Australian Journal of Basic and Applied Sciences, 9(5) March 2015, Pages: 49-54

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Investigation of power consumption in 2D Mesh Interconnect Networks Using Oblivious Routing Protocol in Multicore Processor

[1]M. Lordwin Cecil Prabhaker and [2]A.Gopi Saminathan

[1]Department of Electronics and Communication, Anna University- University College of Engineering, Dindigul, Tamilnadu, India.
[2]Professor and Head, Department of Electronics and Communication, SSMIET, Dindigul, Tamilnadu, India, Phone: 0451 244 8800.

ARTICLE INFO

Article history:
Received 12 November 2014
Received in revised form 26 December 2014
Accepted 29 January 2015
Available online 10 February 2015

Keywords:
Multicore, Interconnect Network, Power consumption, Scheduling Algorithm.

ABSTRACT

Background: Investigation of power consumption in 2D Mesh Interconnect Networks Using Oblivious Routing Protocol in Multicore Processor**. Objective:** Power optimization in Multicore processor is one of the most challenging issues in Multicore/Many core era. This article presents power consumption while using oblivious routing protocol in NxN 2D – Mesh interconnect network in Multicore system. In such system router and linker are major power consumption sources. In the earlier studies the power consumptions sources are modeled and the power is calculated for runtime (dynamic) and ideal time (static). Here the power consumption is calculated with different buffer size (l_b). The system is implemented using event driven simulator with Orion 2.0 power model. The simulator result shown that the router power is increases when the difference between buffer size (l_b) and size of the message(M_S) is less and there is no change in router power when the difference is high. It is also observed that the link power varies depending upon the scheduled link and the dynamic power consumption of linker which is 20% of total power and the remaining power is consumed when the linker is in static condition. **Result and Conclusion:** In this paper, power consumption is calculated while using oblivious routing protocol in NxN 2D – Mesh interconnect network in Multicore system. The system is implemented using event driven simulator with Orion 2.0 power model. The simulator result shown that the router power is increases when the difference between buffer size ($\mathrm{l_b}$) and size of the message$(\mathrm{M_S})$ is less and there is no change in router power when the difference is high. It is also observed that the link power varies depending upon the scheduled link and the dynamic power consumption of linker which is 20% of total power and the remaining power is consumed when the linker is in static condition. In future the power consumption has to calculate for multimedia processing applications like MPEG coder which is implemented on multicore processor.

To Cite This Article: M.Lordwin Cecil Prabhaker and A.Gopi Saminathan, Investigation of power consumption in 2D Mesh Interconnect Networks Using Oblivious Routing Protocol in Multicore Processor ***Aust. J. Basic & Appl. Sci.***, *9(5): 49-54, 2015*

INTRODUCTION

In semiconductor industry, Multicore/Manycore technology is an emerging trend due to device scaling and high performance for complex applications. By increasing operating frequency we can achieve higher performance but it is out dated due to power budget. So the high performance can be achieved through parallelism, while maintaining power at acceptable levels (Anant Balakrishnan and Azad Naeemi, 2001; Maheshkumar, P., Jagtap, 2009). In such technology, the cores are fabricated in a single die. An inter-connection network is used to interconnect cores on the die. In interconnection network the task can be routed using oblivious or adaptive routing protocol. In oblivious routing, the path is completely determined by the source and the destination. Deterministic routing and Dimension order routing is a subset of oblivious routing. To achieve better throughput and latency over oblivious routing, the adaptive routing protocol is used. In which, the routing path for a specific packet is dynamically adjusted depends on network congestion. The complexity of the adaptive routing grows due to balancing the packets between routers and it also requires global knowledge of the current network status, which limits its effectiveness (Michel, A., 2013).. In such system linker and router consumes more power at ideal state and dynamic state. The dynamic and static power consumption can be reduced by hardware approaches (Dynamic Voltage Frequency Scaling (DVFS), Clock Gating (Jungseob Lee, Chi-Chao Wang, 2010) and software approach such as load balancing and task scheduling (Jungsook Yang, 2011; Xing Fu and Xiaorui Wang, 2011).

