Vertical Links Minimized 3D NoC Topology and Router-Arbiter Design

Nallasamy Viswanathan[1], Kuppusamy Paramasivam[2], and Kanagasabapathi Somasundaram[3]
[1]Department of Electrical and Computer Engineering, Mahendra Engineering College, India
[2]Department of Electrical and Computer Engineering, Karpagam College of Engineering, India
[3]Department of Mathematics, Amrita Vishwa Vidyapeetham, India

Abstract: *Design of a topology and its router plays a vital role in a 3D Network-on-Chip (3D NoC) architecture. In this paper, we develop a partially vertically connected topology, so called 3D Recursive Network Topology (3D RNT) and using an analytical model, we study the performance of the 3D RNT. Delay per Buffer Size (DBS) and Chip Area per Buffer Size (CABS) are the parameters considered for the performance evaluation. Our experimental results show that the vertical links are cut down upto 75% in 3D RNT compared to that of 3D Fully connected Mesh Topology (3D FMT) at the cost of increasing DBS by 8%, besides 10% lesser CABS is observed in the 3D RNT. Further, a Programmable Prefix router-Arbiter (PPA) is designed for 3D NoC and its performance is analyzed. The results of the experimental analysis indicate that PPA has lesser delay and area (gate count) compared to Round Robin Arbiter (RRA) with prefix network.*

Keywords: *Network topology; vertical links; network calculus; arbiter; latency; chip area.*

Received June 26, 2014; accepted July 7, 2015

1. Introduction

The advancements in the fabrication technology permit the designers to use 3D Integrated Circuits (ICs) for the design and implementation of logic circuits [11]. It is believed that 3D Network-on-Chip (3D NoC) i.e., NoC can be implemented in 3D ICs, is an obvious alternative to overcome the drawbacks of 2D NoC [7]. In such scenario, the researchers stare at three dimensional NoC 3D NoC which has added advantages like higher packing density, shorter global wire length, reduced area overhead requirement and integration of both homogeneous and heterogeneous modules [14].

The key issues in 3D NoC research are:

1. Evolving an optimum 3D NoC topology.
2. On-chip router design.

In this paper, an experimental study is conducted on a 3D NoC architecture using a simulator 'VNOC3', a versatile software framework for 3D NoC, to exhibit the performance benefits of 3D NoC interconnect architecture [1, 8]. A partially vertically connected 3D NoC topology, 3D RNT, is developed and its performance is evaluated using a network calculus based analytical model [4, 6]. Simulation experiment is conducted using Network Simulator-2 (NS-2) to validate the effectiveness of the model [2]. The performance of 3D RNT is compared with that of 3D Fully connected Mesh Topology (3D FMT) which is a natural choice of topology for 3D NoC. Further, a

Programmable Prefix Arbiter (PPA) for 3D NoC is designed and implemented in FPGA. Its performance is Analyzed and compared with Round Robin Arbiter with prefix networks (RRA).

2. Performance Benefits of a 3D NoC

A cycle accurate NoC simulator 'VNOC3' is used to study the performance benefits of a 3D NoC architecture. The simulator is integrated with an efficient B-Tree- based floor planner for optimized experimental results. The 3D NoC architecture is developed using three layers; first and third layers are used for the placement of SoC modules while second layer is devoted for implementing network. A typical 2D NoC and three layers 3D NoC architectures are shown in Figure1.

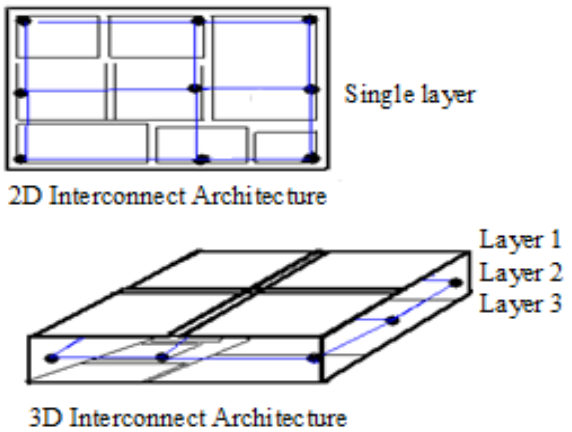

Figure 1. 2D NoC and three layers 3D NoC architectures.

In 2D NoC architecture, network and floor plan are implemented in a 2D plane whereas the floor plans are

placed in first and third layers and network is realized in the second layer of the 3D NoC architecture. The input to the simulation design flow is the application which is represented using Communication Tasks Graph (CTG) employed to map the modules using a distributed task mapping heuristic algorithm presented in [12].

The CTG is partitioned into two sub graphs for 3D NoC architecture. In the partitioning step, h-Metis partitioning algorithm is used to partition a single layer into two layers [12]. Based on a B-Tree representation, we integrate the floor planner such that it will be able to handle vertical constraints and it employs a simulated annealing based algorithm with a cost function that combines area and wire length

$$Cost\ function = \alpha \cdot Area + (1 - \alpha) \cdot Wire\ Length \quad (1)$$

In Equation 1, the coefficient of area α is selected such that $\alpha \in [0, 1]$ in order to provide a good balance between the area and wire length. This cost function is a convex combination of area and wire length. This ration is an optimum ration for an efficient packaging. A number of floor plans are generated for 2D and 3D NoC architectures by running floor planner N times with selected weights for area and wire length. During this step, a number of best M≤ N floor plans are placed in the M list; the selection is made according to the chosen criterion of smaller area or shorter total wire length.

The default values of N and M are N=30 and M=10. In the experiment, four test benches , namely apte, xerox, hp, ami 49 chosen from Microelectronics Center of North Carolina (MCNC) are used whose area is scaled to provide an average size of about 1 cm × 1 cm, which is a typical area required for an NoC implementation. The characteristics of the test cases are given in Table 1.

Table 1. Test cases used for the experimental study.

Test benches	Number of modules	Average Weight / Height (W/H) (sq.µm) of modules	2D topology
ape	8	434/2499	3 × 3
xerox	10	2114/2872	4 × 4
hp	11	4533/924	4 × 4
ami49	49	1089/1123	7 × 7

The parameters chosen for simulation are:

1. Packet size of 5 flits with each flit being 64 bits.
2. Input buffer size 12 flits and the number of virtual channels is 12.
3. XY routing and wormhole flow control.
4. $\alpha = 0.25$.
5. Simulation cycles count:6000.
6. Warm up cycles count:1000.

