K. Latif. A.M. Rahmani, T. Seceleanu, H. Tenhunen. Power- and Performance-Aware IP Mapping for NoC-Based MPSoC Platforms. The 17th IEEE International Conference on Electronics, Circuits, and Systems, ICECS 2010, December 2010, pp. 760-763

Power- and Performance-Aware IP Mapping for NoC-Based MPSoC Platforms

Khalid Latif[1,3], Amir-Mohammad Rahmani[1,3], Tiberiu Seceleanu[2], Hannu Tenhunen[1,3]
[1]Department of Information Technology, University of Turku, Finland.
[2]ABB Corporate Research, Västerås, Sweden. [3]Turku Centre for Computer Science (TUCS), Finland.
{khalid.latif, amir.rahmani, hannu.tenhunen}@utu.fi and tiberiu.seceleanu@se.abb.com

***Abstract*—In this paper, we address the performance of MPSoC platforms with homogeneous processing nodes, where the cores generate and consume the large amount of data, thus the system approaches congestion. Mostly, the time dependent media applications are time critical, where traffic must be delivered on time in order to operate properly. Proper task allocation or placement of IP cores at layout time is very important to meet such application requirements. Apart from meeting the application requirements, it also lowers the traffic congestion, power consumption and Average Packet Latency (APL). For task allocation or IP placement, the prioritization criteria has been proposed, which is used in next step to map the application on MPSoC platform. The proposed technique shows significant improvement in system performance and reduction in power consumption. To estimate the efficiency, the video conference encoding application and MPEG4 video encoder were mapped to 5x5 and 4x4 NoC mesh. Up to 11% reduction in power consumption and 20% reduction in APL has been observed as compared to other proposed mapping techniques.**

I. Introduction

Ever increasing requirements on electronic systems are one of the key factors for evolution of the integrated circuit technology. Continuous technology scaling has made it possible to integrate billions of transistors on a single chip [1]. Thus the entire system with hundreds of components can be integrated on a single chip, which is known as Multiprocessor System-on-Chip (MPSoC). At such integration levels, communication plays the major role in the design and performance. One outcome of higher integration levels is that interconnection platforms are replacing the shred buses. Networks-on-Chip (NoCs) are proposed to be used in complex Sytem-on-Chip (SoC) design for inter-core communication because of scalability, better throughput and reduced power consumption [2].

Due to application requirements, the operation and communication characteristics of the employed devices and architectural instances may vary greatly from system to system. In this situation, we need to design the application and platform specific performance optimization techniques. Unfortunately, it is not possible to define general performance parameters for any MPSoC platform. But most of the modern MPSoC platforms use packet based communication. In case of packet based communication, the standard parameters for optimization are average packet latency (APL) and power consumption.

APL can be optimized either by using an efficient routing technique or by placing the IP cores close to each other, which communicate more often according to the application requirements. If proper placement technique is used, then some simple and power efficient routing technique can be used as well. Major fraction of the total execution time for an application mapped on a MPSoC platform is taken by communication time. Thus, to speed up the execution, we need to optimize the APL value. Different routing algorithms have already been suggested to optimize communication latency. If proper placement of IPs is done and suitable routing technique is used, execution time and power consumption can be further reduced.

NoC platform is composed of routers, physical links between routers and network interface to interact with memory or computation elements. NoC can be expected to handle a much larger bandwidth but here multi-hop topology will affect the lower bound on latency [3]. Thus, average packet latency can be reduced by reducing number of hops. This is only possible for realistic (application) traffic.

The potential applications of multimedia systems span almost all domains for which computers have already proven useful, and seems likely to extend to many new domains as well [4]. At the same time, the demand of high performance multimedia applications is increasing in our daily life. Both performant processing units and high performant MPSoC communication platform are needed to address distributed multimedia applications.

