

A Case for Heterogeneous Network-on-Chip Based H.264 Video Decoders

Milad Ghorbani Moghaddam

Dept. of Electrical and Computer Engr., Marquette Univ.
Milwaukee, Wisconsin
milad.ghorbanimoghaddam@marquette.edu

Cristinel Ababei

Dept. of Electrical and Computer Engr., Marquette Univ.
Milwaukee, Wisconsin
cristinel.ababei@marquette.edu

ABSTRACT

The design of a heterogeneous network-on-chip (NoC) based H.264 video decoder is proposed. A thorough investigation using a system simulator developed as the combination of a cycle accurate NoC simulator together with complete implementations of all the video decoder modules is presented. The target hardware platform is a multicore system-on-chip, where the cores were designed for specific functions that correspond to the modules of the video decoder. Because such cores have different sizes and aspect ratios, a heterogeneous NoC is employed to facilitate the communication between modules. This is different from the reference case of a homogeneous NoC based hardware platform, where all cores are general purpose processors with the same area and where the NoC is a regular mesh NoC. The investigation looks at the impact of core sizes and floorplan for a given technology node as well as at the performance variation across several technology nodes.

CCS CONCEPTS

• **Hardware** → **Network on chip**; • **Computer systems organization** → *Data flow architectures*; *System on a chip*; • **Computing methodologies** → *Image compression*.

KEYWORDS

H.264 video decoder; heterogeneous network-on-chip; homogeneous network-on-chip

ACM Reference Format:

Milad Ghorbani Moghaddam and Cristinel Ababei. 2019. A Case for Heterogeneous Network-on-Chip Based H.264 Video Decoders. In *Great Lakes Symposium on VLSI 2019 (GLSVLSI '19)*, May 9–11, 2019, Tysons Corner, VA, USA. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3299874.3318013>

1 INTRODUCTION

The network-on-chip (NoC) has gained in popularity as a scalable communication mechanism between an increasingly large number of cores in integrated multicore chips [1]. NoCs can be classified into two major types, homogeneous and heterogeneous. In a homogeneous NoC, the routers are identical and arranged in a 2D regular array with the assumption that processing elements (PEs)

are also identical or have the same area and aspect ratio. This naturally leads to identical tiles (tile = NoC router + PE), which in turn enhances the design predictability and simplifies the IC fabrication. We have studied a homogeneous NoC based H.264 video decoder design in our previous work in [2]. In contrast, a heterogeneous NoC is irregular; the routers are not arranged anymore in a regular array. Locations and distances between routers can vary, generally dictated by the size and placement of the heterogeneous cores that make-up the hardware platform. In addition, routers are not restricted to be identical and their number of input/output ports can vary, based on how many different PEs are connected to a given router. All these make the design of heterogeneous NoCs to be more difficult; as a result they have been studied less. In previous work, we introduced a complete design flow for the design and optimization of heterogeneous NoCs and its complete implementation is publicly available [3].

The simulation tool developed in this paper combines the complete video decoder algorithm simulator from [2] with the heterogeneous NoCs synthesis and optimization tool from [3] to investigate the proposed heterogeneous NoC based H.264 video decoder. Thus, the full system simulator developed here is capable of simulating both the heterogeneous NoC and the H.264 modules as a whole system, where the evaluation is done on real video streams, which exercise the NoC with truly realistic traffic.

2 PROPOSED HETEROGENEOUS NOC BASED H.264 VIDEO DECODER

Video coding is a basic algorithm that is used virtually in all video applications including digital TV, mobile TV, and online video streaming. Due to lack of space, we kindly refer the reader to [4] for background information on the H.264 video decoder algorithm.

The main idea of the design presented in this paper is that the modules of the H.264 video decoder are to be implemented and executed as specialized or specific PEs in order to achieve the highest performance. This approach is different from the reference case, where the H.264 modules were assumed to be executed on general purpose CPUs of a regular tiled multicore processor with regular mesh homogeneous NoC based communication, as it was the case in previous work [2]. Because when the modules are implemented as specific PEs their size and aspect ratio can be different, we must use a heterogeneous NoC for communication among modules.