The CPU runtime is approximately 15 seconds in Linux platform at 2.5 GHz speed with 2 GB memory.The screen capture of the simulation output for the test case ‘hp’ after floor planning is shown in Figure 2. The chip area and wire length are obtained from the simulation results. The simulation results for chip area requirement, wire length and floor plan dead-space for each test case are studied.

The simulation results for the available area, used area and wire length for each test case are given in Table 2. In the test case of ‘hp’, the eleven modules are placed at a single layer in 2D interconnect architecture whereas 3 modules are placed in the first layer and the remaining 8 modules are placed in the third layer in the case of 3D NoC architecture as shown in Figure 2.

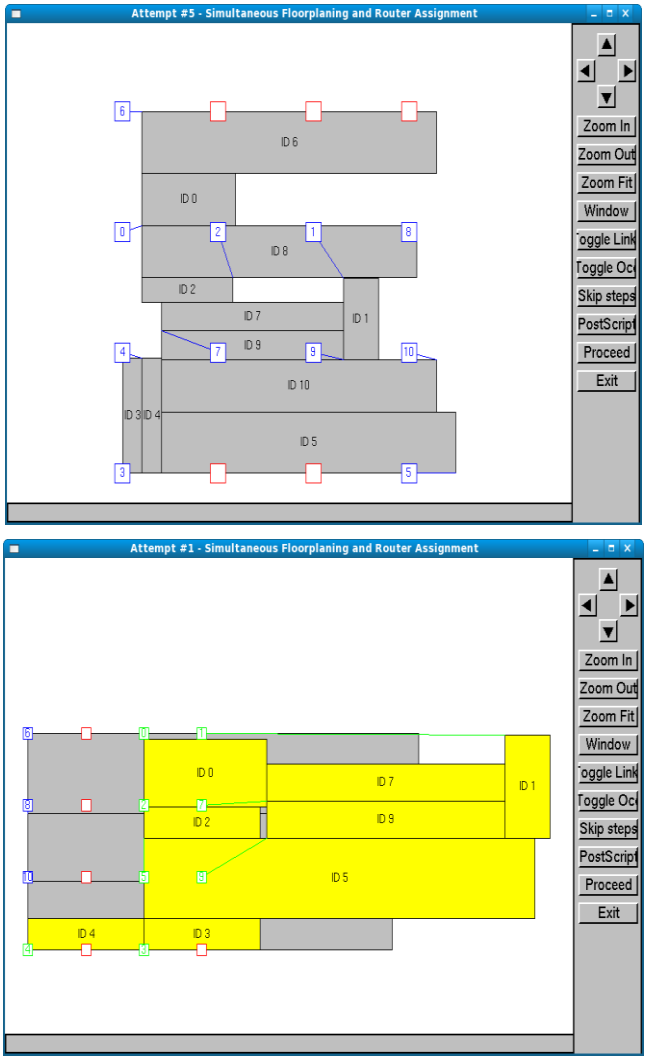

Figure 2. Floor planning and routers assignment for 2D NoC and 3D NOC architectures for the test case ‘hp’.

Table 2. Simulation results for four test cases.

Test cases	Modules partitions		Total Area (H×W) in sq.µm		Wire length in µm		Used area in sq.µm	
	2D NoC	3D NoC	2D NoC	3D NoC	2D NoC	3D NoC	2D NoC	3D NoC
apte	3 × 3	4 & 4	10.930 e+07	4.675e+07	466571	363461	8.6429e +07	4.3215e +07
xerox	4 × 4	6 & 4	4.9198 e+07	2.251e+07	769809	558575	3.6107e +07	1.8853e +07
Hp	4 × 4	3 & 8	3.3128 e+07	1.466e+07	369983	218188	1.6477e +07	0.7795e +07
ami49	7 × 7	11 & 38	11.8495 e+07	4.6550e+07	0.3722e +07	0.2434e +07	6.6132e +07	3.8288e +07

The floor plan dead-space (F_d) is computed as follows:

F_d= *[Total chip area (W*H) in sq. µm – Used chip area (W*H) in sq.µm]* (2)

where W-width of modules; H-height of the modules.

The percentage of reduction in chip area, floor plan dead-space and wire length in 3D NoC is compared with that of 2D NoC architecture as shown in Figure 3. It is observed from Figure 3 that more than 57% reduction in chip area and 84% reduction in floor plan dead-space are obtained in the test cases ' ami49' and 'apte' respectively in 3D NoC architecture compared to 2D NoC. This is because the former test case has 84% more number of modules, even though it has 89% lesser average module area than that of the test case 'apte'.

Similarly, more than 54 % reduction in chip area and 59 % reduction in floor plan dead-space are noticed in the test cases of 'xerox' and 'hp' since the test case 'xerox' has 31% more average module area and 9% lesser number of modules than that of the test case 'hp'.

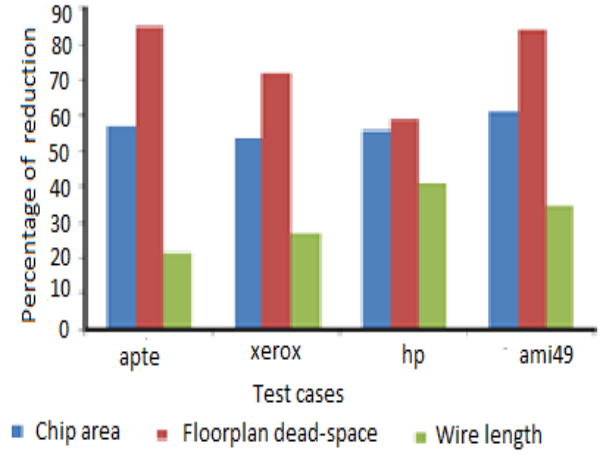

Figure 3. Chip area, floorplan dead-space and wire length of 3D NoC architecture compared to 2D NoC.

It is concluded from the simulation results that an average 57% chip area, 75% floor plan dead-space and an average 31% wire length are reduced in the three layer architecture 3D NoC than that of the single layer architecture 2D NoC. The variation observed in chip area, floor plan dead-space and wire length in each test case is due to the difference in the size and number of modules used in the test cases.

3. Performance Evaluation of 3D RNT

In this section, we develop a 3D Recursive Network Topology (3D RNT); we evaluate its performance using an analytical model and compare with 3D FMT.