Corresponding Author: M.Lordwin Cecil Prabhaker**,** University college of Engineering**,** Electronics and Communications Engineering, Dindigul, India and A.Gopi Saminathan, Professor and Head, Department of Electronics and Communication, SSMIET, Dindigul, Tamilnadu, India, Phone: 0451 244 8800

The main contribution of our work is,

- We propose oblivious routing protocol (Weighted XY) for 2D Mesh Multicore Interconnect Network.
- We investigate dynamic and leakage power consumption of router and Link used in 2D Mesh Multicore Interconnect Network
- Finally we evaluate the performance of Interconnect Network on various buffer sizes(l_b) .

In the remainder of the paper, we first outline the system's architecture modal such as router architecture, network link. The section 3 and 4 explains the power modal (dynamic power and leakage power for router and network link) and oblivious routing protocol for NxN Mesh network. The power consumption is calculated for various buffer sizes (l_b) in NxN mesh network using cycle level simulator is explained in section 5.

System model:

The components of multicore system architecture are tile, network link and router, as depicted in Figure 1. (Pengju Ren, 2012; Saurabh Dighe, 2009). The system model is composed of a number of interconnected tiles. As shown in Figure 2, Each tile consist a processing element (PE), a bridge, and the network switch node. The PE's are defined in terms of an MIPS CPU simulator or a script-driven injector or a Pin front-end. The bridge converts packets into flits and helps to interact to PE with the network. The resultant interface exposes for PE is packet-based and for network is flit-based. For connecting neighboring nodes, the router is used. The router specifies ingress and egress port for connecting the neighbor nodes. The above ports are also used to connect injector (CPU core) to the switch. Using the virtual channel buffers (VCs), the ingress port is used to buffer flits to traverse the crossbar to the next-hop. In Multicore architecture for establishing nodes, the pairwise connections with any form of geometry shapes, such as rings, multilayer meshes can be adopted. Based on local traffic conditions, the bidirectional (ingress and egress port) links can optionally change its direction for every cycle.

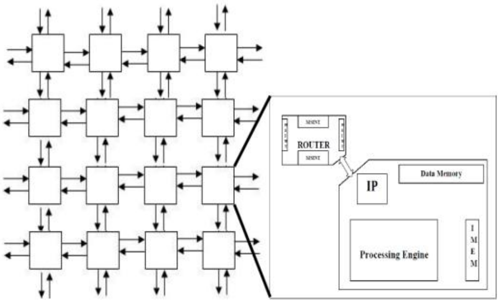


Fig. 1: 2D Mesh - Multicore Architecture.

A $n\,x\,n$ 2D mesh Multicore architecture contains n^2routers. Each router has an address$\{x, y\}$, where x and y belongs to $\{0,1,2, \dots\dots\dots, n-1\}$ in a $n\,x\,n$ mesh. In such topology we defined x increasing along east and $Y-$ increases along north direction. The router architecture consists of one ingress port and one egress port for each neighboring node, as well as for each injector (or CPU core) connected to the switch; each ingress port contains any number of virtual channel buffers (VCs), which buffer flits until they can traverse the crossbar into the next-hop node. If two routers say $u\!: (u_x, u_y)$ and $:(v_x, v_y)$; $\{u_x, u_y, v_x \text{ and } v_y\}$ belongs to $\{0,1,2, \dots\dots\dots, n-1\}$ in a $n\,x\,n$ mesh are connected when the address differs in only coordinate and the difference is equal to 1. In the other words,

$$\{|u_x - u_x| = 1 \qquad ; \; u_y = v_y$$

(or)

$$\{|u_y - u_y| = 1 \qquad ; \; u_x = v_x$$

In Multicore architecture the interconnect networks consumes more power when compare with power utilized by core (Saurabh Dighe, 2009). The interconnect network consists Network link and Router. In our work specifically we are going to investigate link power and router power of interconnect network. To enable power modeling the simulator combines dynamic power model with leakage power model based on ORION - 2.0.

Power Model:

In Multicore architecture power optimization has became one of the major constraints. The first necessary step toward power optimization is power estimation of a system. In such system router and core (Processing Engine) are the power consuming components. For example a high performance router used in Intel 80-core teraflop chip consumes 94% of total power (Andrew, B., 2009). The power consumption in CMOS circuits has two components; they are Leakage power and Dynamic power.

