Power and Performance Optimal NoC Design for CPU-GPU Architecture Using Formal Models

Lulwah Alhubail and Nader Bagherzadeh
Department of Electrical Engineering and Computer Science
University of California
Irvine, California 92697-2625
Emails: {lalhubai, nader}@uci.edu

***Abstract*—Heterogeneous computing architectures that fuse both CPU and GPU on the same chip are common nowadays. Using homogeneous interconnect for such heterogeneous processors each with different network demands can result in performance degradation. In this paper, we focused on designing a heterogeneous mesh-style network-on-chip (NoC) to connect heterogeneous CPU-GPU processors. We tackled three problems at once; mapping Processing Elements (PEs) to the routers of the mesh, assigning the number of virtual channels (VC), and assigning the buffer size (BS) for each port of each router in the NoC. By relying on formal models, we developed a method based on Strength Pareto Evolutionary Algorithm2 (SPEA2) to obtain the Pareto optimal set that optimizes communication performance and power consumption of the NoC. By validating our method on a full-system simulator, results show that the NoC performance can be improved by 17% while minimizing the power consumption by at least 2.3x and maintaining the overall system performance.**

***Index Terms*—Heterogeneous architecture, Network-on-Chip, Optimization, Performance, Power, Evolutionary algorithm**

I. Introduction

Heterogeneous CPU-GPU architectures have been widely used because they benefit from the unique architectural features of each processor. While CPUs are the best match for latency-sensitive and irregular applications, GPUs are most suited for throughput-critical and regular applications. Several products from AMD [1], Intel [2], and NVIDIA [3] have emerged that fused the CPU and GPU on the same chip. However, the difference in their architectural imposes different challenges in totally exploiting their potentials.

In this paper, we focus on designing an efficient 2D mesh interconnection that takes the difference in architecture of the CPU and GPU into consideration. In a 2D mesh, routers are connected to each other via communication links and each one is connected to a PE that can be a CPU core, GPU core (SM), shared L2, or an MC. When running applications simultaneously on CPU and GPU cores, we can expect interference between them [4]. This motivates us to design a heterogeneous NoC to connect the heterogeneous CPU-GPU cores, that satisfies two objectives: communication performance and power consumption savings.

Designing a heterogeneous mesh can be challenging because it involves different problems. First, the placement of PEs within the mesh, which can have a significant impact on the performance and power of the system. Next, choosing the number of VCs and BS for each link in NoC. Increasing the number of VCs enhances performance but consumes more power especially when the buffers consume about 35% of the router power [5].

While there are many researches that tackle the design of heterogeneous NoC, most of them considered one problem at a time with a subset of network parameters. Also, their work was based on empirical studies of a pre-defined set of NoC configurations which limit the exploration of the NoC design space. Moreover, many of the existing work targeted NoC performance only.

In our design, we tackle these problems all at once. This makes the design space complex and extremely large. Hence, we use a heuristic method to solve this multi-objective design problem. To get a measure of the performance, represented as the average packet latency improvement, and power of each design, we rely on formal analytical models.

The rest of this paper is organized as follows: Section II summarizes the previous work in the design of heterogeneous NoC. The models are described in Section III. In Section IV, we introduce our optimization method. Section V presents simulation results and comparison with other NoC design approaches. Finally, Section VI provides concluding remarks.

II. Related Work

There are not so many works in the literature that studied heterogeneous NoC design for fused CPU-GPU architecture. The behavior of ring NoC when running CPU and GPU simultaneously under different design choices was surveyed in [4], and an optimal ring NoC was proposed accordingly. However, the effect of BS was not considered and ring NoC is not scalable. In [6], the placement of buffered and bufferless routers for given 2D mesh with assigned PEs was compared regarding speedup and energy of NoC. Only one aspect of router heterogeneity, that is the buffer size, was considered and even for buffered routers a fixed buffer size was used. [7] adopted routers with different buffer size per port. They implemented a queuing-theory based heuristic algorithm to allocate the buffers to router ports to minimize the variation of the average waiting time of each port. However, they only considered 4 different placements of PEs.