3.1. Topology

A 3D NoC Topology, 3D RNT shown in Figure 4 has three layers in which each node comprises of a switch and module; each four nodes are grouped to create four clusters in each layer thus each layer has sixteen nodes [15]. In each cluster, a node is identified using an ID of three digits xyz where first digit x represents a layer, second digit y represents a cluster and digit z is the ID for either a Cluster Head (CH) or node in the cluster y and layer z. For instance, 100, 111, 122, and 133 are the IDs of cluster head nodes in layer 1and only 25% of nodes in each layer are interconnected with counterpart nodes of neighboring layers using vertical links so as to minimize the number of vertical links [14].

A 3D routing algorithm is developed for flits communication among the modules as follows:

- *Step 1.* Finding the destination layer.
- *Step 2.* Finding the destination cluster in the destination layer.
- *Step 3.* Finding the destination node in the destination cluster and layer [3].

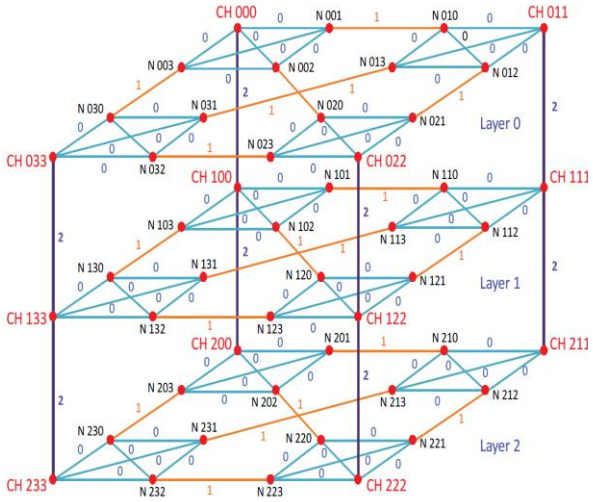

Figure 4. 3D RNT) with substituting and flipping link labels.

The performance of 3D RNT is evaluated and compared with 3D FMT with a mesh size of 4 x 4 x 3. In 3D FMT, one more dimension ' Z ' is added to 2D Mesh Topology (2D MT). In the 3D FMT, all the nodes of a layer are interconnected with the counterpart nodes of the neighboring layers using vertical links. The ID for the nodes of 3D FMT is assigned as similar as the ID assigned for the nodes of 3D RNT. In 3D FMT, ZXY routing algorithm is used for flits communication among the IP cores [10].

We develop an analytical model based on network calculus for evaluating the performance of 3D RNT and 3D FMT in respect of the performance and cost metrics of end-to-end end delay, switch buffer size and chip area. Network calculus provides deep insights into flow related problems encountered in networking [13, 16].

Here, we consider switch arrival curve α (t) as a linear equation $\alpha(t)=b+rT$, output curve $\alpha^*(t)=\alpha(t)+rT$, service curve $\beta(t)= R\ (t-T)$, switch maximum input buffer size $B=b+rT$ and switch delay $D=b/R+T$, where

R is switch service rate, b is burst size of the data flow, r is data injection rate and T is maximum latency caused by a switch [16]. These functions are shown in Figure 5.

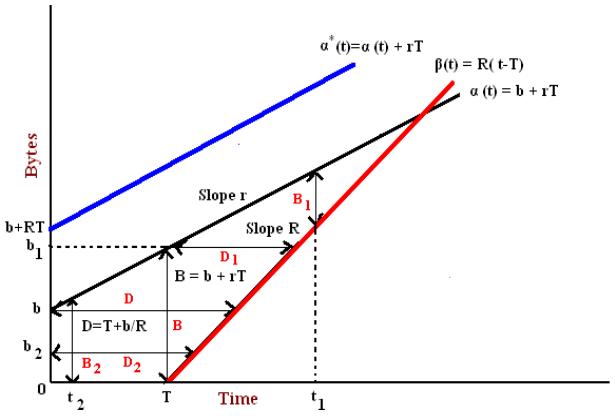

Figure 5. Arrival and service curves, switch input buffer size and delay of the model.

In 3D RNT, the arrival curve of a switch S is defined as $\alpha_s(t)=\alpha_i(t)+\sum\alpha_n(t)$, where $\alpha_i(t)$ is the arrival curve of the input data flow of its own module and $\sum\alpha_n(t)$, is the arrival curve of the input data flows arriving from n neighboring switches. The following five synthetic traffic patterns are considered here to evaluate the performance of the topologies [15, 16].

1. Nearest Neighbor traffic (NN).
2. Hot Spot traffic (HS).
3. Digit Reversal (DR).
4. Transpose (TP).
5. Interlayer maximum distance traffic (IL).

In each topology, randomly selected IP cores are considered as source nodes. For destination nodes, IP cores are selected according to the defined traffic patterns. In 3D RNT and 3D FMT, nine source-destination pairs are selected according to the synthetic traffic patterns to establish nine different data flows and the routing path of each data flow is determined using the 3D routing algorithms developed for the topologies. Table 3 shows the IDs of the source-destination pairs selected for 3D RNT under the five traffic patterns.

Table 3. Source-destination pairs for 3D RNT.

Traffic Patterns	IDs of source-destination pairs
NN	213→211, 131→113, 030→032, 213→233, 222→200, 211→100, 102→222, 222→000,020→233
HS	213→231, 220→200, 200→130, 003→130,010→130, 231→000, 002→122, 210→101,010→113
DR	201→102, 210→012, 100→001, 200→002,122→221, 120→021, 211→112, 122→221,220→022
TP	020→213, 033→200, 023→210, 102→131, 112→121, 100→133, 010→223, 011→222,203→030
IL	231→011, 000→222, 223→010, 113→222, 101→211, 233→111, 130→020, 002→123,122→011

Consider the traffic pattern 'IL' for the two topologies; nine source-destination node pairs are selected in each topology according to the traffic pattern. For 3D RNT, a data flow graph is drawn to indicate the routing paths for each data flow as shown in Figure 6

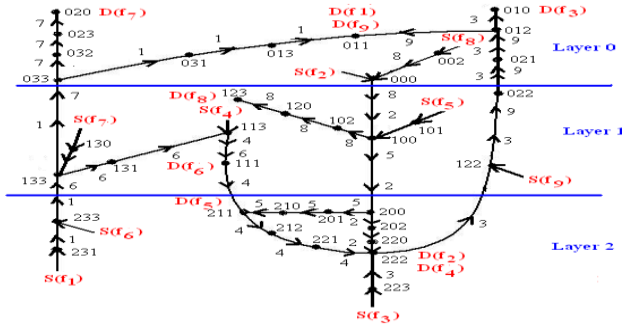

Figure 6. Data flow graph for 3D RNT under the traffic pattern 'IL'.