The steps in the design approach proposed in this paper are illustrated in Fig. 1. The first portion of this design flow is adapted from the previous work on synthesis of heterogeneous NoCs [3]. The input to this design flow is the application communication task graph (CTG). The file with the CTG information also includes information about the specific PEs (sizes and aspect ratios) that execute the corresponding H.264 modules. Furthermore, information about

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GLSVLSI '19, May 9–11, 2019, Tysons Corner, VA, USA

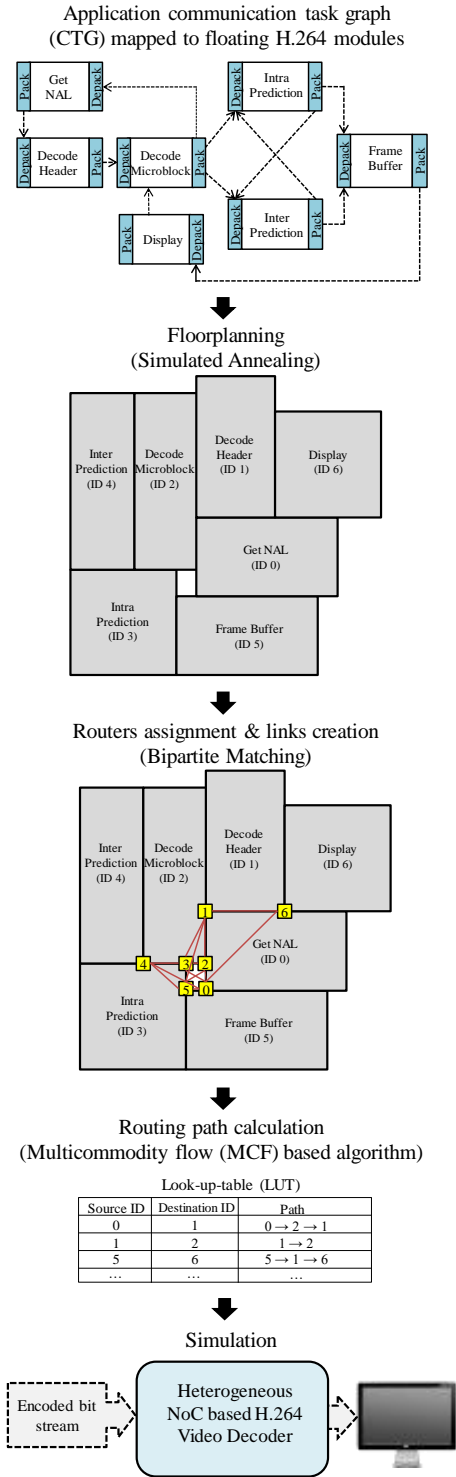


Figure 1: Overview of the proposed design flow to synthesize the heterogeneous NoC for the H.264 video decoder design and to simulate the whole system.

the communication volume between these modules is included as well. This information will be used by the floorplanning algorithm, which will attempt to place PEs that communicate heavily as close as possible to each other. The floorplanning algorithm is based on the popular B*-tree model and it employs a simulated annealing (SA) technique to find the best placement of the PEs. The cost function of the SA algorithm is a weighted combination of the total area and the total wirelength of the design [3].

After the floorplanning is done, in the next step, each PE is assigned to an NoC router. NoC routers are assumed to be located in the corners of the PEs. This router assignment step is implemented through an efficient bipartite matching algorithm, which finds the best matching of PEs to routers such that connectivity between PEs will be implemented via minimal routing paths through the network. Once the routers are assigned, the step of routing paths calculation follows. In this step, the best routing paths between PEs are found using a multicommodity flow (MCF) algorithm. The result of this step is the information that captures the exact routes for all packets between any source and destination PEs. This information is effectively stored in the so-called routing look-up-tables (LUTs) inside each router of the heterogeneous NoC. At the end of this step, we have information on the synthesized heterogeneous NoC, the exact routing paths, in addition to the C++ routines that implement the functions of all the H.264 video decoder modules.

All this information is then used in the second portion of the design flow from Fig. 1. In this portion, we develop a full system simulation framework similar to that in the previous work from [2]. The difference here is that we use a heterogeneous NoC and that we assume the H.264 modules implemented as specific PEs and not as general purpose CPUs. This full system simulation framework is capable of simulating the whole H.264 design in realtime (with video stream files are supplied as input) as a combination of the heterogeneous NoC and of the video decoder function modules. In this way, truly realistic traffic exercises the NoC and the average network latency and power consumption are as accurate as we can get via simulations. The output of the simulation framework includes average network latency and power consumption numbers, reported at the end of the simulation of a given video stream testcase, in addition to rendering in realtime the decoded video on the monitor of the computer used for simulations, as indicated at the bottom of Fig. 1.