The objective of this paper is to present an automatic and algorithmic mapping technique for MPSoC platforms and considering 2D-mesh NoC as case study. The heuristic approaches are efficient but at application startup, it takes longer to start and optimal mapping may not be achieved. The optimal mapping solution makes the application performance and power efficient. We propose the mapping algorithm, which satisfies the bandwidth constraints of NoC and minimizes the APL and power consumption. The IP mapping is done at high abstraction level based on communication requirements of cores extracted from a high level application simulation tools like Matlab Simulink [5]. A cycle accurate simulation of resulting NoC mapped application in VHDL validates our approach. Reduction in power consumption has been validated by the high level NoC power simulator presented in [6].

The remainder of this paper is organized as follows. Section II explains the previously proposed mapping techniques with the pros and cons of each approach. Section III presents the proposed design flow and mathematical analysis and algorithm implementation. Section IV explains our simulation results. Section V draws final conclusions.

II. Related work

The number of task allocation and scheduling techniques for communication and resource management of MPSoC systems with homogenous and heterogenous cores have already been proposed. An application specific algorithm for task allocation and IP core placement is not a recent topic to be addressed but still needs attention to deal with upcoming modern applications.

Srinivasan et al. [7] and Hu et al. [8] presents a heuristic approach for bandwidth constrained mapping of cores onto NoC architecture. In their approach, initially, the core with maximum communication demand is placed onto one of the mesh nodes with maximum number of neighbors.Then for each core, yet to be mapped, the core communicating more with the already mapped cores is selected and placed onto the node, that minimizes the communication cost with mapped cores. After mapping all the cores, iterative pair-wise

swapping is used to improve the mapping till the system delivers the best performance. The issue with these approaches is that they don't consider the traffic distribution of each core. A core may have high communication requirements with many number of nodes but big fraction of the traffic might be only for one core. In this situation, this core should not be the first one to get mapped.

Stensgaard et al. [9] propose an application specific reconfigurable NoC architecture. In this approach, authors ignore the physical distance between IP-block/router as long as a logical link can be established. Any application can be mapped on NoC in this way. However, if physical distance is optimized as well, this approach may become more power and performance efficient. Physical distance can be optimized by considering communication pattern of known applications to be mapped on system under consideration.

Walter et al. [10] propose to divide the cores into number of classes according to the the core functionality. Initially the random mapping is generated and then repeatedly, cores are swapped to reduce the application execution time unless further improvement is not obtained for a predefined number of iterations.The problem with this appraoch is that because of random nature, the optimal solution cannot be achieved all the time.

Seceleanu et al. [11] introduce application specific approach for device allocation on the Segbus platform based on communication scheduling. Main goal of this approach is to minimize the cross border communication and maximize parallelism in communication between IPs. So, this technique is platform specific as well. Similar approach can be adopted for other MPSoC platforms to minimize the communication load.

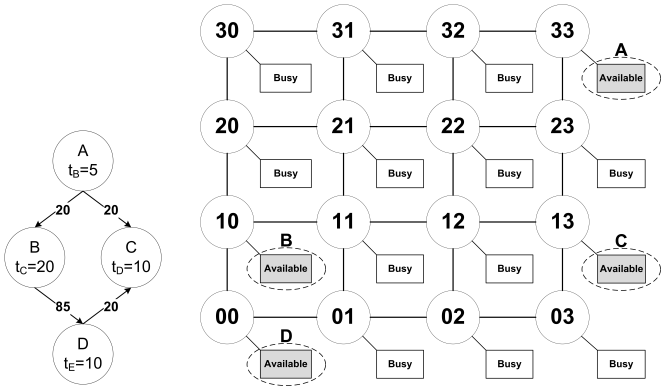
Ogras et al. [12] address the optimization of generic NoC topology with *application-specific long-range* links insertion. The authors propose an algorithm that determines the long-range links that need to be inserted on a regular mesh network. But in this approach, application specific placement of IPs is ignored. If the interconnect topology is decided by considering the application requirements and communication load, we can reduce the number of Long-Range links.