The data flow graph is used to compute the arrival curve $\alpha(t)$ of each switch; the arrival curve is computed iteratively using the equations $B=b+rT$ and $D=b/R+T$. In first iteration, we compute the arrival curve of the switches participating in the first data flow and subsequently we compute the arrival curve of the switches participating in the second data flow.

Likewise, the arrival curve of the switches participating in the remaining data flows are computed in the same manner [15]. The arrival curve of the switches participating in the data flows are given in Table 4. The maximum buffer size requirement B and data transfer delay D of each switch S participating in the data flows can be computed by using the equations $B=b+rT$ and $D=b/R=T$. The equations required to compute the switch buffer size B and the delay D for 3D RNT are derived from the arrival curve of the switches. Similarly, the arrival curves of the switches and equations required for computing switch buffer size and delay are computed for 3D FMT under the traffic pattern 'IL'.

Table 4. Arrival curves of the switches for 3D RNT under the traffic pattern 'IL'.

ID of the Switches	Arrival curves	ID of Switches	Arrival curves
231,002,101,130,223	$rt + b$	020	rt + b + 5.3333rT
000,233	2rt + 2b + rT	221	rt + b + 6.5rT
100,133	$3rt + 3b + 2rT$	222	3rt + 3b + 13.8333rT
102,131	$rt + b + 2.3333rT$	122	2rt + 2b + 5.6111rT
120,032,201,202,031	$rt + b + 3.3333rT$	022	2rt + 2b + 7.6111rT
123,023,013	$rt + b + 4.3333rT$	021	2rt + 2b + 9.6111rT
111	$2rt + 2b + 5.3333rT$	012	2rt + 2b + 11.6666rT
210,220	rt + b + 4.3333rT	010	rt + b + 6.8055rT
113	2rt + 2b + 3.3333rT	011	2rt + 2b + 12.1388rT
211	2rt + 2b + 9rT	033,200	2rt + 2b + 4.6666rT
-	-	212	rt + b + 5.5rT

In the experimental study, it is assumed that each source node injects flits at a deterministic rate r varied from light traffic (20 Gbps) to heavy traffic (100 Gbps)

in order to reflect the real traffic, R=400 Gbps, maximum burst size of the data flow b=100 bytes and flit size k=10 bytes.

Table 5. End- to- end switch delay bound for 3D RNT.

End- to- end switch delay in ps									
r	**f1**	**f2**	**f3**	**f4**	**f5**	**f6**	**f7**	**f8**	**f9**
20	23190	23721	25989	21960	19950	19679	18059	18004	19963
40	24150	23972	26460	22315	20155	19799	18236	18124	20368
60	24397	24223	26930	22668	2036	19920	18414	18246	20773
80	24645	24476	27400	23022	20566	20042	18591	18367	21178
100	24891	24727	27872	23376	2077	20163	18768	18488	21584

The end-to-end switch delay of a data flow is computed by summing up the delay of the individual switches participating in the data flow under the traffic pattern 'IL'and it is given in Table 5 and 6 respectively.

Table 6. End- to- end switch delay bound for 3D FMT.

End- to- end switch delay in ps									
R	**f1**	**f2**	**f3**	**f4**	**f5**	**f6**	**f7**	**f8**	**f9**
20	25972	30092	20029	18012	13952	22207	13927	15945	18062
40	26427	30568	20315	18329	14120	22435	14071	16058	18429
60	26879	31042	20599	18645	14287	22660	14213	16169	18794
80	27332	31518	20883	18962	14456	22887	14356	16280	19160
100	27785	31991	21167	19278	14623	23113	14498	16392	19527

The average end-to-end delay of 3D RNT and 3D FMT under various data injection rates are computed by summing up the delay of the individual switches and links participating in the data flows. The average link delay is computed by assuming the individual link delay is to be 0.1ps and the average end-to-end delay bound of 3D RNT and 3D FMT under various data injection rates is computed and given in Table 7.

Table 7. Average end-to-end delay bound of 3D RNT and 3D FMT under various data injection rates.

	3D RNT			**3D FMT**		
r in Gbps	**Average end-to-end switch delay in ps**	**Average link delay in ps**	**Average end-to-end delay in ps**	**Average end-to-end switch delay in ps**	**Average link delay in ps**	**Average end-to-end delay in ps**
20	21168.4	500	21668.4	19799.8	500	20299.8
40	21508.8	500	22008.8	20083.6	500	20583.6
60	21770.1	500	22270.1	20365.36	500	20865.36
80	22031.9	500	22531.9	20648.26	500	21148.26
100	22293.2	500	22793.2	20930.5	500	21430.5

Further, Table 8 shows the average buffer size requirement of the switches for the data injection rates vary from 20Gbps to 100Gbps. Similarly, the average end-to-end delay and the switch buffer size requirement are computed for 3D RNT and 3D FMT by constituting different data flows using the other four synthetic traffic patterns, NN, HS, TP, and DR.

Table 8. Average switch buffer requirement of 3D RNT and 3D FMT under the traffic pattern 'IL'.

r in Gbps	**Number of switches participating in the data flows**		**Average buffer size required in a switch B_s (Number of bytes)**	
	3D RNT	**3D FMT**	**3D RNT**	**3D FMT**
20	36	38	153	145
40			156	148
60			158	150
80			161	153
100			164	156

Further, we analyze the influence of the buffer size B_s in determining the area required for a chip. Switches, modules and links are the three major sources of a chip area in an NoC and an average area can be computed using the equation given below [13].

$$A = N_{(s)}\left(R_s + a_s d_g s_f B_s\right) + N_b A_b + a_l N_l L_l \qquad (3)$$

Table 9. Chip area for implementing 3D RNT and 3D FMT.

r in Gbps	**Chip area in sq.µm**	
	3D RNT	**3D FMT**
20	56449200	55171470
40	56898080	55605860
60	57346970	56040240
80	57795850	56474620
100	58244750	56909010

For 3D RNT, the parameters other than the average buffer size B_s are assumed in Equation (3) as follows: number of switches N_s=16, switch silicon area required for routing table and logic to implement a routing algorithm R_s=1000 sq.µm, area required for one byte a_s=1 sq.µm , an average number of buffers inside the switch d_g=1, size of the flits S_f = 10 bytes, area requirement for a module A_b =1000 sq.µm, area required for a link a_ℓ= 10 sq.µm, link length L_ℓ=1µm, number of bidirectional links N_ℓ =30. The Equation (3) is reduced as $A=32.0003+0.16B_s$. Similarly, $A=32.00024+0.16B_s$ is the equation for computing the area required for 3D FMT.