978-3-9819263-2-3/DATE19/©2019 EDAA

III. Performance and Power Models

In our design, we are adopting an output-buffered router with three pipeline stages: routing and arbitration, switching and crossbar traversal, and link traversal. For the performance model, we developed the model of [8]. Each output channel is modeled as a server in G/G/1 priority queueing model with arbitrary buffers. Moreover, we added the supports for arbitrary virtual channels. The average latency of a packet in the network can be given by:

$$L = \sum_{\forall S,D} P^{S\to D}\, L^{S\to D} \tag{1}$$

where $P^{S\to D}$ is the probability that source S sends a packet to destination D and $L^{S\to D}$ is the packet latency between S and D nodes.

The total power consumption of the NoC includes the power consumed in the routers and the links. Following the analysis in [9], the power consumed in a router N consists of the power consumed in the routing and arbitration unit, the power consumed in the crossbars, and the total power consumed in the router's links as:

$$P_N^{Router} = P_N^{R\&A} + P_N^{XB} + \sum_{j=1}^{p} P_{N,j}^{TotalLink} \tag{2}$$

The power consumed in the arbitration and routing unit is:

$$P_N^{R\&A} = P_{Header}^{R\&A} \sum_{j=1}^{p} \lambda_j^N \tag{3}$$

where λ_j^N is the arrival rate and $P_{Header}^{R\&A}$ is the power consumed in arbitrating and routing a header flit. The crossbar power consumption is:

$$P_N^{XB} = P_{bit}^{XB}\, K\, m \sum_{j=1}^{p} (\lambda_j^N)^2 \tag{4}$$

where P_{bit}^{XB} is the dynamic power consumed when a bit traverses the crossbar, K is the size of flit in bits, and m is the average size of the packets in flits.

The total power consumed by the router's link consists of the dynamic power consumed by the link, the dynamic, and the leakage power of the buffers:

$$P_{N,j}^{TotalLink} = P_{N,j}^{Link} + P_{N,j}^{BufferD} + P_{N,j}^{BufferL} \tag{5}$$

To find the dynamic link power of an L-millimeter channel j with W bits width connected to a router N that has f_{clk} frequency and V_{DD} supply voltage:

$$P_{N,j}^{Link} = \frac{1}{2}\, \lambda_j^N\, m \left(\alpha_L\, W\, C_L^0 + \alpha_C\, (W-1)\, C_C^0\right) L\, f_{clk}\, V_{DD}^2 \tag{6}$$

where α_L and α_C are the probabilities that different bit values cross over a single and adjacent links, respectively. C_L^0 and C_C^0 are the link and the crosstalk capacities per millimeter.

The dynamic power of the buffers is the power consumed in reading/writing a flit from/to the buffer calculated as:

$$P_{N,j}^{BufferD} = m\, k\, V_j^N\, (\lambda_{j,vc}^N\, P_{bit}^W + \mu_j^N\, P_{bit}^R + Q_j^N\, P_{bit}^{clk}) \tag{7}$$

where V_j^N is the number of virtual channels at port j, P_{bit}^W and P_{bit}^R are the power consumed in writing and reading a bit to/from buffer, respectively. P_{bit}^{clk} is the average power consumed when a one-bit memory element receives a clock switch. Q_j^N is the average number of packets in the output buffer and calculated based on Little's Theorem as:

$$Q_j^N = \lambda_j^N\, W_j^N \tag{8}$$

where W_j^N is the waiting time of output port j of router N.

The leakage power of the output buffer j of router N:

$$P_{N,j}^{BufferL} = B_j^N\, W\, P_{bit}^L \tag{9}$$

where B_j^N is the buffer size in flits, and P_{bit}^L is the average leakage power of one-bit memory element.

Then, the total power consumption of the NoC can be found:

$$P_{NoC} = \sum_{R} P_R^{Router} \tag{10}$$

IV. Optimization Method

Designing a heterogeneous NoC consists of different sub-problems like placing the PEs within the mesh, choosing the number of VCs, and choosing the BS of each link. While each one of these problems has a complex and extensive design space, tackling them all at once enlarges it even more. Also, considering design for two conflicting objectives, performance and power saving, complicates the design problem even further.