The approach we illustrate here does not impose restrictions towards other MPSOC platforms. In our approach, we propose an algorithm for proper placement of IPs according to the application requirements. The requirement of long range link insertion can be checked that either long-links are still needed or not for desired performance level after the mapping. As example , two multimedia application, video conference encoder and MPEG-4 video encoder are mapped on 2D-mesh NoCs platform.

III. Mapping Technique

In this section, we present the mapping approach. The basic parameters to rate the the mapping technique are communication cost optimization and reduction in power consumption. To optimize the communication cost, the cores should be mapped in such a way that the mapped application looks similar to the point-to-point connections in application and average hop count value approaches '1'. Due to limitations of interconnection platform like a node in 2D-mesh NoC cannot communicate directly with more than four nodes, average hop count value goes away from '1', if a core communicates with more than four cores.

The basic approach for any mapping technique is to map the cores on neighboring nodes, which communicate with each other. Due to unavailability of cores at mapping time, it is not an easy task. Consider the application graph shown in Fig. 1(a), which needs to be mapped on 4x4 2D-mesh NoC of homogeneous cores. Now suppose that other applications are already running on the system and only four cores are available to serve the new application. The available cores are marked in Fig. 1(b). It is vary simple to map a four core application as compared to a many core application. But the available cores are far from each other and only two available cores reside side-by-side. In example graph, cores 'B' and 'D' require more bandwidth to communicate with each other as shown by the weighted edges in Fig. 1(a). Thus cores 'B' and 'D' will be mapped first as neighboring nodes at '10' and '00' respectively. Cores 'A' and 'C' are mapped after the mapping of 'B' and 'D' in other two available cores with minimum communication cost. IPs can be prioritized to map first on the basis of communication load and number of neighboring nodes, which communicate directly with that node like node 'B' communicate directly with node 'D'.



(a) Example application graph

(b) Mapping on 2D-mesh NoC

Fig. 1. Example application mapping with limited availability of cores.

In case of 2D NoC mesh, nodes with higher priority should be mapped on central nodes so that they can communicate with more number of nodes with minimum hop count. To generate the application mapping for any MpSoC platform, basic parameters are the total number of packets communicated by IP, nature of packets like multicast or unicast and order of processing of packets. This information can be extracted from high level simulation tool like Matalb Simulink as discussed in section I. The communication between processes is organized as a Packet based Synchronous Data Flow (PSDF) diagram [13]. For video conference encoder application, PSDF diagram is shown in Fig.2. There are two values on each graph edge. The first value represents the number of successive same size transactions during one application cycle and second value represents the relative ordering among the data flows in given application. The application is a group of sub-applications H.264 video encoder, MP3 audio encoder and OFDM transmitter.

The mapping technique can be divided into two steps: Prioritization and Placement. Prioritization of IP cores is platform independent while the placement phase is platform and topology dependent.

A. Prioritization of IP Cores

After having the inter-core communication data for the given application, next step is to prioritize the IP cores for placement. Prioritization of IP cores is based on four parameters: Total number of packets to be communicated to-or-by the core(N_{Pi}), Number of neighboring cores to be communicated (N_i), Traffic distribution (σ^2_{xi}) and Number of cores communicating with the neighbors of the current core (N_f). Total number of packets, transmitted and received by current IP are computed by $N_{Pi} = \sum_j (C_{i,j}+C_{j,i})$. Similarly, N_i

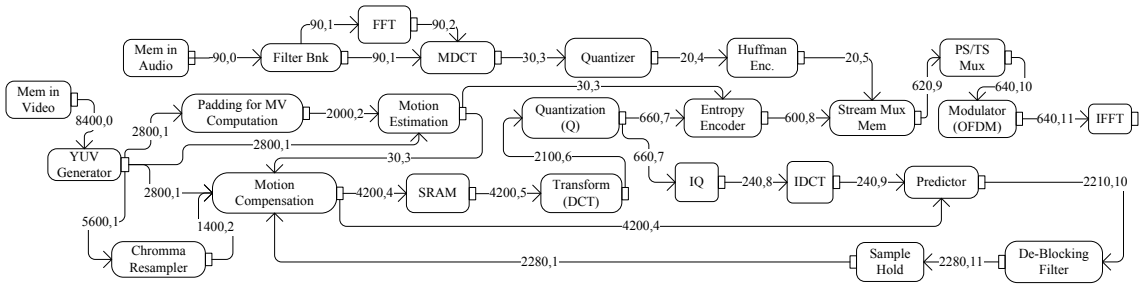


Fig. 2. PSDF for Video Conference encoder application.