An average chip area requirement of 3D RNT and 3D FMT is computed using the average buffer size B_s of the switches. An average chip area requirement of the two topologies of 3D RNT and 3D FMT is computed at various data injection rate r and it is given in Table 9.

The above average chip area overhead analysis is carried out for 3D RNT, 3D FMT using the synthetic traffic pattern 'IL'. Similarly, the average area requirement for 3D RNT and 3D FMT is computed using the other synthetic traffic patterns.

3.2. Performance Comparison of 3D RNT with 3D FMT

The performance and cost metrics of the two topologies in respect of end-to-end delay, switch buffer size and chip area are computed using the analytical model and the performance of the proposed topology 3D RNT is compared with 3D FMT using two figures

of merit

Delay per Buffer Size (DBS) = End to End Delay (ps) / Switch Buffer Size (bytes) (4)

Chip Area per Buffer Size (CABS) = Chip Area sq. µm / Switch Buffer Size (bytes) (5)

The end-to-end delay and chip area of each topology under various traffic patterns are related with an average switch buffer size by using the figures of merit; DBS and CABS are computed and given in Tables 10 and 11.

Table 10. DBS for the two topologies under various traffic patterns and data rate.

Topologies and traffic patterns	DBS (ps / bytes)				
	r = 20 Gbps	r = 40 Gbps	r = 60 Gbps	r = 80 Gbps	r = 100 Gbps
3D RNT-NN	87.47939	87.04446	86.6198	86.20808	85.8074
3D FMT-NN	101.2433	100.2155	99.22183	98.26822	97.3616
3D RNT-HS	172.5089	152.1876	136.4859	124.2251	114.1532
3D FMT-HS	121.4358	115.2829	109.6816	104.9178	100.4659
3D RNT-DR	83.61285	83.19432	82.78752	82.39265	82.0026
3D FMT-DR	79.58067	79.17536	78.77856	78.38148	77.9974
3D RNT-TP	145.5237	144.7492	143.9949	143.2692	142.5611
3D FMT-TP	121.8225	121.0822	120.3401	119.5426	118.7757
3D RNT-IL	141.8037	141.4346	140.5792	139.7568	138.9593
3D FMT-IL	140.1724	139.5166	138.8712	138.2559	137.6576

DBS of the two topologies under various traffic patterns is compared and it is given in Figure 7. It is observed from Figure 7 that 3D RNT has better interconnect architecture in the horizontal planes due to the following factors:

- Each cluster is interconnected with its diagonally opposite cluster .
- Each switch within a cluster has better connectivity with its diagonally opposite switch.

Table 11. CABS of the two topologies under various traffic patterns and data rate r.

Topologies and traffic patterns	CABS (sq.µm / bytes)				
	r= 20 Gbps	r= 40 Gbps	r= 60 Gbps	r= 80 Gbps	r= 100 Gbps
3D RNT-NN	373192.5	370010.2	366921.6	363922.6	361009.1
3D FMT-NN	388503.1	383867.3	379415.8	375137.9	371023.6
3D RNT-HS	435345	398703.3	370446.7	348349	330233.4
3D FMT-HS	406841.3	390817.2	376366.8	363955.6	352504.1
3D RNT-DR	347947.3	345936.8	343968.9	342042.2	340155.4
3D FMT-DR	405818	402820.3	399894.6	397038.8	394250.2
3D RNT-TP	370019.4	366805	363687.4	360662.5	357726.2
3D FMT-TP	475079.3	470305.4	465673.9	461178.7	456813.8
3D RNT-IL	369418.4	365642.8	362000.8	358485.7	355090.7
3D FMT-IL	380965.3	376899.2	372980	369199.9	365551.7

In 3D RNT under the traffic pattern ‘NN’, five of the source-destination node pairs are located within a same layer and a set of five source-destination node pairs are located in diagonally opposite clusters.

All the source-destination node pairs are located in the same clusters even though the source and destination nodes are at different layers in respect of the traffic pattern ‘DR’. Hence, 3D RNT performs better under the traffic patterns NN and DR than that of other traffic patterns in respect of DBS.

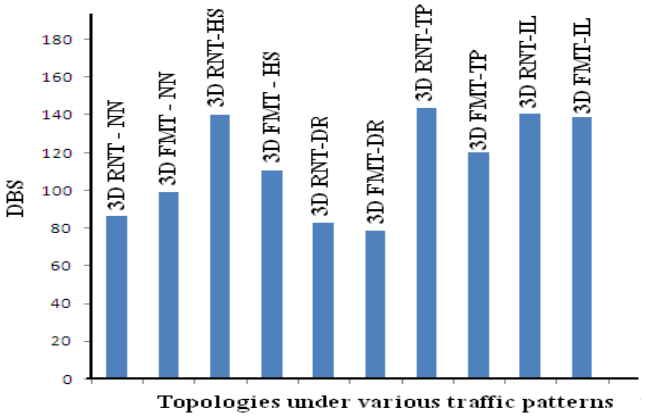

Figure 7. Comparison of DBS of the two topologies under various traffic patterns.

Maximum DBS is observed in 3D RNT under the traffic pattern ‘TP’ due to the following factors:

- 3D RNT has poor connectivity between two neighboring clusters of a 2D plane.
- 3D RNT has limited vertical links.
- Six source-destination node pairs are located in different layers which lead to more interlayer traffic.
- All source nodes have destination nodes in neighboring clusters; source-destination node pairs are located either in a same layer or in different layers.

As traffic in a switch increases, network becomes more congested with heavy traffic; flits have to wait in buffer of a switch if it handles heavy traffic. As a result, DBS is increased under the traffic pattern ‘HS’ in 3D RNT since 33% of the total traffic is targeted to the designation node ID ‘130’. 3D RNT has more DBS under the traffic pattern ‘IL’ also since more than 68% interlayer traffic happen, and use of limited number of vertical links which leads the need of more hops to reach destination nodes.