Leakage power essentially consists of the power used when the transistor is not in the process of switching and is essentially determined by the formula

$P_{\ leakage} = I_{leakage} V_{dd}$

The dynamic power consumption is due to charging and discharging of load capacitances and short circuit current while both PMOS and NMOS networks are partially ON. It is formulated as,

$P_{dynamic} = \frac{1}{2} \alpha C V_{dd}^2 f_{clk}$

Where,

$\alpha = Switching\ activity\ factor$
$C = Switching\ capacitance$
$V_{dd} = Supply\ Voltage$
$f_{clk} = Clock\ requency$

So the total power consumption is sum of static power consumption and dynamic power consumption.

$P_{Total} = P_{\ leakage} + P_{dynamic}$

In our simulation the widely used ORION 2.0 power modal is taken. In this modal the power components of a router are FIFO buffers, crossbar switches, arbiters, clocking of router and link between routers (Andrew, B., 2010; Hang-Sheng, 2000). To estimate dynamic power consumption the switching capacitance is calculated by register based FIFO buffers, clocking and physical links. In a deep micron process the leakage power becomes increasingly important as compared to dynamic power. In their modal the leakage architecture is derived by considering effective transistor width, sub threshold leakage current and gate leakage current. The leakage current is measured for variety of circuit components, operating conditions and different V_{th}.

Oblivious routing protocol:

In oblivious routing protocol the information such as conditions of the network, traffic amounts and congestions are not present. Therefore, it chooses the path randomly. The Dimension order Routing (DOR) and deterministic routing algorithms are the subclasses oblivious routing protocol.

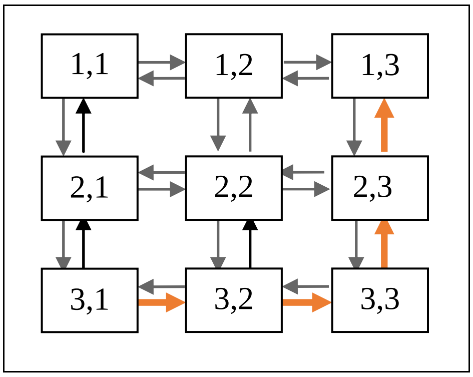


Fig. 2: XY routing for 3X3 Mesh Network.

A. Dimension Order Routing:

The DOR is based on minimal turn model in which the packets are routed by using minimum possible turns.

1) XY Routing:

XY routing is a basic deterministic routing in a Dimension order oblivious routing protocol. In *XY* routing, the flits first traverse in horizontal direction (*X*) to the correct column and then traverse in vertical direction (*Y*) to the receiver as shown in Figure 3.In Figure.3 the flits wants to traverse from *node_7* to *node_3*. For that the direction of *X* increases to reach the correct column then the *Y* direction decreases to reach the destination/receiver.

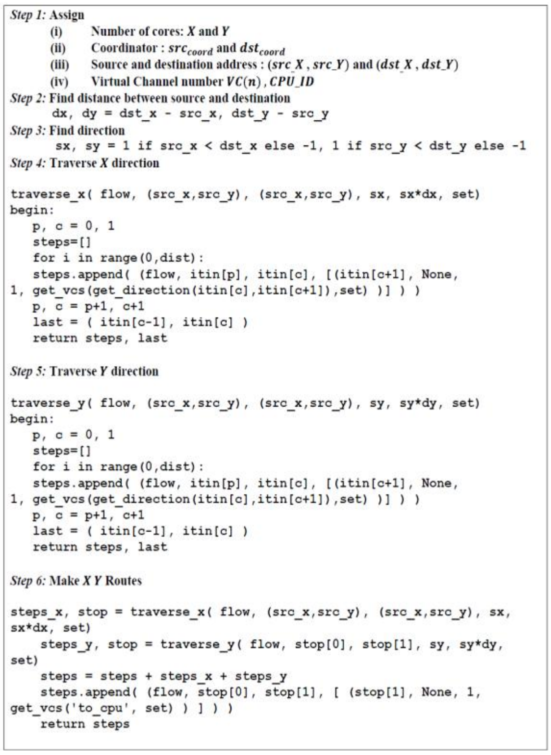
XY Routing is a table based routing model in which each routing entry is stored in the table (data base). According to the destination address the routing table helps the router to route the flits to its destination when it arrives. Based on routing table information the flits make routing decision otherwise flits follows *XY* routing logic. The pseudo code for *XY* routing logic is depicted in Figure 3.