= $\sum_j (nnz(C_{i,j}) + nnz(C_{j,i}))$. Where, i is the current IP and $C_{i,j}$ is the number of packets communicated from node i towards node j. *nnz* is a function to count the number of non-zero load communication edges. To count the number of neighbors for a core, a communication edge is considered between each pair cores. The edges with non-zero values represent the edge between neighbors.

Traffic distribution is the load variance for a core. Suppose, an IP communicates with five different cores. But 95% of its traffic is for a single IP core. With rest of cores, it communicates for very few packets. Thus, only one core is strongly connected and should be mapped on neighboring node of current node. The remaining four cores can be mapped on nodes far way from current node, if enough nodes side-by-side are not available. Thus, current node does not require a central node for mapping even if it communicates with many cores. To deal with such situations, traffic distribution is the measure if the communication for current IP is dedicated to fewer IPs or it is uniform for N_i IPs. Statistical variance in communication can be used to decide either the communication load is distributed for an IP core among it neighbors or not. Statistical variance can be computed by Eq.1.

$$\sigma_{xi}^2 = \frac{1}{N_i} \sum_{n=1}^{N_i} (x_{ni} - \overline{X}_{Ni})^2 \tag{1}$$

Where, $\overline{X}_{Ni}$ = Average link load for IP_i.
= N_{Pi} / N_i
x_{ni} = Number of packets to be communicated between *nth* and IP_i.

After extracting the communication parameters, discussed above, next step is to define the prioritization expression. Priority is directly proportional to N_P and N_i and is inversely proportional to the σ_x^2. Now, we have an expression in Eq.2.

$$P_i = \frac{N_{Pi} * N_i}{\sigma_{xi}^2} \tag{2}$$

Where P_i is the priority of *ith* processing element. The factor N_f mentioned above is not considered, while formulating Eq.2. N_f is the number of shared IPs to communicate with *ith* and current IP, where *ith* element has already been placed. Thus, the factor N_f is computed by using the output of Eq.2. Final priority value is directly proportional to N_f given by $P_i * N_f$. The priority sequence for the application shown in Fig. 2 is Motion compensation, YUV Generator, Motion estimation, SRAM etc. respectively in descending order.

B. Placement (Platform Dependent)

After having the prioritized and sorted sequence of IP cores to be mapped, next step is to map the application cores onto the MPSoC platform. The placement algorithm is platform and topology dependent. The task of placement algorithm is to optimize the APL value and reduce the power consumption. With higher value of average hop count, both the dynamic and static power consumption values are increased. So, optimization of average hop count also optimizes the power/performance characteristics. Mapping algorithm for 2D-mesh NoC is shown in Algorithm III.1.

ALGORITHM III.1:

```
P_i = Prioritization (N_p, N_i, σ_xi^2, N_f);
Prioritized_Cores = Sort (P_i);
MPSoC_Platform; // 2D mesh NoC.
Cores_ Availability; // Free cores on MPSoC platform to provide the service.
Placement (Prioritized_Cores, PSDF, MPSoC_Platform, Cores_ Availability)
    MPSoC_Platform() ← Prioritized_Cores(1);
    for (j = 2 to toal_number_of_cores)
        for ( k = 1 to j-1 )
            if ( neighbors (Prioritized_Cores(j), Prioritized_Cores(k), PSDF )
                then Neighbor_Cores[]←Prioritized_Cores(k);
            if ( shared (Prioritized_Cores(j), Prioritized_Cores(k), PSDF )
                then Shared_Cores[]←Prioritized_Cores(k);
        if ( Neighbor_Cores or Shared_Cores )
                then place(Prioritized_Cores(j), Neighbor_Cores, Shared_Cores,
                        MPSoC_Platform, Cores_ Availability);
        else map_away (Prioritized_Cores(j), MPSoC_Platform, Cores_ Availability);
    return mapping information of nodes on MPSoC_Platform;
```