In 3D FMT, the nine source-destination pairs are located in the same clusters even though the source and destination nodes are at different layers which reduce the number of hops required in a 2D plane. Hence the traffic pattern ‘DR’ shows better performance in 3D FMT.

The traffic pattern ‘IL’ requires more DBS in 3D FMT as more hops are needed for data transfer in each horizontal plane in addition to the hops required for vertical links due to the clusters which are not connected diagonally in each 2D plane. Figure 8 shows CABS of the two topologies under various traffic patterns. On comparing 3D FMT with 3D RNT, the former topology has more CABS under the traffic pattern ‘TP’ due to the following factors:

- 100% nodes are interconnected with the counterpart nodes of the adjacent layers in 3D FMT and hence interlayer traffic is distributed to many vertical links.

- All source nodes have destination nodes in neighboring clusters; source-destination node pairs are located either in a same layer or in different layers.
- 3D FMT is better connected between neighboring clusters in each 2D plane.
- Average buffer size of the switches participating in the data flows is reduced.

In 3D RNT, the traffic pattern 'DR' shows minimum CABS due to the following factors:

- 3D has limited vertical links.
- All nine data flows have interlayer traffic.
- More traffic is in vertical links.
- Average buffer size of the switches participating in the data flows is increased.

It is inferred for 3D RNT from the figures of merit that

- Each cluster is interconnected with its diagonally opposite cluster.
- Each switch within a cluster has better connectivity with its diagonally opposite switch.
- 3D RNT has limited vertical links, i.e. upto 75% of vertical links are cut down than 3D FMT.
- Average switch buffer size and average latency is increased in 3D RNT since interlayer traffic is handled by the limited vertical links.
- In a chip, area required for implementing 3D RNT is increased than 3D FMT.
- 'NN' is an apt traffic pattern for 3D RNT if more intra-layer traffic is required.
- In 3D NoC, hybrid technologies can potentially be integrated; each layer can have different technologies. 'DR' is a suitable traffic pattern if more interlayer traffic is required.

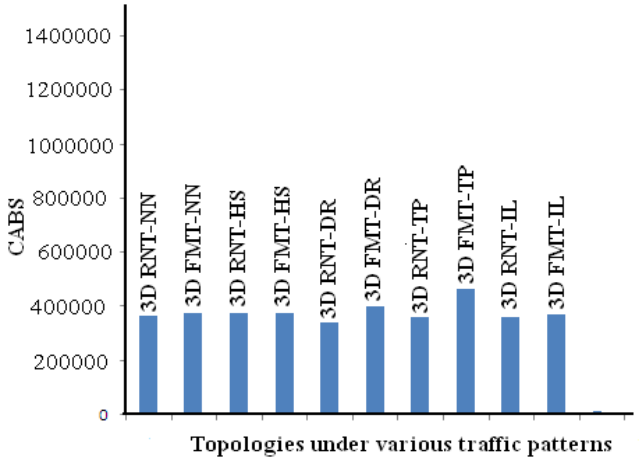

Figure 8. Comparison of CABS of the two topologies under various traffic patterns.

On comparing the proposed topology 3D RNT with 3D FMT, upto 75% of vertical links are cut down in 3D RNT at the cost of around 8% increase in DBS (ps/bytes), but 3D RNT has 10% lesser CABS (sq. µm/bytes) than that of 3D FMT.

3.3. Performance Evaluation using Simulation

In contrast to traditional networks, a NoC has considerably short distance wires (4.5 mm in a 20mm x 20mm chip, for instance) and very large bandwidth (ranging from 8 Gbits/sec to 500 Gbits/sec). This can be realized by setting the link delay and bandwidth attributes of the links accordingly in NS-2; the simulation tool is an open source software, which runs in Linux platform, used for the simulation [2].

The various input simulation parameters that are used in the experiment are:

1. Injected traffic rate is varied between 20 to 100 Gbps.
2. Flit size -10 bytes.
3. Message size - 1024 bytes.
4. Network operating frequency - 250 Mhz.
5. Link delay - 0.1 ps.
6. Number of concurrent connections - 9.
7. Switch buffer size - 200 bytes.

Figure 9 shows the performance comparison of the analytical model with simulation in respect of the traffic patterns 'NN' and 'HS'. It is observed from Figure 9 that an average13%.

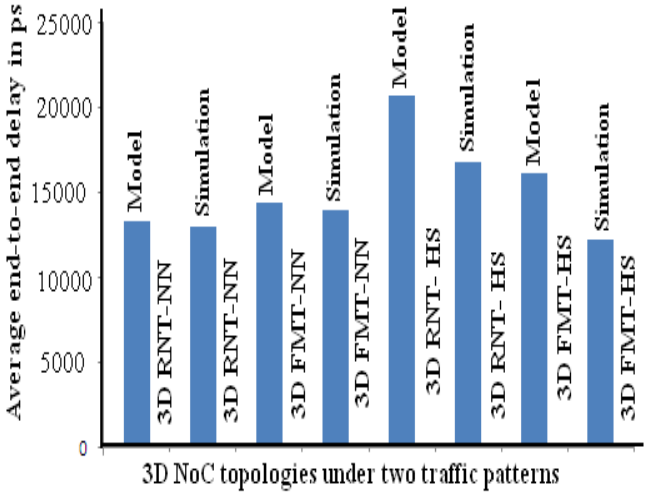

Figure 9. Performance comparison of the analytical model with simulation in respect of end-to-end delay.

End-to-end delay is reduced in simulation on comparing with the model as the simulation uses efficient routing/scheduling algorithm whereas the delay of both the switches and links is fixed in the model. Hence, the results obtained using the model is in the same order of magnitude with the results of the simulation.

4. Programmable Prefix Arbiter Design and Its Performance Analysis

Router is the backbone of the NoC interconnect architectures and its primary function is to forward each flit that arrives on one of its input ports to an appropriate output ports. The router consists of four

components which are input ports, crossbar scheduler, output ports and crossbar switch.

The fundamental operation performed by the crossbar scheduler is to mediate the multiple input requests to access a shared resource and the act of coordinating the access is called as arbitration. A logic circuit that performs the function of arbitration is called as arbiter. An arbiter grants one of multiple incoming requests to an output port based on priority depends on crossbar scheduling algorithms. The priority must be transferred from the highest priority input to next higher priority input if the highest priority input does not have a request [9, 13].

