like to avoid in NoC. Thus, the authors have chosen to focus on contiguous strategy.

A lot of research has been done to increase the efficiency of processor allocation algorithms for 2D-mesh topology, e.g. [6], [26], [30], [31]. Similarly, 2D-torus networks are addressed in [28], [31], [32].

In [29] authors of this paper have presented an idea of hardware implementation of JS and PA, and integrating them on one die together with the processing elements. In [30], new allocation algorithms for 2D-mesh NoC were presented. Additionally, the hardware implementation and synthesis results of the PA for mesh-based CMP were shown. The torus topology was addressed in [28], where allocation techniques and their comparison to mesh schemes were discussed. In [32], synthesis results of the PA driven by the schemes for a torus networks were presented together with their mesh counterparts. Finally in [31], the energy aspects of NoC-based CMP with integrated a PA were explored and energy model for the two components was demonstrated. In this paper, we examine the NoC-based CMP with an integrated processor management system. The most significant NoC architectures are presented in brief. Similarly, the most efficient, bit map allocation algorithms for PAs are described, together with short overview of different techniques. For CMP with the presented components, an experimentation system is proposed and described in detail. The evaluation methodology considers such performance parameters as energy consumption, network resources usage and traffic balance. Based on the presented simulation system, examples of experiments are shown, together with their results.

The reminder of this paper is organized as follows: Section II presents CMP with integrated processor management system and short overview on NoC structures. A PA and allocation techniques are explained in Section III. Evaluation system is presented and described in Section IV, while experimental results are in Section V. Section VI contains final remarks and conclusions.

II. CMP WITH INTEGRATED PA AND JS

In CMPs, interconnects and processing elements are significantly closer than in off-chip multiprocessor systems. It implies better latency and bandwidth performance, but such properties like power, area and cost restrictions become crucial. Also, the complexity of designing efficient and scalable on-chip communication solutions will be increasing together with the number of PEs integrated on a single die. To provide scalability and effective use of resources available in ULSI technology, a tiled architecture is proposed [22] (Fig. 1). The PEs are connected by a NoC that replaces the traditional on-chip buses. The chip area is divided into square tiles. Each tile contains networking elements (router, PE network interface, network channel) and PEs (processor, cache memory, etc.). Each tile is connected to a network Router (R) by PE network interface. Communication among tiles is done by sending messages over the network using tiles' network interfaces (routers). In order to get high performance and efficiency, two components of the processor management system, i.e. the PA and the JS, are proposed to be implemented in hardware and placed as tile on the same die as the PEs in the CMP [29] (Fig. 1).

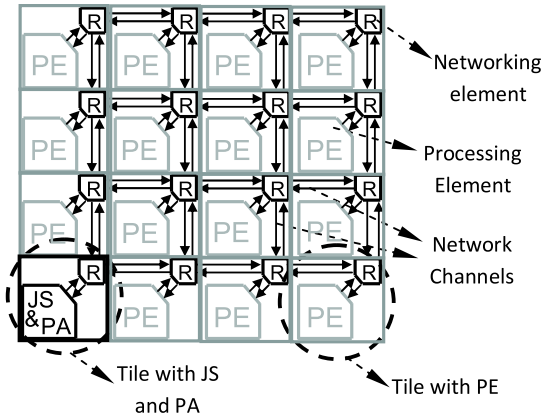

Fig. 1. Tiled CMP (4×4 2D-mesh) with integrated a job scheduler and processor allocator.

JS receives request and places job in the queue, which is controlled in a particular way (in this paper in FCFS fashion). The scheduled job is moved to a PA, which assigns the job to available PEs according to allocation algorithm. As decided by the PA, PEs are reserved and operands are sent to PEs to process them. After execution, PEs return the results and a "release" message is sent to the PA, which updates the status of processors. Operands and results are sent through I/O ports, which for simplicity are not shown in the Fig. 1. All data (control messages, operands and results) are sent by the implemented NoC. The PA and the JS are implemented using the same networking elements (R) like PEs that give communication possibility.

A. NoC's Topology

Topology is one of the main properties that characterizes NoC. It describes the layout and structure of the nodes and links on the chip [12]. The degree of a node decides the number of ports in router. Smaller the value of the degree, lesser the cost, while its homogeneity leads to uniform routers. It implies desire of small and fixed degree. The diameter of a network characterizes the distance between nodes that has direct impact on the latency of the network. Lower latency gives shorter distances that imply a need for smaller network diameter. The topology has significant influence on flow control and routing algorithms. Simple and regular topology reduces the complexity of routing algorithm. An optimal topology is also characterized by its path diversity. Multiple minimal paths among nodes reduce the impact of defects in manufacturing process. Path diversity also allows balancing the load across channels and makes the network more tolerant to faulty channels and nodes. Whatever the topology, the network needs to be laid out in a die. Due to poor progress of "3D stacking" that is still under research, a 2D die is considered. This implies that for networks with dimension higher than 2D, topological adjacency does not lead to spatial adjacency. Thus higher dimensional topologies have negative

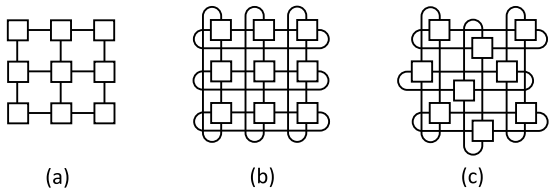

Fig. 2. 2D-mesh and 2D-torus networks: (a) a 3-ary 2-mesh, (b) a 3-ary 2-cube, (c) a folded 3-ary 2-cube.

impact on the wire delay and the wiring density. These are possible reasons for the necessity of long wires. However, implementation of long wires can affect the operating frequency and power consumption. Besides on-chip embedding issues, low dimensional networks represent also better performance in comparison to their higher dimensional counterparts [7], [12], [14].

2D-mesh (k-ary 2-mesh) topology [12], [13] meets all properties described above (Fig. 2a). It is also an obvious and natural choice for tiled architecture due to its close match with the physical layout of the die. 2D-mesh can be enriched with additional wraparound channels that connect the external nodes in each row and column (Fig. 2b). 2D-torus (k-ary 2-cube) topology created in this way is degree regular, has doubled bisection and decreased diameter. Long wraparound links can increase propagation delay, decrease operating frequency and cause using additional repeaters. All these problems can be avoided by folding the torus as shown in Fig. 2c. The folding keeps the topology untouched but eliminates the wraparound channels at the expense of doubling the length of the other channels, however such a doubled channel possesses acceptable latency characteristic and does not require repeaters.

B. Flow Control

Flow control describes allocation of NoC's resources (channel bandwidth, buffer capacity and control state) for packets traversing the network. Flow control policy decides if packet should be dropped, blocked in place, buffered or rerouted. A well designed flow control allocates these resources effectively in order to get good bandwidth and low latency. There are two approaches to flow control: i) Problem of resource allocation – resources have to be assigned to each packet that traverses network (e.g. routing algorithm determines, which resources are allocated to packets), ii) Problem of resolving contention – when an outgoing channel is requested by packets arriving on different inputs, flow control mechanism has to allocate this channel to one packet and do something with the others, e.g. block or drop.