In this algorithm, *Neighbors* is a function to check the point-to-point communication between two cores in actual application. *Shared* function is used to investigate and find that if there are some IP cores, which are mutual neighbors of the parameter IP cores. Using *Shared_Cores*, *Neighbor_Cores*, *MPSoC_Platform* and *Cores_Availability*, current IP is placed by using function *Place*. *Cores_Availability* keeps record of already placed IPs as well. If the next IP core to be mapped does not communicate with already placed IPs, it is placed away from already placed IP cores using the function *Map_Away*.Placement of IPs for video conference encoder application presented in Fig.2 is shown in Fig.3. Insertion of Long-Range links proposed by [12] can be considered with mapping algorithm to further enhance the system performance.

IV. Experimental results

To demonstrate the better power and performance characteristics of the proposed mapping algorithm, a cycle-accurate NoC simulation

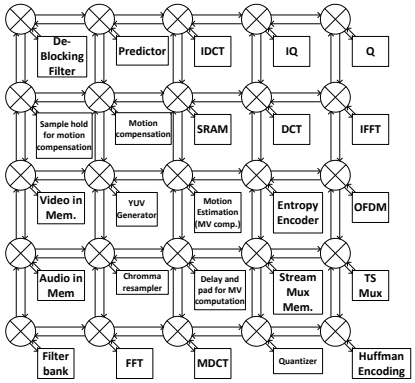


Fig. 3. Mapping generated for the application shown in Fig. 2.

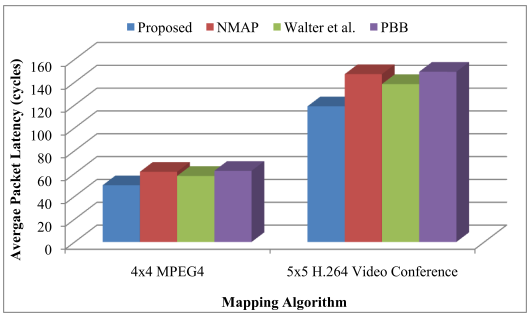


Fig. 4. Average Packet Latency with XY routing algorithm.

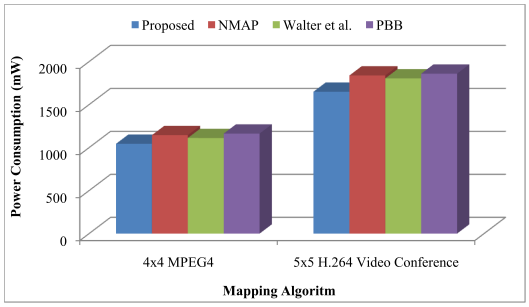


Fig. 5. Power consumption with XY routing algorithm.

environment was implemented running two applications mapped with four different mapping algorithms (Proposed, NMAP [7], Walter et al. [10], and PBB [8]) in HDL. The simulations were performed for 4x4 and 5x5 mesh NoCs with MPEG4 and video conference encoder applications, respectively. The comparison is performed in terms of power consumption and APL. The packet latency was defined as the time duration from when the first flit is created at the source node to when the last flit is delivered to the destination node. For each simulation, the packet latencies were averaged over 50,000 packets. Latencies were not collected for the first 5,000 cycles to allow the network to stabilize. It was assumed that the packets had a fixed length of 66 flits, the buffer size of each synchronous buffer was eight flits and the data width was set to 32 bits. The NoC switches exploit 2 virtual channels for each input port. To perform the simulations, we used an XY wormhole routing algorithm. A 50 MHz clock frequency is applied to the NoC, resulting in a maximum transmission rate per link equals to 400 Mbps. To estimate the switch power consumptions, the high level NoC power simulator presented in [6] was used.