4.1. Design of Two Bits Programmable Prefix Arbiters

In a carry look ahead two bit adder, it is known that the carry c_i of stage i can be determined by using the equation $C_i = G_i \vee (P_i \wedge C_{i-1})$ where g_i is the carry generate bit, p_i is the carry propagate bit and c_i is the carry at stage i [9]. In a PPA design, the carry generate bit g_i is replaced by the priority signal P_i which is also called as priority generate bit and instead of the carry propagate bit P_i, the inverted input request signal R_i is used. The priority transfer signal X_i is computed by using the signals P_i and R_1 as follows. The signal X_i is used to transfer the priority from the highest priority input to next higher priority input if the highest priority input does not have a request.

$$X_0 \leftrightarrow (P_0 R_1).(P_1 R_0) = (P_0 \vee (R_1 \wedge P_1), R_1 \wedge R_0) \quad (6)$$

$$X_1 \leftrightarrow (P_1 R_0).(P_0 R_1) = (P_1 \vee (R_0 \wedge P_0), R_0 \wedge R_1) \quad (7)$$

The priority transfer signals $X_o = P_o\,(R_1 \times P_1)$ and $X_1 = P_1\,(R_o \times P_o)$ can be obtained using Equations (6) and (7). The grant signal G_{r1} is generated by multiplying the input request signal Ri with the priority transfer signal X_i. Figure 10 shows the circuit implementation for the two bit PPA where the priority transfer signal is computed using the input request signal and the generated priority signal.

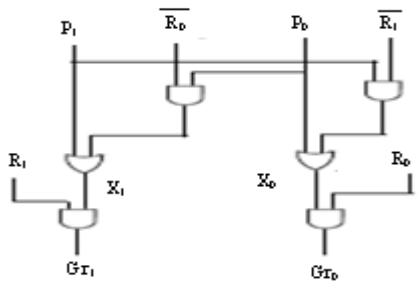

Figure 10. Circuit implementation for two bit PPA.

The two bit PPA can be implemented in a router with two input /output ports. It is observed from the logic circuit design that the priority is transferred from Po to P_1 if input request Ro is inactive and R_1 has an active request.

Further, the priority is transferred from P_1 to P_o as the input request Ro is active. Similarly four, eight and sixteen bits PPA circuits are designed and implemented.

4.2. Synthesis and Implementation of PPA

The circuits designed in the previous section for realizing the PPAs are synthesized in Xilinx ISE 9.2i. Source code for realizing the circuit is written in Verilog HDL. The target technology for the synthesis tool is 3E-XC3S500E-5FT256C. MODELSIM simulator is used to simulate the arbiter and the screen capture of arbiter/req-pr and arbiter/grt-pr which are the sixteen port PPA input, output wave forms respectively, is shown in Figure 11.

In Figure 11, more than one request is applied to the arbiter through sixteen input ports 'arbiter/req-pr' at three different instances. The grant signals 'arbiter/grt-pr' generated from the arbiter at the three different instances are shown.

For instance, sixteenth input port request is inactive at any instance; the priority is transferred to fifteenth input port and the fifteenth input port is selected by the arbiter if it has an active request. The PPA circuit is implemented in Xilinx FPGA Spartan 3E-XC3S500E-5FT256C and the operation of the PPA is verified with the FPGA kit. The input port selection for various requests of a sixteen port arbiter is observed in the kit; the output results of the kit are verified with the results of the MODELSIM simulator output.

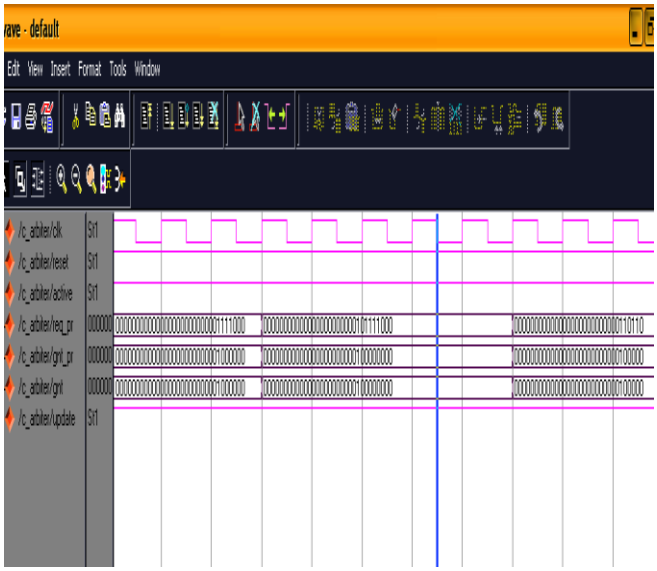

Figure 11. Screen capture of the MODELSIM simulation tool waveform for sixteen port PPA.

Further, with prefix network RRA used in [5] is synthesized by using the same synthesis tool and device. The scaling behavior of the arbiter is evaluated by selecting the arbiter with two, four, eight and sixteen ports. The number of input / output ports of the arbiter is selected based on 2^n where n=1, 2, 3, and 4.

4.3. Performance Analysis of PPA

In this section, the performance of the proposed arbiter with two, four, eight and sixteen ports is analyzed in respect of area (gate count), delay and energy consumption and compared with RRA. The experimental results of PPA and RRA are given in Annexure A and it is observed that the average area (gate count) occupied by the arbiters is reduced by 7% and propagation delay is reduced by 9% in PPA compared to that of RRA.

The reduction of the area and delay in PPA is observed due to 7% lesser number of gates used in the circuit design and 12% increase in operating frequency. Hence it is concluded from the analysis that PPA outperforms RRA in respect of area (gate count), delay and operating frequency at the cost of 2% increase in power consumption.

5. Conclusions

It is exhibited by using VNOC3 simulator that an average 57% chip area, 75% floor plan dead-space and an average of 31% wire length are reduced as a chip single layer is divided into three layers. We develop a 3D NoC topology, 3D RNT and its performance is evaluated using an analytical model and compared with an ideal 3D FMT. It is observed that upto 75% of vertical links are cut down in 3D RNT at the increase of around 8% DBS (ps/bytes) and 3D RNT has 10% lesser CABS (sq. μm/bytes) than that of 3D FMT. The effectiveness of the analytical model is validated with a simulation experiment and an average of 13% end-to-end delay deviation between model and simulation is observed.