Buffering data is more efficient than waiting to get network resources, like it is in circuit switching [12]. It can be done in units of packets (e.g. cut-through flow control) or flits (e.g. virtual-channel or express-virtual-channel flow control). Flits are fixed sized, the smallest units of information recognized by flow control method, and grouped together to create a variable-length packet. Usage of flits significantly reduces the amount of storage at each node, what saves crucial space on the chip [33].

Virtual-channel flow control [8] logically separates channels that share the same physical channel. It leads to the possibility of existing flits of many packets in the channel. Practically, Virtual Channels (VCs) are flit buffers associated with a single physical channel. By introducing VCs, packets are forwarded in the network over them, that separates allocation of buffers from allocation of channels. The physical channel is blocked only when all its VCs are blocked.

Express-virtual-channel flow control [17] consists of two kinds of virtual channels at each port of a router: 1-hop regular VCs and k-hop Express Virtual Channels (EVCs) that carry flits k-hops at a time. EVCs provide virtual express lanes in the network, which allow bypassing intermediate routers by skipping the router pipeline. The virtual bypassing in a router forwards EVC flits as soon as they reach the router without any buffering and arbitration, that significantly reduces packet latency and router energy consumption. Beside lower latency (up to 84%) and better throughput (up to 23%), EVCs reduce router switching activity, limit number of buffers, and reduce contention that makes this solution energy (up to 38%) and area efficient.

C. Routing

Routing is the procedure of selecting a path from a source node to a destination node in a particular topology. A good routing algorithm balances the load in the network channels, routes paths as fast as possible and is still able to work in the presence of faults. It should also be easily implemented in hardware. There are three main classifications of routing algorithms:

- Deterministic – always choose the same path between a pair of nodes. Balance of load is very poor in this case, but they are commonly used due to easy implementation,
- Oblivious – route packets without considering the network's state (deterministic algorithms are a subset of oblivious),
- Adaptive – use information about the network's state (e.g. channel load information, length of queues for resources, etc.) to make routing decisions.

If the path selected by a routing algorithm is the shortest path between the source and destination, the algorithm is said to be minimal. In a non-minimal routing, the chosen path can be longer, that allows reacting to current network condition.

Another important property of routing algorithms is freedom from deadlocks and livelocks. Livelocks can be avoided by implementing in the packet a field indicating its non-minimal progress. Once the specified value of non-minimal progress reaches a threshold (often called misroute value), only a minimal path can be chosen [12].

Deadlocks can be avoided by eliminating cycles in the resource dependence graph [10]. We can do it by providing some restrictions on routing like it is e.g. in a well known deterministic algorithm - Dimension Order Routing DOR (or xy algorithm) [12]. The DOR sends every packet from source to destination over exactly the same path. A path diversity offered by topology is ignored. Similarly, load balancing and reliability is very weak. These issues were addressed in

Valiant's algorithm [24], where a packet sent from a source to a destination is first sent from the source to a randomly chosen intermediate node and then from that node to the destination. It reduces the load of any traffic pattern. However the good performance given by randomization provides decreased locality. Better load balance can be also achieved by randomizing the order of dimensions in which packet is traversed [12]. At each node either for DOR or Valiant routings x-first or y-first direction is randomly chosen. Such enhanced algorithms are named in this paper as DOR Load Balanced (DOR-LB) and Valiant Load Balanced (Valiant-LB). DOR-LB provides minimal, load balanced oblivious routing and preserves locality.

Another approach to create routing algorithms that are deadlock free is exploiting VCs, that allows the design of highly adaptive algorithms. Very popular are hybrid solutions, which combine splitting network resources (VCs) with restricting the paths for packets. Such algorithms as 3P [21], mesh_route [4] and PFNF [23] are highly adaptive and need only two VCs per physical channel to ensure deadlock freedom.

D. Considered NoCs and Energy Model

In this paper we investigate NoC architectures for CMPs with parameters:

- Topology: k-ary 2-mesh and k-ary 2-cube;
- Flow control: virtual-channel and express-virtual-channel;
- Routing: DOR, Valiant, DOR-LB, Valiant-LB and Adaptive.

The Adaptive routing technique considered in the paper uses historical data of network resource usage in order to route the packet. It is the hybrid, deadlock avoidance technique. In order to avoid livelocks, the misroute value is implemented.

The energy model for NoCs with 2D-mesh and 2D-torus topologies is proposed in [31], where for all considered algorithms, the average energy consumption of sending one bit of data from tile t_i to tile t_j is expressed by:

- For k-ary 2-mesh:

$$E_{bit}^{t_i,t_j} = 0.98(N_{hops}^{VC}+1)+0.23N_{hops}^{EVC}+0.57N_{hops}, \quad (1)$$

- For k-ary 2-cube:

$$E_{bit}^{t_i,t_j} = 0.98(N_{hops}^{VC}+1)+0.23N_{hops}^{EVC}+0.75N_{hops}, \quad (2)$$

where N_{hops}^{VC} is the number of regular VCs traversed by a packet between tile t_i and t_j, N_{hops}^{EVC} is the number of EVCs traversed by a packet between tile t_i and t_j, and N_{hops} is the number of physical channels (number of EVCs + number of VCs − 1) traversed by a packet between tile t_i and t_j.

III. Processor Allocator

Internal architecture of the PA may vary with implemented allocation algorithm that leads also to different I/O structures. The detailed description of the PA structure that is considered in this paper can be found in [30]. An allocation technique used by a PA is most important part of the PA. We consider most effective algorithms for 2D-torus and 2D-mesh systems. More allocation techniques can be found in [6], [19], [26], [28], [30], [31], [32].

A. Processor Allocation - Problem Statement

A 2D-torus topology, denoted by $T(w,h)$, consists of $w \times h$ nodes arranged in a $w \times h$ 2D grid. Each node in the torus refers to an on-chip processor. The node in column c and row r is identified by address $\langle c,r\rangle$, where $0 \leq c < w$ and $0 \leq r < h$. A node $\langle c,r\rangle$ is connected by direct communication channel to its neighboring nodes $\langle c \pm 1, r\rangle$ and $\langle c, r \pm 1\rangle$, where in case if: i) $c = -1$, $c \leftarrow w-1$; ii) $r = -1$, $r \leftarrow h-1$; iii) $c = w$, $c \leftarrow 0$; iv) $r = h$, $r \leftarrow 0$; thus each node has four neighboring nodes.