3 SIMULATION RESULTS

The complete design flow from Fig. 1 was implemented in C++ as a computer program that automates all steps. The implementation of the first portion of Fig. 1 is adapted from the previous work on heterogeneous NoC synthesis [3] while the second portion that does (heterogeneous NoC + H.264) simulation is a modified version of the simulator from the previous work in [2], which uses C++ complete descriptions of each H.264 module from [5]. We report simulation results for ten different video stream benchmarks. These benchmarks are listed in Table 1, where their resolution is given in numbers of pixels horizontally (i.e., on a row) x number of pixels vertically (i.e., on a column). The files of these benchmarks were downloaded from [6, 7].

The investigation looks at the impact of core sizes and floorplan within a given technology node as well as at the performance

Table 1: Summary of the ten video benchmarks.

Benchmark	Resolution (width x height)	Benchmark	Resolution (width x height)
Girl	352x288	Stefan	352x240
Freeway	352x288	Bigbuckbunny	352x240
Golf	352x288	Coastguard	352x240
Plane	352x288	Flower	352x240
Shore	352x288	Foreman	352x240

variation across several technology nodes. Simulation results are reported as the average network latency, power consumption, and power-delay-product (PDP) for each video benchmark. The average network latency numbers are directly reported by the heterogeneous NoC simulator. To estimate the power consumption, the NoC simulator is integrated with the Orion 2.0 power model for NoCs for a default 65 nm technology node [8], validated with real data from the Intel’s 80 core chip [9]. This is the most accurate NoC power model to date, that considers the switching activities inside each NoC router as resulted from the handling of real data carried as packets through the network.

3.1 Heterogeneous NoC based H.264 Video Decoder vs. Homogeneous NoC based H.264 Video Decoder

In the first set of simulations, we look at how the performance of the heterogeneous NoC based H.264 video decoder changes for different video testcases in comparison with the homogeneous NoC based H.264 video decoder from [2]. For the heterogeneous NoC based design, we assume different levels of optimization of the specific PEs that implement the various decoder modules. These levels of optimization translate into different areas of the PEs, which can result into different floorplans of the heterogeneous chip. Therefore, better PE optimization assumes a smaller area for the PE. Specifically, we considered four different levels of such optimization, which were translated into four different assumed areas for the PEs.

All area scaling experiments used as a reference the larger area of the general purpose CPU that was used in the homogeneous NoC based decoder design from [2]. The heterogeneous NoC + H.264 design used in all simulations used the best floorplan (see Fig. 2) identified during the floorplanning step from the design flow in Fig. 1, where for each PE we assumed available three different aspect ratios, as illustrated in Fig. 3. The simulation results for the default 65 nm technology node are reported in Table 2. We note that the heterogeneous NoC based H.264 video decoder design offers consistently predictable performance.

The results from the comparison between the reference homogeneous NoC based decoder design and the proposed heterogeneous NoC based decoder design are reported in Fig. 4. Aside from the four different levels of assumed heterogeneous PE optimization indicated as 90%, 70%, 50% and 30% smaller PE area, we also include the case when no optimization of the heterogeneous PE is done at all. That is indicated as the 100% in Fig. 4. This case is essentially a design with heterogeneous PEs with the same size as the homogeneous general purpose CPUs but with communication done via a heterogeneous NoC. These results are calculated under the assumption that the global interconnect delay varies linearly with the wire length, which is possible when wires are integrated

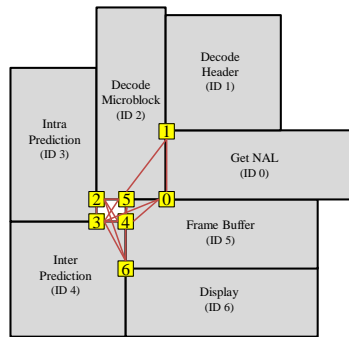


Figure 2: The best floorplan identified and used in all simulations. The heterogeneous NoC is formed by seven routers numbered from 0 to 6.