The NoC system performance and power consumption for four different mapping algorithms are given in Fig. 4. It can be observed from the Fig. 4 that the proposed approach shows more than 20% reduction in APL as compared to the PBB mapping technique. Similar are the values for other techniques. The difference in performance is more as the number of cores in application are increased and can be observed in Figures 5 and 4. The mapping technique shows more improvement for 5x5 NoC as compared to the application mapped to 4x4 mesh NoC. The power consumptions of the interconnection network which are based on 35nm standard CMOS technology are presented in Fig.5. Though the pattern of power comparison is similar to APL but reduction factor is different. The reason is that reducing the number of hops directly affects APL. But power consumption is not reduced by same factor because packet generation and static power are the major power sinks, which are independent of the mapping technique.

V. Conclusions and Future Work

Heuristic application mapping technique for NoC shows significant improvement in system performance. Platform independent core prioritization criteria was decided. On the basis of that, power and performance aware mapping algorithm for NoC based MPSoC platform was proposed. A significant improvement in system performance and reduction in power consumption was observed.

The mapping algorithm for 3D-mesh NoC can be developed in next phase. During the mapping, off-chip communication will be considered because the transmitter or receiver nodes for off-chip communications should be mapped on edge or corner nodes. This can significantly reduce the the APL value and power consumption significantly. Multicasting of data is another issue, which needs to be addressed while designing the mapping algorithm.

Acknowledgements

This work is supported by the Nokia Foundation and DOMES project (123518/2008) funded by the Academy of Finland.

References

[1] Shekhar Borkar. *Designing reliable systems from unreliable components: The challenges of transistor variability and degradation* IEEE Micro,25(6):1016, 2005.
[2] A. Jantsch and H. Tenhunen (Eds.) Networks on Chip Kluwer Academic Publishers, 2003.
[3] Rikard Thid, Ingo Sander, Axel Jantsch. *Flexible Bus and NoC Performance Analysis with Configurable Synthetic Workloads.* Proceedings of the 9th EUROMICRO Conference on Digital System Design 2006, pp.681-688.
[4] Alok Choudhary, Ian Foster, Rick Stevens. *Multimedia Applications and High-Performance Computing.* IEEE Parallel and Distributed Technology, vol. 3, no. 2, June 1995, pp. 2-3.
[5] Matlab/simulink, http://www.mathworks.com.
[6] G. Guindani et al. *NoC Power Estimation at the RTL Abstraction Level* In Proc. of ISVLSI 2008, pp. 475-478.
[7] Srinivasan Murali, Giovanni De Micheli *Bandwidth-Constrained Mapping of Cores onto NoC Architectures.* DATE 2004, pp. 896-903.
[8] J.Hu, R.Marculescu *Energy-Aware Mapping for Tile-based NOC Architectures Under Performance Constraints* ASP-DAC 2003.
[9] Mikkel B. Stensgaard, Jens Sparsø. *ReNoC: A Network-on-Chip Architecture with Reconfigurable Topology.* Proceedings of the Second ACM/IEEE International Symposium on Networks-on-Chip (nocs 2008), pp. 55-64.
[10] I. Walter, I. Cidon, and A. Kolodny, D. Sigalov. *The Era of Many-Modules SoC: Revisiting the NoC Mapping Problem* Second International Workshop on Network on Chip Architectures (NoCArc), 2010. pp. 43-48.
[11] Tiberiu Seceleanu, Ville Leppänen, Olli S. Nevalainen. *Device allocation on the SegBus platform based on communication scheduling cost minimization.* Proceedings of the IEEE International SOC Conference (SOCC), Sept. 2007, pp. 191-196.
[12] Umit Y. Ogras, Radu Marculescu. *"It's a small world after all": NoC performance optimization via long-range link insertion.* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, July 2006, pp. 693-706.
[13] D. Truscan et. all. *A Model-Based Design Process for the SegBus Distributed Architecture.* 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), 2008. pp. 307 - 316.