2D Subtorus: It is a subgrid $T(p,q)$ in the torus $T(w,h)$ such that $1 \leq p \leq w$ and $1 \leq q \leq h$. A job requesting a subtorus $p \times q$ is denoted by $J(p,q)$. A subtorus S is identified by its *base* (lower left node) and *end* (upper right node) and is denoted as $S[\langle x_b, y_b\rangle\langle x_e, y_e\rangle]$.

Busy Subtorus: A busy subtorus β is a subtorus, where all of its nodes have been allocated to jobs. Similarly, a subtorus is free when all of its nodes are free.

Busy Array: A busy array of a torus $T(w,h)$ is a bit map $B[w,h]$, in which element $B[c,r]$ has a value 1 or 0 if node $\langle c,r\rangle$ is busy or free, respectively.

Busy List: A busy list is a set of all busy subtoruses in the system. Similarly, a free list is a set of all free subtoruses in the system.

Base: A node that can be used as a base to allocate incoming job. A base block is a subtorus whose nodes can be used as base for free subtoruses to allocate a job. A set of disjoint base blocks is called the base set.

Coverage: The coverage of a busy subtorus β with respect to a job J is denoted by $\xi_{\beta,J}$ and it is a set of processors such that use of any node in $\xi_{\beta,J}$ as the base of free subtorus for the allocation of J will cause the job J to be overlapped with β. The coverage set with respect to J is denoted by C_J and it is the set of the coverages of all busy subtoruses.

For 2D-mesh topology $M(w,h)$, the definitions are equivalent. Additional terms used only in meshes are:

Reject Area: The reject area with respect to a job J, denoted by R_J, is a set of processors such that use of any node in R_J as the base of free submesh for the allocation of J will cause the job J cross the boundary of the mesh.

Sink: The sink of the reject area is the processor with coordinates $\langle w-p+1, h-q+1\rangle$.

An illustration and detailed description of the given terms can be found in [6], [26], [28], [30], [31], [32].

For a given job J, C_J represents the set of processors, which can not be the base of the free subtoruses (for meshes it is $C_J \cup R_J$). Thus the base set for J is $Z - C_J$ (for a mesh it is $Z - C_J - R_J$), where Z refers to the set of all processors in the system. It is important to note, that the base set with respect to $J(p,q)$ is different from this with respect to $J(q,p)$, when $p \neq q$.

B. Allocation Algorithms

1) Improved First Fit (IFF): The IFF algorithm [30] is the approach with a bit map representing the allocation status of processors in the mesh. The bit map idea has also been used in [27], where the First Fit (FF) and the Best Fit (BF) techniques are presented. In the IFF with respect to an incoming job $J(p,q)$, the busy array B (without considering reject area R_J) is scanned in to create a coverage array C_T, which is a bit map representing the coverage set. In order to form the C_T in an efficient way, each coverage $\xi_{\beta,J}$ is divided into three regions: job coverage, left coverage and bottom coverage. Then, two scans are necessary: 1) All rows from right to left (determining a job and left coverages); 2) All columns (left to right) from top to down (creating a bottom coverage). The first available node that does not belong to the coverage set is returned and can serve as a base for the job J. The IFF strategy is recognition complete by considering two job orientations: If allocation of $J(p,q)$ fails and $p \neq q$, then $J(q,p)$ possibility is checked.

With the IFF allocation, the achieved time complexity is $O(wh)$. Deallocation of processors in the systems with a busy array reduces to clearing all the elements in bit map B. It is done also in $O(wh)$.

2) Bit Map Allocation for Torus (BMAT): The BMAT technique [28] is developed for 2D-torus systems. It is based on the bit map approach used in the IFF algorithm. With respect to an incoming job, the busy array B is scanned to create a coverage array C_T in the form of bit map. The methodology of creating coverage array C_T is similar to IFF scheme, however due to lack of reject area R_J in the torus networks, all nodes have to be considered. Also similarly to IFF, each coverage $\xi_{\beta,J}$ is divided into three regions: job coverage, left coverage and bottom coverage. However, instead of two scans required by IFF, the BMAT needs four passes through C_T: 1) All rows from right to left, two times each (determining a job and left coverages); 2) All columns (left to right) from top to down, two times each (creating a bottom coverage). These additional two scans are due to wraparound channels in a torus and they are obligatory.

The BMAT technique is recognition complete by manipulating the job orientation. If for a given $J(p,q)$ the allocation fails and $p \neq q$, the scheme will change the orientation of the job and then $J(q,p)$ possibility is checked. When both attempts fail, the allocation of the job fails.

The time complexity of the BMAT for allocation and deallocation is $O(wh)$.

3) Other Allocation Schemes: The other interesting allocation algorithms (for k-ary 2-meshes and k-ary 2-cubes) replace a bit map with allocation status of cores by a list with busy subgrids (list of processors that are busy). That strategy can be found e.g. in IAS, ISBA and IQA techniques [6], [30] for 2D-meshes or in BLAT, SAT and SBAT schemes [28] for 2D-toruses. However, as it is reported in [30] and [32], the algorithms are hardly synthesizable and they consume a lot of logic. Moreover, simulations results [28] and energy estimation [31] show that the performance of these algorithms is not so bright. Thus in this paper, busy list algorithms are not considered.

Another approach to the processor allocation problem is not keeping in memory, information about subgrids that are busy, but maintaining a list with free subgrids (keep list of processors that can be allocated for a requested job). Such approach is taken, e.g. in the FSL and CFL algorithms [1], [30]. However, a synthesis of free list techniques is even more difficult than busy list schemes [30]. Moreover, based on computer simulations [1], [26], they are not performing better than other techniques. Thus, similarly to the busy list algorithms, the free list schemes are not considered in this paper.

C. Energy of Processor Allocator

A synthesis of PAs based on allocation algorithms for 2D-meshes and 2D-toruses is presented by the authors of this paper in [30] and [32], respectively. Synthesis of the PA is done for Altera's Stratix III family device EP3SL150F780C2. The presented results contain such parameters as the maximum frequency f_{max}, logic utilization, number of registers used and number of combinational ALUTs needed. Based on these results, the energy consumed in a cycle for PA is estimated in [31]. The results are used in this paper to describe the energy consumed by PA.

IV. CMP - Evaluation System

The goal of creating an evaluation system for CMP is to examine the NoC-based CMP with an integrated PA, without time consuming and costly physical implementation. The structure of the system allows testing of the PA driven by all processor allocation techniques mentioned in this paper (Section III). In the testing environment, the PA can be connected with other nodes by the NoCs, which were analyzed in the Section II. The testing environment provides the possibility to test many NoC configurations. The experimentation system is characterized by its modular design that allows complex input-output tests for subsystems. The concept of the system, its design and implementation is done in such a way to make the experimental environment as close as possible to the real CMP.

A. Proposed Structure of the System

A physical structure of the system is based on the concept presented in Fig. 1. The logical modules of the system and information flow are presented in Fig. 3. Each module is described in next section.