2) Turn Modal:

The turn modal is a systematic routing approach to avoid deadlock in NoC. There are three different approaches available in turn modal, they are 1.West - first Routing, 2. North – last Routing and 3. Negative - first Routing. In west-first routing algorithm the flits are maximum traverse in the west direction. First the flits moving towards west direction will be transmitted and later it is not possible transmitting in west direction. In the second approach, it is not possible for the packets to turn away from north. Thus the packets which need to be routed to north must be transferred there at last. In third approach, the router does not allow the flits to traverse from positive direction to negative direction but it allows the flits to traverse in all other turns. First the router must transfer the flits in negative direction before transmitting the flits in all other direction.

B. *Deterministic Routing Algorithm:*

In deterministic routing algorithm, the router routes the flits using fixed path for every cycle for example the flits from node A to its destination node B will be transferred through certain fixed path. Both regular and irregular network topology can be configured by using this approach. For congestion free network the above mentioned algorithm are reliable and this will have low latency. The flits are always reaches its destination in correct order and the reordering is not necessary so this approach is also suits well for real time systems. Based on routing table information the router sets the fixed path and then the routing information will be broadcast to all other router in the network.



```
Step 1: Assign
        (i)     Number of cores: X and Y
        (ii)    Coordinator : src_coord and dst_coord
        (iii)   Source and destination address : (src_X , src_Y) and (dst_X , dst_Y)
        (iv)    Virtual Channel number VC(n) , CPU_ID
Step 2: Find distance between source and destination
        dx, dy = dst_x - src_x, dst_y - src_y
Step 3: Find direction
         sx, sy = 1 if src_x < dst_x else -1, 1 if src_y < dst_y else -1
Step 4: Traverse X direction

traverse_x( flow, (src_x,src_y), (src_x,src_y), sx, sx*dx, set)
begin:
   p, c = 0, 1
   steps=[]
   for i in range(0,dist):
   steps.append( (flow, itin[p], itin[c], [(itin[c+1], None,
1, get_vcs(get_direction(itin[c],itin[c+1]),set) )] ) )
   p, c = p+1, c+1
   last = ( itin[c-1], itin[c] )
   return steps, last

Step 5: Traverse Y direction

traverse_y( flow, (src_x,src_y), (src_x,src_y), sy, sy*dy, set)
begin:
   p, c = 0, 1
   steps=[]
   for i in range(0,dist):
   steps.append( (flow, itin[p], itin[c], [(itin[c+1], None,
1, get_vcs(get_direction(itin[c],itin[c+1]),set) )] ) )
   p, c = p+1, c+1
   last = ( itin[c-1], itin[c] )
   return steps, last

Step 6: Make X Y Routes

steps_x, stop = traverse_x( flow, (src_x,src_y), (src_x,src_y), sx,
sx*dx, set)
    steps_y, stop = traverse_y( flow, stop[0], stop[1], sy, sy*dy,
set)
    steps = steps + steps_x + steps_y
    steps.append( (flow, stop[0], stop[1], [ (stop[1], None, 1,
get_vcs('to_cpu', set) ) ] ) )
    return steps
```

Fig. 3: *XY* Routing logic.

Simulation environment:

In our simulation the power consumption is evaluated on a nine core Multicore system.

A *3x3* Mesh topology is used to interconnect the cores. Each core consists of process engine, bridge, I/O stream, local memory and registers. For our simulation a chatter message of 256 flits is taken as an input job. The Multicore system is designed with the characteristics as illustrated in table.1on HORNET-1.0 an event driven simulator. The event driven simulator composed in C++ environment and it has the following characteristics topology, CPU, I/O, power modal, router architecture and memory architecture. The mesh topology is configured by giving the height and width of a network. For each simulation, the network was warmed up for 20,000 cycles and then simulated for 100,000 cycles to collect statistics, which was enough for convergence.

