Time-Triggered Frequency Scaling in Network-on-a-Chip for Safety-Relevant Embedded Systems*

*

1st Rakotojaona Nambinina, Daniel Onwuchekwa, Hamidreza Ahmadian, Dinesh Goyal, Roman Obermaisser
Chair for Embedded Systems
University of Siegen
Siegen, Germany
Andrianoelisoa.Rakotojaona@uni-siegen.de

***Abstract*—Networks on Chip (NoC) are used in Multiprocessor System on a Chip (MPSoC) architectures and safety-relevant systems as a communication backbone to cope with high communication traffic. Nevertheless, network interfaces and the routers within the NoC add to the power consumed by the chip. This paper presents models and algorithms for low power techniques in NoC based on dynamic frequency scaling for safety-relevant real-time systems. The novel technique is time-triggered frequency scaling to enable power and energy efficiency while preserving safety and real-time guarantees. Time-triggered Frequency Scaling (TTFS) is used in the routers where each router frequency is adjusted according to a schedule. Three techniques, namely global, cluster-based, and route-based TTFS are considered. Using the Orion 3 tool, we show that up to 59 % of the power consumed by an NoC can be saved when a route-based power saving technique is applied.**

***Index Terms*—Dynamic frequency scaling, Network-on-a Chip, Time-triggered Multi-Processor, System on a Chip .**

I. Introduction

The rapid development of technology has brought chip design into the field of billion-level transistors, which means that many cores can be integrated into a single chip. With the increase of the core's number in a single chip, the shared bus used in traditional SoC has no longer meets the design requirement. At this time, Network on Chip (NoC) was proposed as a new solution to maintain the performance of the complex SoC [14].

Many cyber-physical devices continue to require improvements to attain desired operational low power. Power optimisation is still a topic in the design of MPSoC when trade-offs are made considering the size of an NoC. Some conventional techniques already exist to enhance both static and dynamic power consumptions in MPSoC. These techniques include clock gating, power gating, Dynamic Voltage and Frequency Scaling (DVFS). These techniques are often combined to improve the power consumption of NoC. Power gating switches off the voltage domains/components that are not in operation to reduce the leakage power. Clock gating disables non-used logic block frequency until the logic block receives an enable signal from the input port. Dynamic frequency scaling and Dynamic Voltage Scaling are popular techniques used in many synchronous circuits to reduce the power consumption of NoC. These techniques scale down the frequency/voltage of specific logic blocks on the fly to reduce the power consumption of devices. Due to the ever-increasing application requirements for cyber-physical systems, the bandwidth requirement for on-chip communication is increasing. As the on-chip resources continue to require more bandwidth, there is the impending need to scale up the size of the NoC to accommodate the communication traffic. Scaling up the size of the NoC conversely results in straining the power-saving solutions mentioned above.

This paper introduces a novel Time-Triggered Frequency Scaling (TTFS) technique to optimize the NoCs router's energy consumption in the MPSoC. The proposed TTFS technique is used to alleviate the energy impact when scaling the NoCs to satisfy higher bandwidth requirements for on-chip resources. A time-triggered schedule controls the TTFS to provide low-power services and real time guarantees to maintain the safety of the systems. A TTFS is implemented in the routers such that each router frequency is triggered according to a time-triggered schedule. We propose and evaluate three techniques in which TTFS can be applied. These techniques include global, cluster-based, and route-based TTFS. The Orion 3 tool is then used to show how these techniques compare and improve the NoC's power consumption.

The remainder of this paper is organised as follows. Section II discusses the related work, and section III describes the NoC. Section IV discusses the proposed model for the TTFS solution. The use case description is presented in section V. Section VI discusses the experimental results, and finally, section VII concludes the paper.