PPA is designed and implemented in FPGA. The Performance of the arbiter is analyzed and compared with RRA. PPA occupies 7% lesser area (gate count) and 9% lesser delay than RRA. PPA therefore outperforms RRA in respect of area (gate count), delay and operating frequency at the cost of 2% increase in power consumption.

References

[1] Ababei C., VNOC3, http://venus.ece.ndsu.no dΛak .edu/~cris/software.html, Last Visited, 2009.

[2] Ali M., Welzl M., Adnan A., and Nadeem F., "Using the NS-2 Network Simulators for Evaluating Network on Chips (NoC)," *in Proceedings International Conference on Emerging Technologies*, Peshawar, pp. 506-512, 2006.

[3] Al-Sadi J., Day K., and Khaoua M., "A Fault-Tolerant Routing Algorithm for 3-D Torus Interconnection Networks," *The International Arab Journal of Information Technology*, vol. 1, no. 0, pp. 69-79, 2003.

[4] Bakhouya M., Suboh S., Gaber J., and El-Ghazawi T., "Analytical Performance Comparison of 2D Mesh, WK-Recursive, and Spidergon NoCs," *in Proceedings of IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum*, Atlanta, pp. 1-6, 2010.

[5] Becker D., Efficient Micro Architecture for Network-on-Chip Routers, Thesis, Stanford University, 2012.

[6] Dahule S. and Gaikwad M., "Design and Analysis of Matrix Arbiter for NoC Architecture," *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 1, no. 5, pp. 100-103, 2012.

[7] Dally W. and Towles B., "Route Packets, Not Wires: On Chip Interconnection Networks," *in Proceedings of Design Automation Conference*, Las Vegas, pp. 683-689, 2001.

[8] DePaulo V. and Ababei C., "3D Network -on-Chip Architectures Using Homogeneous Meshes and Heterogeneous Floor Plans," *International Journal of Reconfigurable Computing*, vol. 2010, pp. 1-12, 2010.

[9] Dimitrakopoulos G., Chrysos N., and Galanopoulos K., "Fast Arbiters for On-Chip Network Switches," *in Proceedings of the IEEE International Conference on Computer Design*, Lake Tahoe, pp. 664-670, 2008.

[10] Ebrahimi M., Daneshtalab M., Liljeberg P., Plosila J., Flich J., and Tenhunen H., "Path-Based Partitioning Methods for 3D Networks-on-Chip with Minimal Adaptive Routing," *IEEE Transaction on Computers*, vol. 63, no. 3, pp. 718-733, 2014.

[11] International Technology Roadmap for Semicond uctors (ITRS) 2008, http://www.itrs.net. Last Visited, 2008.

[12] Karypis G., Aggarwal R., Kumar V., and Shekhar S., "Multi Level Hyper Graph Partitioning: Applications in VLSI Domain," *IEEE Transactions on Very Large Scale Integration VLSI Systems*, vol. 7, no. 1, pp. 69-79, 1999.

[13] Suboh S., Bakhouya M., Gaber J., and El-Ghazawi T., "Analytical Modeling and Evaluation of Network-on-Chip Architectures," *in Proceedings of the International Conference on High Performance Computing and Simulation*, Caen, pp. 615-622, 2010.

[14] Tyagi S. and Bohare S., "Review of 3-D Network-on-Chip Topologies," *in Proceedings of World Congress on Information and Communication Technologies*, Mumbai, pp. 783-788, 2011.

[15] Viswanathan N., Paramasivam K., and Somasundaram K., "An Optimized 3D Topology for On-Chip Communications," *International*

Journal of Parallel, Emergent and Distributed Systems, Taylor Francis, vol. 29, no. 4, pp. 346-362, 2014.

[16] Viswanathan N., Paramasivam K., and Somasundaram K., "Performance and Cost Metrics Analysis of a 3D NoC Topology Using Network Calculus," *Applied Mathematical Sciences*, vol. 7, no. 84, pp. 4173-4184, 2013.

Nallasamy Viswanathan received his bachelor degree in electronics and communication engineering from Bharathiar University, Coimbatore, Tamilnadu, India in 1995. He earned his master degree in communication systems and Ph.D. from Anna University, Chennai, Tamil Nadu in 2007 and 2014 respectively. Currently he is working as a professor in the department of electronics and communication engineering, Mahendra Engineering College, Namakkal, Tamil Nadu, India. His research interests include VLSI design, computer networks and Network-on-Chip.

Kuppusamy Paramasivam presently with Karpagam College of Engineering (Autonomous), Coimbatore, India as Professor and Head, ECE Department. He completed UG (BE-ECE with Distinction), PG (ME-AE) from Bharathiar University during 1995 and 1996. He completed his Ph.D. degree (Highly Commended) at PSG college of Technology, Coimbatore, India. He has around 20 years of teaching experience with 13 years of research experience. He has published 78 research papers in Journals and Conferences. His areas of interest are Microprocessors, VLSI Design, Low power VLSI testing and Network on Chip(NOC).

Kanagasabapathi Somasundaram received MS degree in Mathematics from St, Joseph's College, Trichy, India, in 1995 and a doctoral degree in mathematics from Bharathiar University, Coimbatore, India, in 2003. He completed his post doctoral research fellowship from Turku Centre for computer science, University of Turku, Finland, in 2009. Currently he is professor in the department of mathematics, Amrita Vishwa Vidyapeetham, Coimbatore, India. His research interests include linear algebra, graph theory and network on chip

Annexure A

Performance comparison of PPA with RRA

Parameters	RRA Port 2	PPA Port 2	RRA Port 4	PPA Port 4	RRA Port 8	PPA Port 8	RRA Port 16	PPA Port 16
Number of Slice Flip Flops	1	1	2	2	3	3	7	4
Number of 4 input LUTs	3	3	21	19	66	58	144	143
Number of occupied Slices	2	2	11	10	35	30	77	75
gate count for design	29	29	154	139	432	381	944	899
Maximum Frequency in Mhz	467	467	163	241	90	117	75	80
Maximum .Delay in ns	6.34	6.34	8.68	7.85	11.49	9.83	12.31	11.37
Memory Usage in MB	134	134	134	134	135	135	136	135
Power consumption in mW	104	104	93	95	92	95	96	97