Evaluation:

In this section we are going to evaluate the link power and router power for a chatter application. The chatter application consists of 256 flits (8 bytes). The task will be flow from one node another one node as per XY routing topology. The flow between node 3 and node 1 is given below,

```
******
[Flows]
0x000300 = 0:0 1:10 3:4 3:2
0x000301 = 0:1 2:5 3:11 3:3
0x010200 = 1:0 0:6 2:4 2:2
0x010201 = 1:1 3:5 2:7 2:3
0x020100 = 2:0 0:8 1:10 1:2
0x020101 = 2:1 3:11 1:9 1:3
0x030000 = 3:0 1:8 0:6 0:2
0x030001 = 3:1 2:7 0:9 0:3
```

The left part in flow represents the source and destination node identifier and the right side of the flow represents the virtual channel (VC) identifier. In this paper we are going to find the power consumption of chatter application for various buffer sizes. . The dynamic and static link power for various buffer lengths is shown in figure 5. The link power gets increased gradually until the buffer length is 1024 Flits. When buffer length is 4096 Flits the link power goes the peak value due to linker’s dynamic power consumption. After that the link power is not varied even though the increment in buffer length.

The figure 6 shows the overall power consumption on the proposed system for various buffer sizes. The router power of 2D mesh Multicore architecture composed of port, bridge, virtual channel queue, crossbar switch and clock. In router dynamic power consumes approximately 20% of total power and the remaining 80% power consumed due to leakage. The router power for the buffer size 4096 is tabulated in TABLE II. From this table we can understand the port, bridge, virtual channel queue and clock consumes very less power. The average power of these four components is $0.0425\ mW$, but the crossbar switch consumes $0.6603\ mW$. When we consider the overall router power it increases gradually until the buffer length is 1024 Flits. When buffer length is 4096 flits the router power reaches the peak value due to router's dynamic power consumption. After that the router power is not varied despite the increment in buffer length.

Table 1: Multicore system characteristics and configuration

Characteristics	*Configuration*
Number of Cores	6
Interconnect Network	3X3 mesh Topology
Routing Protocol	XY - Oblivious Routing
Queue Size	8 flits
Assigned Message Length (M_L)	256 flits
Bandwidth	
a. CPU	8flits
b. Network	8 flits (2 for each direction)
c. Link bandwidth	1 flit/cycle
VC allocation	Dynamic, EDVCA
VCs per port	4, 8
VC buffer size	4, 8 flits
Warmup cycles	200000 for synthetic traffic, 0 for applications
Analyzed cycles	2000000 for synthetic traffic, full running time for

Table 2: Router power components and it's power consumption for buffer size 4096 flits.

Components	Dynamic	Static
Port	0.00348585	0.054223
Bridge	0.00469594	0.027112
Virtual Channel	0.00818179	0.081335
Crossbar	35428.6	4.54E+06
Switch	135769	804817
Clock	0.0168859	0.004632

Conclusions:

In this paper, power consumption is calculated while using oblivious routing protocol in $N x N$ 2D – Mesh interconnect network in Multicore system. The system is implemented using event driven simulator with Orion 2.0 power model. The simulator result shown that the router power is increases when the difference between buffer size (l_b) and size of the message(M_S) is less and there is no change in router power when the difference is high. It is also observed that the link power varies depending upon the scheduled link and the dynamic power consumption of linker which is 20% of total power and the remaining power is consumed when the linker is in static condition. In future the power consumption has to calculate for multimedia processing applications like MPEG coder which is implemented on muticore processor.

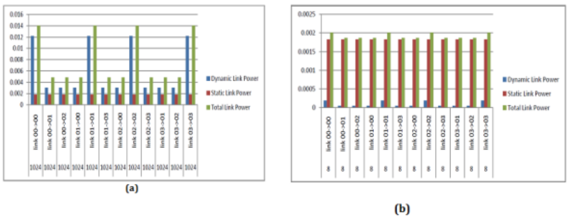


Fig. 4: Dynamic and Static – Link Power for various buffer lengths a) 1024 flits. b) 8 flits.

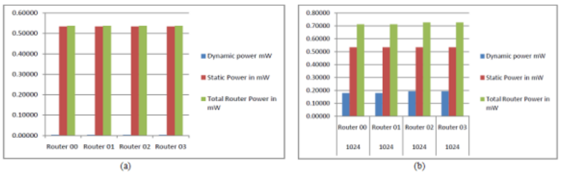


Fig. 5: Dynamic and Static - Router power for various buffer lengths a) 8 flits. b) 1024 flits.