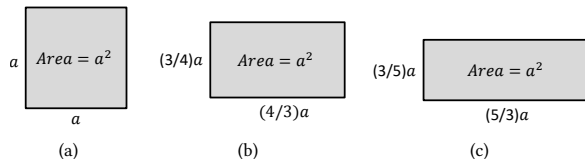


Figure 3: Three different aspect ratios assumed for each PE.

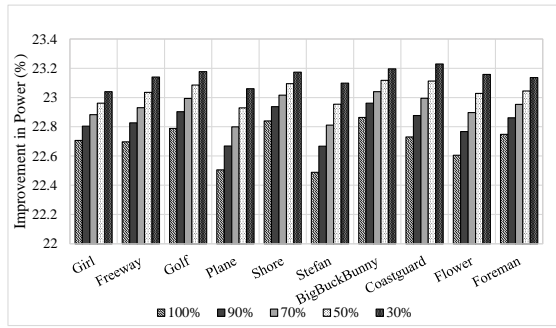
Table 2: Hetero NoC + H.264 modules simulation, 65 nm.

Benchmark	Power consumption (W)	Average latency (cycles)	Power Delay Product, (PDP)
Girl	0.1028	7.5740	0.7786
Freeway	0.1032	7.5751	0.7817
Golf	0.103	7.5800	0.7807
Plane	0.1033	7.5724	0.7822
Shore	0.1027	7.5778	0.7782
Stefan	0.1034	7.5819	0.7839
Bigbuckbunny	0.1029	7.5886	0.7808
Coastguard	0.103	7.5783	0.7805
Flower	0.1034	7.5827	0.7840
Foreman	0.1029	7.5822	0.7802

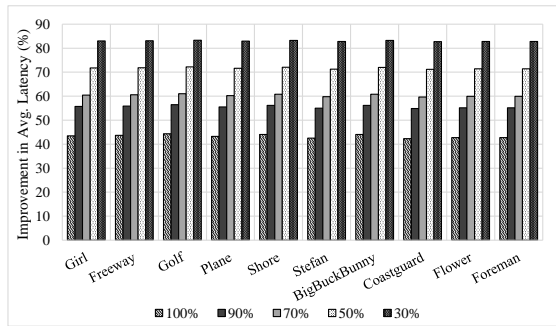
with repeaters according to the ITRS interconnect roadmap [10]. Note that if we assumed a quadratic dependency of wire delay with length, the comparison would be significantly more in favor of the heterogeneous NoC based decoder design. We observe that the heterogeneous NoC based decoder significantly outperforms the homogeneous NoC based decoder, with more than 20% in terms of power consumption and with more than 40% in terms of average network latency.

3.2 Impact of Technology Downscaling

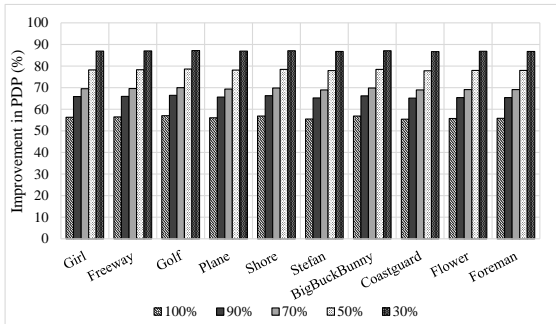
In this section, we used the technology downscaling models from [11, 12] and the information on technology trends reported in the ITRS roadmap from [10] to investigate the variation of the performance of the proposed heterogeneous NoC based H.264 video decoder across different technology nodes. The result of this investigation is reported in Fig. 5 for the *Plane* benchmark only. Similar plots were obtained for all other simulated video benchmarks. This graph shows on the left y-axis the actual power-delay-product (PDP) numbers, with the delay being used as the average number



(a)



(b)



(c)

Figure 4: Improvement of the heterogeneous NoC based decoder over the homogeneous NoC based decoder for heterogeneous PE sizes not-scaled (100%) and scaled to 90%, 70%, 50% and 30% of the reference homogeneous CPU sizes. (a) NoC power improvement, (b) average NoC latency improvement, and (c) NoC PDP improvement.