B. Description of Modules

1) Parameters: It contains global parameters, common for three modules: Data Generator, PA and NoC-based CMP. The global parameters are:

- P_{G1}: denoted by w – horizontal size of the mesh/torus,
- P_{G2}: denoted by h – vertical size of the mesh/torus,
- P_{G3}: defines, which topology is evaluated. In our system, two topologies are supported: 2D-mesh and 2D-torus.

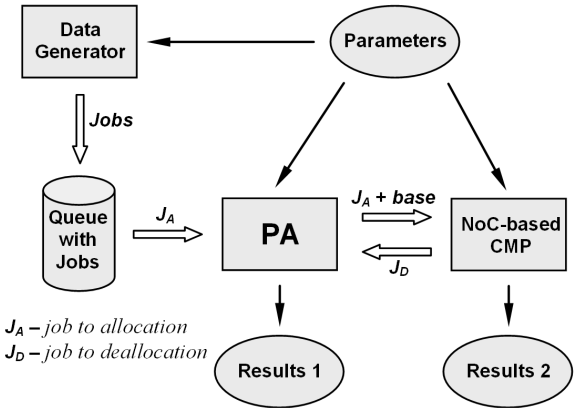

Fig. 3. The logical organization of the experimentation system.

2) Results 1: The allocation algorithms quality criteria are collected in the module Results 1. The evaluated criteria are described in [28], and they are:

- E_1: denoted by t_a – time of allocation,
- E_2: denoted by t_p – processing time,
- E_3: denoted by L – system load,
- E_4: denoted by F_e – external fragmentation.

3) Results 2: In this module, NoC utilization is recorded. The NoC utilization criteria are:

- E_5: denoted by $R_{VC\langle x,y\rangle}$ – VC count, defined for virtual channels for each router in a network, where $\langle x, y\rangle$ is a network address of the router. The $R_{VC\langle x,y\rangle}$ contains information, how many packets are transmitted through VCs in the router $\langle x, y\rangle$, e.g. if twenty packets are transmitted through router with address $\langle 3, 3\rangle$ using its VCs, the $R_{VC\langle 3,3\rangle}$ is equal 20,
- E_6: denoted by $R_{EVC\langle x,y\rangle}$ – EVC count, defined for express virtual channels for each router in a network, where $\langle x, y\rangle$ is a network address of the router. The $R_{EVC\langle x,y\rangle}$ contains information, how many packets are transmitted through EVCs in the router $\langle x, y\rangle$,
- E_7: denoted by $T_{R_{VC}}$ – total VC count, contains VC count value for all routers in the network, and it is defined as:

$$T_{R_{VC}} = \sum_{i=0}^{i=w-1} \sum_{j=0}^{j=h-1} R_{VC\langle i,j\rangle} \tag{3}$$

- E_8: denoted by $T_{R_{EVC}}$ – total EVC count, contains EVC count value for all routers in the network, and it is defined as:

$$T_{R_{EVC}} = \sum_{i=0}^{i=w-1} \sum_{j=0}^{j=h-1} R_{EVC\langle i,j\rangle} \tag{4}$$

4) Data Generator: Based on global (P_{G1} and P_{G2}) and local parameters, the Data Generator module generates the queue of tasks to allocate for the considered CMP. The logical structure of the Data Generator module is described by the relation $O = R(P)$ and is presented in Fig. 4. The elements of the sub-system are:

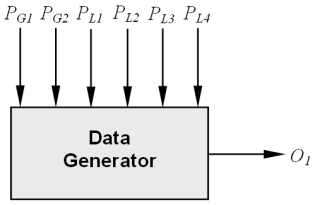

Fig. 4. Block-diagram of the Data Generator module as input-output system.

- Problem parameters:
  - P_{G1}, P_{G2}: global parameters,
  - P_{L1}: local parameter – number of jobs created by generator,
  - P_{L2}: local parameter – maximum horizontal p and vertical q size of the generated job $J(p, q)$,
  - P_{L3}: local parameter – maximum execution time for job J in PEs,
  - P_{L4}: local parameter – type of distribution, an element of the set $\{$normal, uniform$\}$.
- Output O_1: queue of jobs. Each job is characterized by: job number, horizontal and vertical size and required execution time.

The generator can produce normally and uniformly distributed pseudo-random numbers. For the size of a job $J(p, q)$ and the execution time of the job, generated numbers are between 1 and local parameters P_{L2} and P_{L3}, respectively. The job number is assigned according to the job position in the queue (first job has number 1, second 2, etc.).

5) Queue with jobs: The queue is the result of Data Generator module. The queue of jobs is generated once before experiment. The jobs in the queue are ordered in FCFS fashion and are passed on to the PA module as such.

6) PA - processor allocator: The physical structure of the PA is described in [30]. The PA determines the base for a job supplied by the Queue module. The logical structure of the PA module is described by the relation $(O, E) = R(A, P)$ and presented in Fig. 5. The elements of the sub-system are:

- Inputs:
  - A_1: job to allocate, supplied by the module Queue. The job is characterized by: job number, horizontal and vertical size and required execution time,
  - A_2: job to deallocate, supplied by the module NoC-based CMP. The job is characterized by its size and a base (bit map case) or job number (busy list).

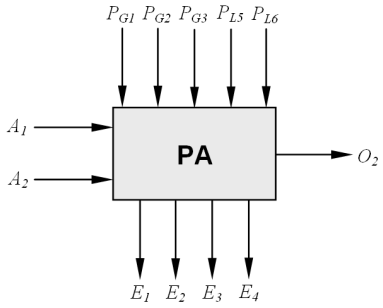

Fig. 5. Block-diagram of the PA module as input-output system.

- Problem Parameters:
  - P_{G1}, P_{G2}, P_{G3}: global parameters,
  - P_{L5}: local parameter – processor allocation algorithm used in order to determine a base,
  - P_{L6}: local parameter – address $\langle x, y \rangle$ of the PA in the network. A node with address of the PA can not be a PE for jobs.
- Outputs:
  - O_2: job to allocate with a base determined by the PA, characterized by job number, size and execution time,
  - E_1, E_2, E_3, E_4: evaluation criteria described in the Results 1 module.

7) NoC-based CMP: This module is simulating the NoC of the CMP. The physical structure was discussed in Section II. The logical structure of the NoC-based CMP module is described by the relation $(O, E) = R(A, P)$ and presented in Fig. 6. The elements of the sub-system are:

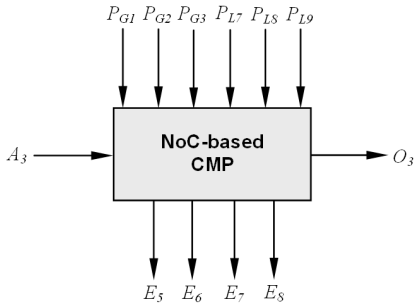

Fig. 6. Block-diagram of the NoC-based CMP module as input-output system.