To formulate the problem, we have a set of PEs and routers connected in 2D mesh style NoC. Also, a communication rate matrix between the PEs and the injection rate of each PE. We need to map the PEs into the routers of the NoC and configure the number of VCs and the BS for each link of the NoC. All these problems should be solved while minimizing the total delay of the NoC as in (1) and the total power consumption as in (10).

SPEA2 [10] is an evolutionary algorithm that is efficient for finding the Pareto optimal set for multi-objective problems. It works with two fixed-size populations; a regular population of solutions and an archive that keeps the non-dominated solutions. A solution is non-dominated when none of the objective functions can be improved without degrading some of the other objectives. A flow chart of our proposed method is shown in Fig.1.

It starts with a random population of solutions and an empty archive. Then, a fitness is assigned to each solution using the solution's raw fitness and its density. The raw fitness of a solution is calculated as the sum of the strength value of all the solutions that dominate it; where the strength of a solution represents the number of solutions it dominates. The density of a solution is a decreasing function of the distance, in the objectives space, to the k-th nearest neighbor solution; where k is the square root of the sum of the population and the archive sizes. Then, the fitness of the solution is:

$$Fintess(S) = Raw\_Fitness(S) + Density(S) \tag{11}$$

Next, the non-dominated solutions of the combined regular population and archive are copied to a new archive. A truncation operation is applied on the new archive, if it exceeds the

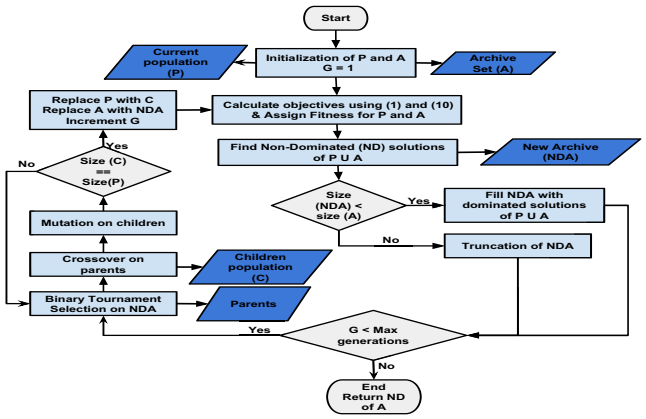

Fig. 1. Flow chart of the proposed multi-objective heterogeneous NoC optimization based on SPEA2

fixed size, iteratively. In each iteration, the solution with the minimum distance to another solution is chosen for removal.

To generate the children population, reproduction operators are applied to the new archive. First, parents are chosen using binary tournament selection where the solution that dominates the other is chosen as a parent. Then, two types of crossover are applied on the parents to generate two children. The placements of the PEs in the children are obtained by applying Partially Mapped Crossover on the parents to ensure a one-to-one mapping between the routers and the PEs. For the port settings, VC and BS, a one-point crossover is applied to the parents, swapping the port settings of the parents after a random point to generate the children. Next, up to three types of mutation are applied to each router of the children randomly. The first type swaps the PE assigned to the router with the PE assigned to another random router. The second type changes the BS by either assigning a random new BS to a random port or scrambling the BS of three random ports of the router. The third type changes the number of VC in a similar fashion. After replacement, the process is repeated for a maximum number of generations. Finally, the algorithm returns the optimal Pareto set with the best PEs mapping and NoC BS and VC configurations.

V. Experimental Results

To simulate a heterogeneous NoC for fused CPU-GPU architecture, we used full-system gem5-gpu simulator [11] after modifying it to support different BS and VC for each port. This simulator provides a choice to use a shared page walk cache (PW) that is accessed upon a miss in the SM's L1 TLBs [12].

For our simulations, we used the configurations shown in Table I. Based on the observations in [7], we adopted for our baseline a simple placement of the PEs that places the shared caches and MCs in the middle of CPU cores and GPU cores, as shown in Fig.2. Note that the use of PW is optional and the design does not depend on it.