of cycles, as reported by the NoC simulator. The right y-axis shows the ratio between the PDP values of each of the heterogeneous cases and the PDP value of the reference homogeneous case. This figure provides interesting insights into the effect of technology downscaling from 65 nm down to 7 nm on the PDP performance metric. We note that while the PDP numbers go down with technology downscaling in all cases, the heterogeneous NoC based decoder maintains a performance advantage over the homogeneous NoC based decoder. However, this performance advantage decreases

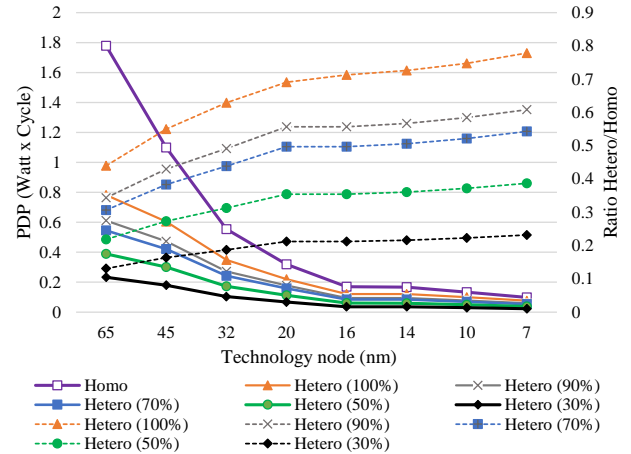


Figure 5: Impact of technology downscaling impact on the power-delay-product for the *Plane* benchmark.

slightly as we move from 65 nm towards 20 nm, after which it remains almost the same as we move towards 7 nm. In addition, the differences between different optimized versions of the heterogeneous NoC based decoder are not as significant at deeper technology nodes.

4 CONCLUSION

The design of a heterogeneous NoC based H.264 video decoder was presented and compared with the reference case of a homogeneous NoC based H.264 video decoder via full system simulations. The impact of heterogeneous core sizes and floorplan within a given technology node as well as the performance difference variation across technology nodes from 65 nm down to 7 nm were investigated. Simulation results indicated that the heterogeneous design offered consistently better performance compared to the reference homogeneous case across all technology nodes, although, at deeper technology nodes the performance gap shrinks slightly.

REFERENCES

- [1] R. Marculescu, U.Y. Ogras, L.-S. Peh, N.E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives," *IEEE TCAD*, vol. 28, no. 1, pp. 3-21, 2009.
- [2] M.G. Moghaddam and C. Ababei "Performance evaluation of network-on-chip based H.264 video decoders via full system simulation," *IEEE Embedded Systems Letters*, vol. 9, no. 2, pp. 49-52, 2017.
- [3] C. Ababei, "Efficient congestion-oriented custom Network-on-Chip topology synthesis," *IEEE Int. Conf. on Reconfigurable Computing and FPGAs (ReConFig)*, 2010.
- [4] ITU-T H.264, Advanced video coding for generic audiovisual services, 2012. [Online]. Available: <http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11466>
- [5] Martin Fiedler, Implementation of a basic H.264/AVC Decoder, 2016. [Online]. Available: <http://keyj.emphy.de/files/projects/h264-src.tar.gz>
- [6] FastVDO: H.264 Video streams, 2018. [Online]. Available: <http://www.fastvdo.com/H.264.html>
- [7] YUV Video Sequences, 2018. [Online]. Available: <http://trace.eas.asu.edu/yuv>
- [8] A.B. Kahng et al., "ORION 2.0: A power-area simulator for interconnection networks," *IEEE TVLSI*, vol. 20, no. 1, pp. 191-196, 2012.
- [9] S.R. Vangal et al., "An 80-tile sub-100-W TeraFLOPS processor in 65-nm CMOS," *IEEE Journal of Solid-state Circuits*, vol. 43, no. 1, pp. 29-41, 2008.
- [10] The International Technology Roadmap for Semiconductors (ITRS), 2013. [Online]. Available: https://www.semiconductors.org/clientuploads/Research_Technology/ITRS/2013/2013Interconnect.pdf
- [11] A. Stillmaker and B. Bevan, "Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm," *Integration, the VLSI Journal*, vol. 58, pp. 74-81, 2017.
- [12] H. Ron, K.W. Mai, and M.A. Horowitz, "The future of wires," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 490-504, 2001.