- Input A_3: is a job with a base. Based on the A_3, the module is able to determine, which PEs can be reserved for the job. Reservation is done by sending allocation packets from the PA to each processor required by the job,
- Problem parameters:
  - P_{G1}, P_{G2}, P_{G3}: global parameters,
  - P_{L7}: local parameter – a flow control implemented in the network and number of nodes that can be virtually bypassed using EVCs. For virtual-channel flow control, P_{L7} is set to 0 (zero nodes can be bypassed). Values greater than 0 mean that considered flow control is an express-virtual-channel with number of bypassed nodes specified by the parameter,
  - P_{L8}: local parameter – a routing algorithm that determines the route of the packet, being an element of the set {DOR, Valiant, DOR-LB, Valiant-LB, Adaptive},
  - P_{L9}: local parameter – indicates an allowed number of misroutes (misroute value). The parameter is valid only in case, when the P_{L8} is set to "Adaptive".
- Outputs:
  - O_3: job to deallocate with its size and the base,
  - E_5, E_6, E_7, E_8: NoC utilization criteria described in the Results 2 module.

C. Implementation of the System

All modules of the experimentation were developed in C. The generator returns the queue of jobs as a text file. The queue is processed in the next step by the PA module. The results generated by the PA and NoC-based CMP modules are in the form of text and excel files, to make further data analysis easier.

The PA takes jobs from the Queue, job after job and tries to find free PEs in order to allocate the job. If such free PEs exist, the PEs are allocated for the job (the job is sent together with a base to the NoC-based module). If there is no free PEs, the PA waits until another job will release some PEs (FCFS fashion). In order to find free PEs for a job, the PA module uses one of the allocation schemes, presented in Section III of this paper. Thus, all mentioned processor allocation algorithms can be tested.

If there is enough free PEs for a job, the PEs can be allocated. An allocation process is done by the NoC-based CMP module. In order to allocate PEs, the allocation message has to be sent from a PA to each PE assigned to a job. It is assumed, that this message takes one flit, e.g. if job requires 6 processors, 6 flits have to be sent from a PA to all PEs assigned to the job. Similarly, if a job is done, a deallocation message (that takes also one flit) has to be sent from each involved PE to the PA. Thus, the PA is updated and the just-released PEs can accommodate another job. The evaluation system is proposed to analyze a PA and traffic generated by that PA, so, only allocation and deallocation messages are considered. In a real system, a lot of different packets are sent among nodes, e.g. just after allocation, messages with commands and operands have to be sent to all the relevant PEs. Similarly while processing, the PEs can exchange control messages and data among themselves. When processing is done, result messages have to be sent as well. However, in this experimentation system, all messages other than allocation and deallocation message are passed over.

The messages are sent using NoC. In the described experimentation system, NoCs with parameters presented in Section II can be evaluated.

As it was mentioned at the beginning of this Section, the module responsible for NoC was written in C language, which is sequential. It provided higher implementation flexibility over concurrent languages (like e.g. SystemC) without decreasing the quality of this paper. Many NoCs have already been implemented, e.g. [3], [5], [15], [25], and all performance and implementation aspects have been described. Thus in this experimentation system, the exact analysis of the NoC parameters, like performance of routing algorithms, analysis of deadlock and livelock occurrences, timing issues, etc., are not included. Instead of focusing on parameters of NoC, the evaluation system performs detailed analysis of the PA. For the analysis of the PA and its impact on the NoC, the sequential simulation is convenient and adequate. The chosen implementation had an impact on implemented routing algorithms, where deadlock situation could not occur. The DOR, Valiant, DOR-LB and Valiant-LB are oblivious techniques, thus the livelock can not occur as well. The Adaptive routing

technique employed in the system uses historical data of network resource usage in order to route the packet. A physical channel that is used not so often has a priority over the channel used very intensively. As in earlier mentioned techniques, the deadlock situation in the Adaptive algorithm case can not occur. However, the livelock situation is possible. In order to avoid livelocks, the misroute value (which can be defined in the simulation system) is implemented.

The presented experimentation system allows exact analyzing of PA behavior, based on all algorithms mentioned in this paper. Additionally, for the presented NoC solutions, investigation of the traffic generated by PA is also possible. Together with the energy model proposed in [31], an energy consumption of the traffic can be determined as well. The synthesis results presented in [30] and [32] provide the energy utilization by the PA that allows complete energy analysis of the processor allocation for the NoC-based CMP.

V. Experimental Results

The CMP simulator was run on the on Intel Pentium 4 machine (2×3GHz processor) with 2 GB of RAM. According to conclusion from Section III, the final experiments were performed for allocation techniques based on bit map, i.e. the IFF and BMAT algorithms. For the IFF algorithm, the free PEs allocated to a job are always adjacent like in mesh topology. However, such adjacent PEs can also communicate among each other and the PA using the torus topology. Thus, for the PA driven by the IFF algorithm, we consider two cases:

- IFF-mesh – where PEs adjacent like in the mesh communicate between each other using mesh-based NoC,
- IFF-torus – where PEs adjacent like in the mesh communicate between each other using torus-based NoC.

The BMAT technique finds free PEs for the requested job based on the torus topology, thus the neighboring PEs allocated to a job can be adjacent by using the wraparound channels. In order to exchange messages, the PEs could use a mesh-based NoC. However, in such case we could destroy locality of the PEs allocated to one job. Additionally, jobs allocated to PEs using wraparound channels would not be contiguous, if a mesh-based NoC would be used. In this paper, the contiguous allocation strategies are considered, thus for the PA based on the BMAT strategy, we consider only a torus-based NoCs as the communication medium.

A. Routing Algorithms and Topology Comparison

In this experiment we analyzed five routing algorithms: DOR, Valiant, DOR-LB, Valiant-LB and Adaptive. The objective of this experiment is to evaluate the routing techniques and determine, which routing technique and what PA used in the CMP provide the best energy-performance characteristic. In the experiment, queue of jobs is generated using the Data Generator module with problem parameters: i) P_{G1}: 10, ii) P_{G2}: 10, iii) P_{L1}: 1000, iv) P_{L2}: $\lceil 0.4 * w \rceil$ and $\lceil 0.4 * h \rceil$, v) P_{L3}: 500, vi) P_{L4}: normal.

Jobs from the queue are allocated by the PA configured with IFF and BMAT algorithms. The jobs are allocated to PEs using NoC driven by one of the five presented routings. NoC is tested for all described routings. For each routine, the same job queue is applied. In the experiment, the PA and NoC-based CMP modules are configured as follows: i) P_{G1}: 10, ii) P_{G2}: 10, iii) P_{G3}: {2D-mesh, 2D-torus}, iv) P_{L5}: {IFF, BMAT}, v) P_{L6}: $\langle 0, 0 \rangle$, vi) P_{L7}: 0, vii) P_{L8}: {DOR, Valiant, DOR-LB, Valiant-LB, Adaptive}, viii) P_{L9}: {1, 2, 3, 4}.