We grouped CPU benchmarks from Parsec [13] (Blackscholes, Bodytrack, Canneal, Dedup, Fluidanimate, Freqmine,

TABLE I
GEM5-GPU CONFIGURATIONS

Parameter	CPU	GPU
Number of cores	4	6
Core Clock	2 GHz	1.4 GHz
Private L1 I cache	2-way 32 kB	
Private L1 D cache	2-way 32 kB	4-way 32 kB
Shared L2 cache	8-way 2 MB	
Memory Controller	4 (each 8 banks, 4 channels)	
	3.006 GHz, 1 KB row-buffer	
	FR-FCFS scheduler	
DRAM	DDR3-1600 16GB	
Coherency Protocol	MESI-Two-Level	
Baseline NoC configurations		
VC	4 per port (8-flit buffer)	
Link	16 B width, 1 cycle latency	
Routing	x-y Routing	

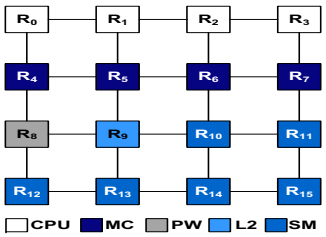

Fig. 2. The PEs placement in the baseline NoC architecture.

Streamcluster, Swaption, X264) and GPU benchmarks from Rodinia [14] (Backprop, Gaussian, HotSpot, LU Decomposition, Nearest Neighbor, Needleman-Wunsch, Path Finder) into 7 TestSets. Each TestSet has 3 independent CPU benchmarks and 1 GPU benchmark. Then, we ran each TestSet on gem5-gpu using the homogeneous baseline configurations till the GPU benchmark finished. Then, we rotated the benchmarks over different CPU cores and reran them. We repeated this process 2 more times and took the average of the 4 generated traffics and injection rates as inputs to our optimizer.

By running our optimizer, we get a non-dominated Pareto optimal set of solutions; each represents a NoC design with optimal mapping of PE and optimal assignment of BS and VC per link. Among the set, we chose to compare the solution with the best performance (SPEA2-Latency), the solution with the best power consumption (SPEA2-Power), and the solution with the best fitness as in (11) (SPEA2-Fitness).

We chose to compare against the homogeneous configuration (the baseline) and the Dual configurations inspired from [5] which utilizes two types of routers, big (7 VC) and small (3 VC). Based on the traffics generated using the baseline configuration, we chose 4 routers with the highest injection rates to be big, and the rest are small. We ran all these different configurations again on gem5-gpu and compared them regarding the performance of NoC, power of NoC, and the performance of the system. The power was obtained by feeding the output of the simulation to DSENT [15] using 22nm technology node after modifying it to support heterogeneous BS and VC per port.

Fig.3 shows the improvements in NoC average packet

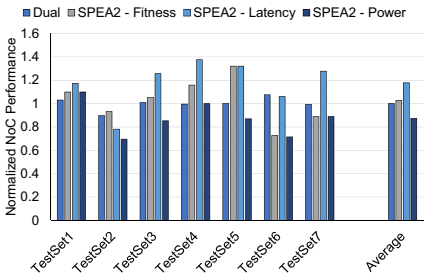

Fig. 3. Improvements in NoC latency using different configurations normalized to the baseline.

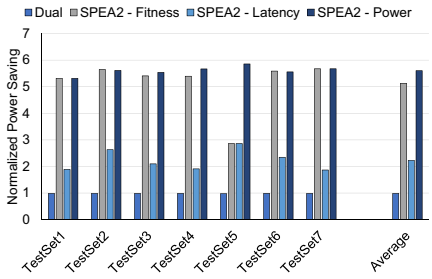

Fig. 4. NoC power consumption savings using different configurations normalized to the baseline.

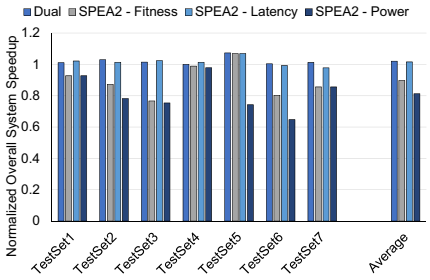

Fig. 5. Overall speedup of the system gained by using different configurations normalized to the baseline.

latency of all the configurations. Regarding the optimal configurations obtained from SPEA2, the configuration with the best latency provides better improvement than other configurations and on average provides 17% improvement. Generally, the configuration with the best fitness improves the performance of the NoC while the configuration with optimal power does not provide any improvement. On the other hand, Dual configuration slightly improves the latency for all except 3 TestSets but has less improvement than the SPEA2 optimal fitness and optimal latency configurations.