This work has received funding from the ECSEL Joint Undertaking(JU) under grant agreement No 877056. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Italy, Austria, Germany, France, Finland, Switzerland

II. Related Work

DVFS is the most common method employed in MPSoC to save energy. Thus, the technique has been extensively researched to provide strategies for selecting the optimal voltage and frequency used for specific applications and architectures. Several approaches have been introduced before to improve the power consumption of MPSoC. Those techniques are applied in several devices as explained in the following works [13],[17],[11],[21],[6].

In [13], the authors proposed a data-driven frequency scaling technique by predicting a given application's power and execution over different clock settings. The data was first collected from application profiling and then used to train models with a regression-based machine learning technique for prediction. This method was applied to GPUs, and the results obtained showed a model prediction accuracy with average RMSE values of 0.38 and 0.5 for energy and time, respectively. The proposed method consumed 15% less energy compared to the baseline policies used in the investigation.

In [17], the authors investigated the state of the art of the most up-to-date dynamic voltage and frequency scaling techniques in low-power microcontrollers in wireless sensor networks. They measured DVFS on the MPS43542 platform. The result demonstrated the impact of DVFS on power consumption and the energy of the processor. The result showed that high voltage/frequency increased the normalized power up to 57 %, while 37% power is increased when only frequency scaling is considered. In contrast, our work focuses on frequency scaling in hardware NoC layers.

In [11], the authors proposed a method to scale the voltage and frequency of multiple cores by considering the trade-off between performance and energy consumption of multicore CPUs using ARMV8 microprocessors (X-Gene2 and X-Gene 3). An online monitoring daemon was used to monitor the running processes on the system. It was used to guide the Linux scheduler to allocate the processes to the appropriate cores. Simultaneously, the daemon was used to dynamically adjust the voltage and frequency settings according to the optimal policies. The result showed that 25% of average energy is saved using X-Gen2 with a minimal performance penalty of 3 %, and 22 % of energy is saved using X-Gene3 with a minimal performance penalty of 2%. In contrast, our work focused on applying frequency scaling at the routers of an NoC.

In [18], the authors showed that disregarding DVFS granularity as a design constraint in frequency scaling reduces the efficiency for certain application classes. Three Intel processors were used in the work to demonstrate the effect of disregarding DVFS granularity in design. The result showed that ignoring DVFS granularity might increase the design's power consumption by 19 %. The work focused on the core-level and not the NoC routers in contrast to our work. Although we did not investigate the frequency scaling granularity, it is considered for future work.

In [6], the authors proposed a Learning-enabled Energy-Aware Dynamic Voltage/frequency scaling method to improve the power consumption in NoCs. A machine learning technique was used to compensate for the trade-off between performance and energy consumption. This method enables a proactive energy management strategy that relies on an offline trained regression model and provides a wide variety of voltage/frequency models. The scheduler then uses the models to schedule the voltage and frequency in the NoC. The simulation result showed that 17 % of average dynamic energy is saved with minimal loss of 4% in throughput and no increase in latency. In contrast, our work focuses on NoCs that can be used to fulfill hard real-time requirements and low-power requirements. For this reason, Time-Triggered concept is adopted to provides a real-time guarantees for the on-chip communication and this concept is associated with source-based.

III. Network on a Chip

Network-on-Chip technology serves to interconnect several processing elements or modules on an integrated circuit like an MPSoC. A typical NoC-based MPSoC is shown in Figure 1. It comprises several components that can be referred to as nodes. The nodes include CPUs, custom IPs, DSPs, and storage elements (embedded memory blocks). All the nodes are connected via the Network Interfaces (NIs). The nodes send and receive packets through the network composed of routers and they are connected through physical links or channels. Typically, each link is a pair of opposite, unidirectional, point-to-point connection.

Routing is implemented in the communication protocol as a set of rules to avoid deadlock and livelock while reducing latency. The NI is the logic connecting the IP cores to the network, separating computation from communication [7]. Several protocols are used to connect IP cores to the network, such as the Open Core Protocol (OCP), Virtual Component Interface (VCI), Advanced eXtensible Interface (AXI), and Device Transaction Level (DTL) protocols [12] . For example, the Xilinx Versal programmable NoC implements an AXI interconnect for sharing data between IP endpoints in the programmable logic (PL), the processing system (PS), and other hard blocks [2].