The obtained energy results are collected in Table I, where energy of PA, NoC and the total energy are presented. An Adaptive-Mx algorithm in the table represents the Adaptive algorithm with a maximum number of x misroutes. The presented energy results are calculated according to formulas (1) and (2), and based on the results of synthesis presented in [30] and [32]. During calculations it was assumed that the width of one flit is 32 bits, which is the most popular width used in research [20].

The largest amount of energy was consumed in mesh-based NoC case (IFF-mesh case in the Table I). Lack of wraparound channels in mesh topology makes longer routes that increases the number of routers used in transmission together with the amount of energy needed to send a message.

The PA with BMAT allocation strategy consumed significantly (6 times) more energy than IFF scheme. It is a price for wraparound channel recognition offered by BMAT. Among NoCs considered in the experiment, the NoC with the DOR routing technique achieves the lowest energy usage. It is not a surprise because DOR is the easiest possible algorithm and it routes packets through minimal paths. The Valiant algorithm requires a high amount of energy to route the traffic. Even the most advanced Adaptive routing algorithm with three misroutes allowed, achieves better energy performance.

In Fig. 7 and Fig. 8, the VC count $R_{VC\langle x,y \rangle}$ of each router with DOR, DOR-LB and Adaptive-M3 routing techniques is shown. Both figures present NoC, where the PA with IFF allocation scheme is implemented, but in Fig. 7 the 2D-mesh topology is employed, while in Fig. 8 it is the 2D-torus. The traffic was better balanced for torus-based NoCs: IFF-torus (Fig. 8) and BMAT (not shown). The balancing of traffic has significant impact on the thermal aspects of chip utilization. If traffic has a good balance, routers used are spread across the whole chip. In such case, it is much easier to deal with heat dissipation than in the case, where there is weaker traffic balance (Fig. 7). In both the figures, the weak traffic balance for DOR routing can be noticed. However, this disadvantage can be compensated by very low energy consumption (Table I). Low traffic balance provided by DOR can be eliminated by load balance extension (DOR-LB case in the figures). The DOR-LB brings significant balance improvement, moreover, it remains still minimal and still needs the smallest amount of energy. Among all the considered routing techniques, the Adaptive scheme is characterized by the best traffic balance and together with its energy performance becomes very attractive, especially in comparison to Valiant techniques.

The experiment reveals the enormous advantages of IFF-based PA with torus-based NoC. The IFF-Torus approach is characterized by very good energy performance, moreover, torus topology of NoC ensures better traffic balance than mesh. This solution in conjunction with DOR-LB routing technique gives very good energy-balance characteristic. The BMAT-

TABLE I
ENERGY CONSUMPTION OF THE PA, NOC AND TOTAL CMP DEPENDING ON THE USED ROUTING ALGORITHM

	PA Energy [μJ]		**NoC Energy [μJ]**			**Total CMP Energy [μJ]**		
	BMAT	***IFF***	***BMAT***	***IFF-Mesh***	***IFF-Torus***	***BMAT***	***IFF-Mesh***	***IFF-Torus***
DOR	1633.126	256.854	4.223	6.28	4.252	1637.349	263.135	261.106
Valiant	1633.126	256.854	7.535	10.16	7.578	1640.661	267.014	264.432
DOR-LB	1633.126	256.854	4.223	6.28	4.252	1637.349	263.135	261.106
Valiant-LB	1633.126	256.854	7.578	10.173	7.586	1640.698	267.027	264.44
Adaptive-M1	1633.126	256.854	5.397	7.415	5.413	1638.523	264.269	262.267
Adaptive-M2	1633.126	256.854	6.531	8.601	6.568	1639.657	265.455	263.422
Adaptive-M3	1633.126	256.854	7.525	9.658	7.579	1640.651	266.512	264.433
Adaptive-M4	1633.126	256.854	8.536	10.727	8.553	1641.662	267.58	265.407

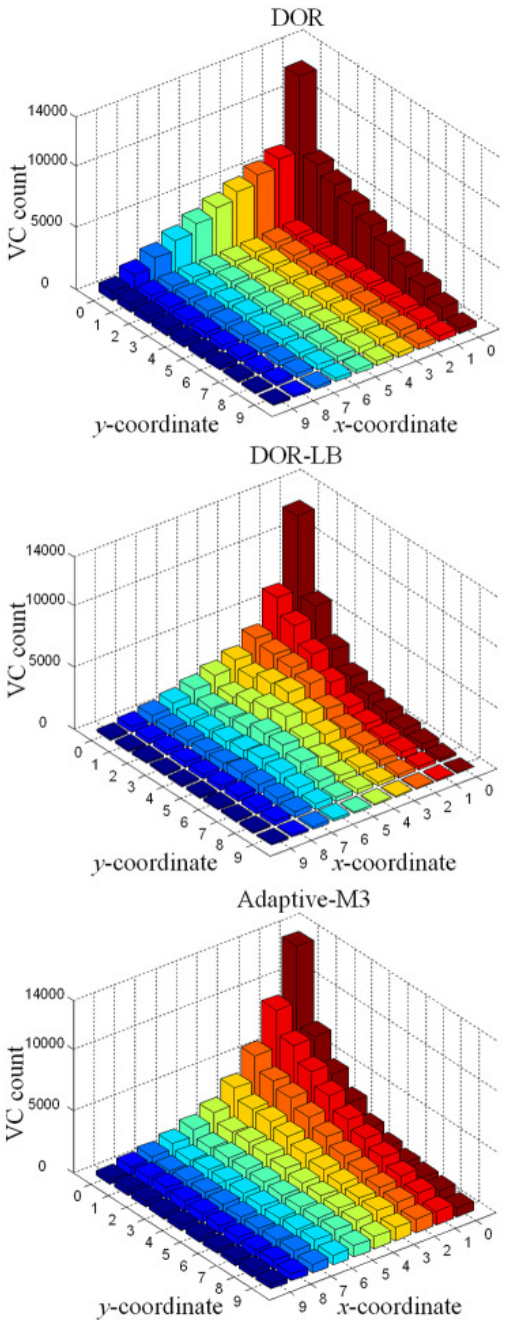

Fig. 7. VC count of each router in the network in IFF-Mesh case for DOR, DOR-LB and Adaptive-M3 routing algorithms.

