): 729-738.
[22] MO Agyeman and A Ahmadinia. *An adaptive router architecture for heterogeneous 3d networks-on-chip*. In: NORCHIP. 2011: 1–4.
[23] X Yu, L Li, Y Zhang, H Pan and S He. *Mass message transmission aware buffer-less packet-circuit switching router for 3D Noc*. in: 10th IEEE International Conference on Control and Automation (ICCA). 2013: 983- 986.
[24] K Tatas et.al. *A Novel NoC Architecture Framework for 3D Chip MPSoC Implementations*. in: Design, Automation and Testing in Europe (DATE) Friday Workshop on 3D Integration. 2010.
[25] B Chemli and A Zitouni. *A Turn Model Based Router Design for 3D Network on Chip.* in: World Applied Sciences Journal. 2014; 32(8): 1499.
[26] Tan Hai, Shahnawaz Talpur, Amir Mahmood Soomro, Chen Hong Mao. A Long-wire-connected and Multi-channel 3D Network-on-chip Design for Many-core System. *TELKOMNIKA.* 2013; 11(12): 7081~7087.
[27] Sonal S Bhople, MA Gaikwad. Design of Mesh and Torus Topologies for Network-On-Chip Application. *International Journal of Reconfigurable and Embedded Systems (IJRES)*. 2013; 2(2): 76~82.