For the NoC power savings, as in Fig.4, all the SPEA2 configurations save more power than the baseline. The savings is up to 5.1x, 2.3x and 5.6x on average using fitness, latency, and power optimal SPEA2 configuration, respectively. However, Dual configuration does not provide any power saving. This is mainly due to the considerable reduction in NoC area obtained from our method. It reaches up to 4x, 2.1x, and 4.3x on average using fitness, latency, and power optimal configuration, respectively with no reduction using Dual.

We compared the instruction per cycle obtained using each configuration and used the geometric mean of CPU speedup and GPU speedup as a measure of the overall system performance, as in Fig.5. Generally, Dual configuration slightly improves the overall system speedup with an average of 2%. The optimal latency configuration provides an average improvement of 1.6%. On the other hand, both the optimal fitness and power solutions degrade the overall system speedup.

VI. Conclusion

This paper tackles the multi-objective optimization problem of designing a heterogeneous NoC for a fused CPU-GPU architecture, by solving three sub-problems simultaneously. It presents a method based on SPEA2 to explore the design space of PEs placements and NoC link configurations (BS and VC). It relies on formal analytical models to measure the performance and the total power of the NoC and obtain a Pareto optimal set. This gives the NoC designer a set of design choices to choose from depending on the target architecture goals; power or performance.

Acknowledgment

This work is supported by a scholarship from Kuwait University.

References

[1] W. V. Winkle, "Amd fusion: How it started, where it's going, and what it means," August 2012. [Online]. Available: http://www.tomshardware.com/reviews/fusion-hsa-opencl-history,3262.html

[2] D. Kanter, "Intel's sandy bridge microarchitecture," September 2010. [Online]. Available: https://www.realworldtech.com/sandy-bridge/

[3] B. Dally, "Project denver processor to usher in new era of computing," January 2011. [Online]. Available: https://blogs.nvidia.com/blog/2011/01/05/project-denver-processor-to-usher-in-new-era-of-computing/

[4] J. Lee, S. Li, H. Kim, and S. Yalamanchili, "Design space exploration of on-chip ring interconnection for a cpu–gpu heterogeneous architecture," *Journal of Parallel and Distributed Computing*, vol. 73, no. 12, pp. 1525–1538, 2013.

[5] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "A case for heterogeneous on-chip interconnects for cmps," in *ACM SIGARCH Computer Architecture News*, vol. 39, no. 3. ACM, 2011, pp. 389–400.

[6] J. Fang, Z.-Y. Leng, S.-T. Liu, Z.-C. Yao, and X.-F. Sui, "Exploring heterogeneous noc design space in heterogeneous gpu-cpu architectures," *Journal of Computer Science and Technology*, vol. 30, no. 1, pp. 74–83, 2015.

[7] Z. Li, N. Goswami, and T. Li, "Iconn: A communication infrastructure for heterogeneous computing architectures," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 11, no. 4, p. 42, 2015.

[8] A. E. Kiasari, Z. Lu, and A. Jantsch, "An analytical latency model for networks-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 1, pp. 113–123, 2013.

[9] M. Arjomand and H. Sarbazi-Azad, "Power-performance analysis of networks-on-chip with arbitrary buffer allocation schemes," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 10, pp. 1558–1571, 2010.

[10] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[11] J. Power, J. Hestness, M. S. Orr, M. D. Hill, and D. A. Wood, "gem5-gpu: A heterogeneous cpu-gpu simulator," *IEEE Computer Architecture Letters*, vol. 14, no. 1, pp. 34–36, 2015.

[12] J. Power, M. D. Hill, and D. A. Wood, "Supporting x86-64 address translation for 100s of gpu lanes," in *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on*. IEEE, 2014, pp. 568–578.

[13] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*. ACM, 2008, pp. 72–81.

[14] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, S.-H. Lee, and K. Skadron, "Rodinia: A benchmark suite for heterogeneous computing," in *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*. Ieee, 2009, pp. 44–54.

[15] C. Sun, C.-H. O. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L.-S. Peh, and V. Stojanovic, "Dsent-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling," in *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on*. IEEE, 2012, pp. 201–210.