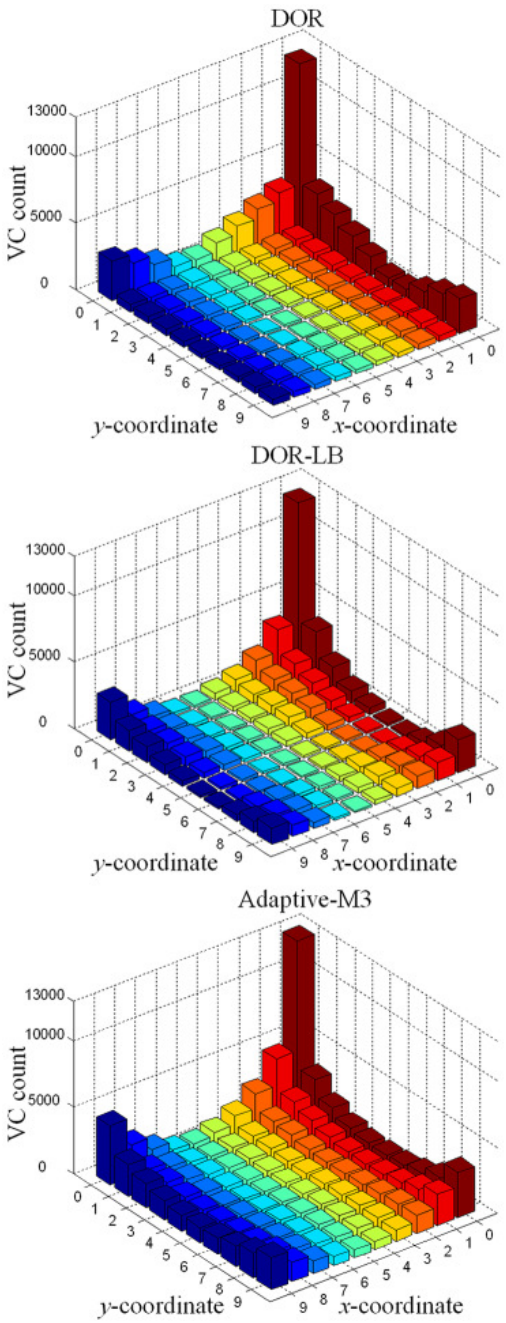

Fig. 8. VC count of each router in the network in IFF-Torus case for DOR, DOR-LB and Adaptive-M3 routing algorithms.

based PA demonstrates very high energy consumption that makes this solution less attractive.

B. Impact of Express Virtual Channels

In this experiment, the express-virtual-channel flow control is implemented and compared with its virtual-channel counterpart. It is done by enriching routers in the experimentation systems by express buffers. As in previous experiments, the same queue of jobs has been taken. A PA with IFF and BMAT allocation algorithms is employed for the queue. For all combinations, a NoC with all five routing algorithms is investigated. For each routing technique, two flow control mechanisms (virtual-channel and express-virtual-channel) are implemented. In this experiment, the PA and NoC-based CMP modules are configured as follows: i) P_{G1}: 10, ii) P_{G2}: 10, iii) P_{G3}: {2D-mesh, 2D-torus}, iv) P_{L5}: {IFF, BMAT}, v) P_{L6}: $\langle 0, 0 \rangle$, vi) P_{L7}: {0, 1, 2, 3}, vii) P_{L8}: {DOR, Valiant, DOR-LB, Valiant-LB, Adaptive}, viii) P_{L9}: {1, 2, 3, 4}.

The results of experiment are presented in Table II. The length of EVC in the table represents the parameter P_{L7} – number of routers that can be virtually bypassed by EVC. As it can be noticed, implementation of EVCs for all considered routing techniques decreases the amount of energy used. Especially for mesh-based NoC, the saving offered by express-virtual-channel flow control is significant. The best achieved results are for cases where the number of VC buffers used is reduced by employing more express buffers. Thus, if more express buffers are used, the gain in energy saving is higher. The number of express buffers used depends on the length of EVC, size of mesh/torus and size of jobs. For the NoC under consideration, in almost all cases the higher EVC usage and energy saving is achieved for P_{L7} equals 2 – one express channel bypasses virtually two nodes. In few cases for IFF-Mesh instance (DOR, DOR-LB, Adaptive-M1), configuration with P_{L7} equals 3 delivered better results.

In all the investigated cases, implementation of express-virtual-channel flow control brings energy saving to make it the clear choice for modern CMPs. Length of EVCs had impact on the amount of energy saved and its choice has to be made individually for each system.

The advantage of EVCs can be observed especially for NoC with mesh topology, where the saving is the highest. However, even with significant benefits offered by express buffers in meshes, the torus topology provides better energy characteristic in all considered cases.

VI. Conclusions

We have proposed an experimental scheme to evaluate the NoC-based CMP with integrated processor management system. The system can analyze many NoCs that differ in topology, routing algorithm and flow control. Similarly, PAs with many allocation algorithms can be investigated. The suggested system considers energy efficiency and traffic characteristic as decision making criteria. Together with the proposed evaluation system, most important approaches to both, NoC and PA have been described. The proposed simulation system has been employed to conduct experiments, results of which have been presented and discussed.

The outcomes confirmed the good results of CMP with PA driven by IFF algorithm, while BMAT turned out to be very energy inefficient. However, for NoC, the best load balance and energy consumption are achieved for the torus network. This led to the idea of implementing the PA driven by IFF with torus topology of the NoC. This approach brought the best effects. The IFF-based PA and torus-based NoC driven by DOR-LB routing with express-virtual-channel flow control delivered the best load balance and energy characteristic. If higher reliability and load balance are needed, the adaptive routing technique with carefully chosen parameters can also be a very good idea.

References

[1] I. Ababneh, "An efficient free-list submesh allocation scheme for two-dimensional mesh-connected multicomputers," *Journal of Systems and Software*, vol. 79, no. 8, pp. 1168–1179, 2006.

[2] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP onchip networks," in *Proceedings of 20th International Conference on Supercomputing*, 2006, pp. 187–198.

[3] T. Bjerregaard, "The mango clockless network-on-chip: Concepts and implementation," Ph.D. dissertation, Technical University of Denmark, 2005.

[4] Y. M. Boura and C. R. Das, "Efficient fully adaptive wormhole routing in n-dimensional meshes," in *Proceedings of the 14th International Conference on Distributed Computing Systems*, 1994, pp. 589–596.

[5] S. Bourduas, "Modeling, evaluation, and implementation of ring-based interconnects for network-on-chip," Ph.D. dissertation, McGill University, 2008.

[6] G. Chmaj, D. Zydek, and L. Koszalka, "Comparison of task allocation algorithms for mesh-structured systems," in *Computer Systems Engineering Theory & Applications, 4th PBW*. IEE Control and Automation Professional Network, 2004, pp. 39–50.

[7] W. J. Dally, "Performance analysis of k-ary n-cube interconnection networks," *IEEE Transaction on Computers*, vol. 39, no. 6, pp. 775–785, 1990.

[8] ——, "Virtual-channel flow control," *IEEE Transaction on Parallel and Distributed Systems*, vol. 3, no. 2, pp. 194–205, 1992.

[9] W. J. Dally and C. L. Seitz, "The torus routing chip," *Journal of Distributed Computing*, vol. 1, no. 4, pp. 187–196, 1986.