Today, several topologies can be used in NoCs to improve the performance and power efficiency of design. Those topologies include ring, star, mesh, torus, fat tree, binary tree and butterfly topologies [8]. In the ring topology, tiles are linked using single wires in ring manner. This representation results in single points of failure. In the star topology, the tiles are linked to a central tile. In the mesh topology, tiles are linked together and are scalable, as seen in figure 1. The torus topology differs from the mesh topology, with the end tiles linked directly. The mesh topology is widely used due to its path diversity but has an upper limit on the number of router output ports [9].

A network of routers is used to connect cores. A crossbar switch is mainly used in the router design and serves for the physical interconnect between the input and output ports.

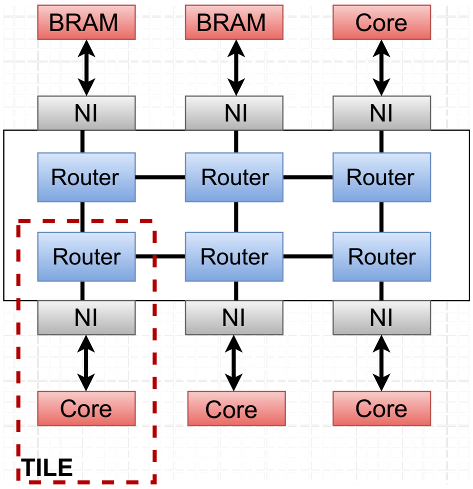


Fig. 1: NoC structure

The source-destination path could be either circuit-switched or packet-switched. In-circuit switching, a dedicated path is created for the duration of the transmission [19]. In packet switching, messages are divided into packets where, each packet contains a header flit with routing decisions.

Routing algorithms of NoC determine the path from the source through which flits or packets are delivered. Several routing algorithms can be used in NoC design, such as deterministic routing (e.g. XY, YX), adaptive routing, and source-based routing as described in [16] [5] [5] [5] [15].

IV. System Model of Adaptive Time-Triggered Frequency Scaling

A. Power Model

The power consumption of digital circuits can be observed in two phases. The first phase is the active phase, where the inputs are active, and output is produced. The second phase is the standby phase, where the chip is inactive. Both dynamic and static power dominate the power consumption of the chip during the active phase. Dynamic power is dissipated when a transistor switches from high to low, thereby producing an output.

The main factors contributing to the MPSoC power dissipation are dynamic power consumption, short circuit power consumption, and power loss due to the leakage power. The complexity of new generations of MPSoCs makes it hard to model the power consumption accurately. A viable approach for modelling the power consumption of NoCs is to model their building blocks components, which are mainly made of logic gates [20]. The total power in the digital CMOS circuit was described in [20]. It can be expressed using equation 1 below:

$$P_{total} = P_{Dyn} + P_{Stat} \tag{1}$$

Where P_{Dyn} is the dynamic power, and P_{Stat} is static power. The dynamic power is described in equation (2):

$$P_{Dyn} = P_{SW} + P_{SC} \tag{2}$$

P_{SW} is the switching power, and P_{SC} is the short circuit power. The switching power and the short circuit power are further described in equation (3) and (4).

$$P_{SW} = \alpha \cdot F \cdot C_L \cdot (V_{dd})^2 \tag{3}$$

Where α is the activity factor, F is the clock frequency , C_L is the load capacitance, and V_{dd} is the power voltage

$$P_{SC} = 10\% \cdot P_{dyn} \tag{4}$$

The static power is due to leakage current in each logic block.The static power is proportional to the supply voltage and can be express using the relation :

$$P_{Stat} \approx \beta \cdot V_{dd} \cdot e^{-V_{th}/\gamma \cdot V_T} \tag{5}$$

where β and γ are experimentally derived constants , and V_{dd} is the power voltage , and V_{th} is the threshold voltage and V_T is the Boltzmann thermal voltage that is linearly proportional to the temperature [10].