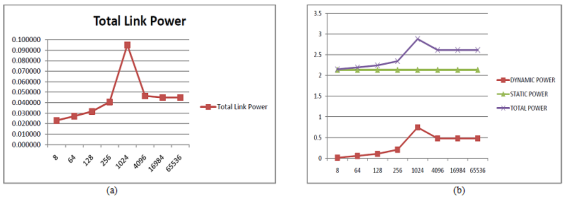


Fig. 6: Total power for 3x3 2d Mesh Multicore System (a) Linker (b) Router.

REFERENCES

Anant Balakrishnan and Azad Naeemi, 2001 Senior Member, IEEE "Interconnect Network Analysis of Many-Core Chips" IEEE Transactions on Electron Devices, 58(9): 2831-2837.

Michel, A., Kinsy, Myong Hyon Cho, Keun Sup Shim, Mieszko Lis, G. Edward Suh, 2013. Member, IEEE, and Srinivas Devadas, Fellow, IEEE, "Optimal and Heuristic Application-Aware Oblivious Routing" IEEE Transactions on Computers, 62(1): 59-63.

David Wentzlaff, 2007. "On-Chip Interconnection Architecture of the Tile Processor" IEEE Computer Society, 27(5) 15-31.

Saurabh Dighe, 2009. "Lessons Learned from the 80 Core Tera-Scale Research Processor" Intel Technology Journal, 13(4): 118- 129.

Mieszko Lis, 2011. "Scalable, accurate Multicore Simulation in the 1000-Core era" IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 175-185.

Jungseob Lee, Chi-Chao Wang et.al, "Workload-Adaptive Process Tuning Strategy for Power- Efficient Multi-Core Processors" ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), pp.225-230, 2010

Jungsook Yang, Chuny Chun, Bagherzadeh, N. Seung Eun Lee, 2011. " Load Balancing for Data-Parallel Applications on Network-on-Chip enabled Multi-Processor Platform", 19th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 436-446.

Pengju Ren, Mieszko Lis, Myong Hyon Cho, Keun Sup Shim, Christopher W. Fletcher, Omer Khan, Member, IEEE, Nanning Zheng, Fellow, IEEE and Srinivas Devadas, Fellow, IEEE, 2012. "HORNET: A Cycle-Level Multicore Simulator", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 31: 890 - 903.

Xing Fu and Xiaorui Wang, 2011. "Utilization-controlled Task Consolidation for Power Optimization in Multi-Core Real-Time Systems" IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 73-82.

Andrew, B., Kahng, Bin Li, Li-Shiuan Peh and Kambiz Samadi, 2009. "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration" Proceedings of the Conference on Design, Automation and Test in Europe, 423-428.

Hang-Sheng Wang Xinping Zhu Li-Shiuan Peh Sharad Malik, 2002. "Orion: A Power-Performance Simulator for Interconnection Networks". Proceedings. 35th Annual IEEE/ACM International Symposium on Micro-architecture, 294-305.

Kariniemi, H., J. Nurmi, 2004. Arbitration and Routing Schemes for On-chip Packet Networks. Interconnect-Centric Design for Advanced SoC and NoC (toim: J. Nurmi, H. Tenhunen, J. Isoaho & A. Jantsch), Kluwer Academic Publishers, 253-282.

Myong Hyon Cho, Mieszko Lis, Keun Sup Shim, Michel Kinsy, Tina Wen and Srinivas Devadas, 2009. "Oblivious Routing in On-Chip Bandwidth-Adaptive Networks" 18th International Conference on Parallel Architectures and Compilation Techniques, '09: 181-190.

Maheshkumar, P., Jagtap, 2009. "Era of Multi-Core Processors" DRDO Science Spectrum, 87-94.

Ville Rantala,Teijo Lehtonen, Juha Plosila, 2006. "Network on Chip Routing Algorithms"TUCS Technical Report, 779.

Michel Kinsy, Myong Hyon Cho, Tina Wen, Edward Suh, Marten van Dijk, Srinivas Devadas, 2009. "Application-Aware Deadlock-Free Oblivious Routing" Proceedings of the 36th annual international symposium on Computer architecture, 208-219.