[10] ——, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Transactions on Computers*, vol. 36, no. 5, pp. 547–553, 1987.

[11] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proceedings of the 38th annual Design Automation Conference*, 2001, pp. 684–689.

[12] ——, *Principles and Practices of Interconnection Networks*. San Francisco: Morgan Kaufmann, 2004.

[13] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks*. San Francisco: Morgan Kaufmann, 2003.

[14] D. N. Jayasimha, B. Zafar, and Y. Hoskote, "On-chip interconnection networks: Why they are different and how to compare them," Intel, Tech. Rep., 2006.

[15] N. K. Kavaldjiev, "A run-time reconfigurable network-on-chip for streaming DSP applications," Ph.D. dissertation, University of Twente, 2007.

[16] P. Krueger, T. H. Lai, and V. A. Dixit-Radiya, "Job scheduling is more important than processor allocation for hypercube computers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 5, pp. 488–497, 1994.

[17] A. Kumar, P. K. L. S. Peh, and N. K. Jha, "Express virtual channels: Towards the ideal interconnection fabric," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 150–161, 2007.

[18] P. Mohapatra, C. Yu, C. R. Das, and J. Kim, "A lazy scheduling for improving hypercube performance," in *Proceedings of the 1993 International Conference on Parallel Processing (ICPP '93)*, vol. 1, 1993, pp. 110–117.

[19] C. A. F. D. Rose, H. U. Heiss, and B. Linnert, "Distributed dynamic processor allocation for multicomputers," *Parallel Computing*, vol. 33, no. 3, pp. 145–158, 2007.

TABLE II
ENERGY CONSUMPTION BY NOC WITH CONSIDERED ROUTING ALGORITHMS, BASED ON FLOW CONTROL

	Total VC Count $T_{R_{VC}}$			**Total EVC Count $T_{R_{EVC}}$**			**NoC Energy [μJ]**		
EVC's Length	***BMAT***	***IFF-Mesh***	***IFF-Torus***	***BMAT***	***IFF-Mesh***	***IFF-Torus***	***BMAT***	***IFF-Mesh***	***IFF-Torus***
	DOR and DOR-LB								
0	76288	126622	76802	0	0	0	4.223335	6.280482	4.25179
1	58434	75732	58870	17854	50890	17932	3.794839	5.059122	3.821422
2	55752	66034	56290	20536	60588	20512	3.730471	4.82637	3.759502
3	61012	66016	61622	15276	60606	15180	3.856711	4.825938	3.88747
	Valiant								
0	136094	204088	136994	0	0	0	7.53419	10.122796	7.584019
1	101950	126466	102165	34344	79230	34123	6.72101	8.301033	6.725983
2	102924	117066	102858	33488	88212	33856	6.748088	8.064732	6.755974
3	114470	122448	114409	22158	82404	2142	7.031965	8.182994	6.400887
	Valiant-LB								
0	134642	200826	138684	0	0	0	7.453812	9.961001	7.677578
1	102251	126408	102017	33137	79300	33709	6.699823	8.299948	6.704807
2	103250	116626	101644	33432	89378	33176	6.764379	8.072758	6.667442
3	115564	123352	115014	21876	84048	20982	7.083686	8.269919	7.025202
	Adaptive-M1								
0	97490	151290	97774	0	0	0	5.397078	7.504015	5.4128
1	80400	100429	80500	18572	51177	18622	5.033393	6.291441	5.040497
2	75928	87972	76222	21994	61660	22124	4.893137	5.941938	4.91349
3	82870	87057	83369	14892	62199	14733	5.054727	5.910353	5.077366
	Adaptive-M3								
0	135936	194710	136910	0	0	0	7.525448	9.657647	7.579369
1	111871	131938	111927	23177	59162	23187	6.920041	8.058703	6.923454
2	107332	114890	107792	26018	69122	26016	6.757855	7.468098	6.783258
3	116254	115937	116863	18066	67101	17883	7.002402	7.468292	7.030378

[20] E. Salminen, A. Kulmala, and T. D. Hamalainen, "Survey of network-on-chip proposals," in *White Paper, OCP-IP*, 2008, pp. 1–13.

[21] C. C. Su and K. G. Shin, "Adaptive deadlock-free routing in multicomputers using only one extra virtual channel," in *Proceedings of the 1993 International Conference on Parallel Processing*, vol. 1, 1993, pp. 227–231.

[22] M. B. Taylor and et al., "The raw microprocessor: A computational fabric for software circuits and general-purpose programs," *IEEE Micro*, vol. 22, no. 2, pp. 25–35, 2002.

[23] J. Upadhyay, V. Varavithya, and P. Mohapatra, "A traffic-balanced adaptive wormhole routing scheme for two-dimensional meshes," *IEEE Transactions on Computers*, vol. 46, no. 2, pp. 190–197, 1997.

[24] L. G. Valiant and G. J. Brebner, "Universal schemes for parallel communication," in *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, 1981, pp. 263–277.

[25] D. Wiklund, "Development and performance evaluation of networks on chip," Ph.D. dissertation, Linkoping University, 2005.

[26] B. S. Yoo and C. R. Das, "A fast and efficient processor allocation scheme for mesh-connected multicomputers," *IEEE Transaction on Computers*, vol. 51, no. 1, pp. 46–60, 2002.

[27] Y. Zhu, "Efficient processor allocation strategies for mesh-connected parallel computers," *Journal of Parallel and Distributed Computing*, vol. 16, no. 4, pp. 328–337, 1992.

[28] D. Zydek and H. Selvaraj, "Fast and efficient processor allocation algorithm for torus-based chip multiprocessors," *Journal of Computers & Electrical Engineering*, 2009, submitted for publication.

[29] ——, "Processor allocation problem for NoC-based chip multiprocessors," in *Proceedings of 6th International Conference on Information Technology: New Generations (ITNG 2009)*, 2009, pp. 96–101.

[30] ——, "Hardware implementation of processor allocation schemes for mesh-based chip multiprocessors," *Journal of Microprocessors and Microsystems*, vol. 34, no. 1, pp. 39–48, 2010.

[31] D. Zydek, H. Selvaraj, G. Borowik, and T. Luba, "Energy characteristic of processor allocator and network-on-chip," *Journal of Applied Mathematics and Computer Science*, 2010, submitted for publication.

[32] D. Zydek, H. Selvaraj, and L. Gewali, "Synthesis of processor allocator for torus-based chip multiprocessors," in *Proceedings of 7th International Conference on Information Technology: New Generations (ITNG 2010)*, 2010, pp. 13–18.

[33] D. Zydek, N. Shlayan, E. Regentova, and H. Selvaraj, "Review of packet switching technologies for future NoC," in *Proceedings 19th International Conference on Systems Engineering (ICSEng 2008)*, 2008, pp. 306–311.