B. Time-Triggered Frequency Scaling

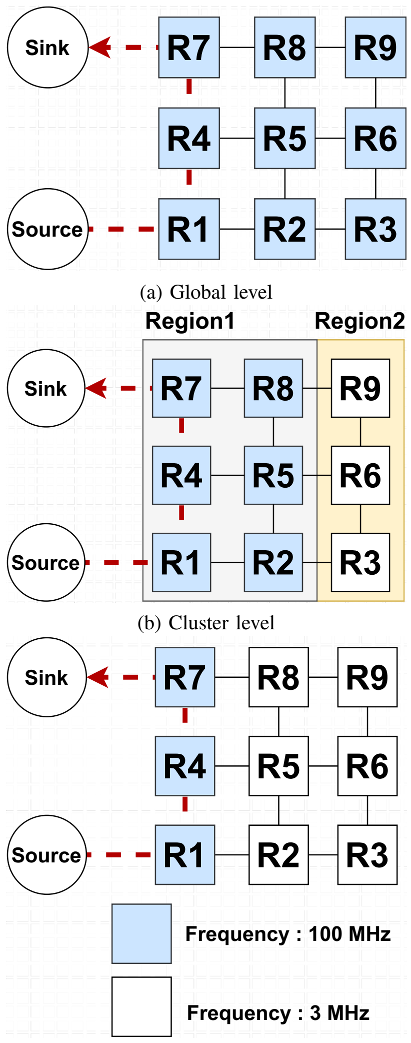
The operation of the NoC in this work consists of two states, an active state and an idle state. The active state describes when flits traverse the routers. The routers operate at a higher frequency in the active state. The idle state describes when there are no flits to be transmitted through the NoC. The router frequency is set at a lower value in this state. In our model, two types of communication paradigm are considered within the NoC; time-triggered and event-triggered communication. The major difference between these paradigms lies in the location of control in transmitting messages. In event-triggered communication, the decision when a message is transmitted via the NoC is controlled by an external event. In the time-triggered communication, the NoC network interface uses a schedule to control the transmission of messages within the NoC. Our model is built upon the mixed-criticality NoC prior work carried out in [3].

The TTFS is a mechanism that allocates the frequency at which each NoC router operates according to a time-triggered schedule. A time-triggered schedule is precomputed at design time to handle the transmission of flits within the NoC. The NoC network interfaces control the transmission of flits following the precomputed schedule. The time-triggered schedule also holds information about the idle state of the individual NoC routers. Therefore, the time-triggered schedule implicitly contains frequency control information which are the non-active times of each router. The TTFS scales up routers' frequency at runtime when a router is in the active state and scales down the routers' frequency when a router is in the idle state. In this way, the TTFS aims to provide low energy services using frequency scaling when no message is scheduled for transmission.

Furthermore, information about the message path is contained in the header flits for both event-triggered and time-triggered messages. Therefore, all the routers that fall within this path are known from the header flit's content. The energy control information is tied to the paths the flits will travel. The scheduled slots that permit the transmission of event-triggered and time-triggered messages on a schedule-timeline is known in advance. The TTFS uses the energy control information derived from the header flit to scale the affected router's frequency when a message is received from an external event.

Three approaches are described in this work to apply the TTFS to minimize the power consumption of the NoC. These approaches are illustrated using Figure 2. The first approach is scaling frequency at a global level, shown in Figure 2a. In this case, every router in the NoC operates at the same frequency for both the active state and the idle state. This technique is only feasible when all the routers have a common idle time. All the active routers (R1-R9) are switched to operate at the operational frequency (100 Mhz in the figure 2) during the active state. At idle times, the TTFS scheme simultaneously switches all routers to a lower frequency.

The second approach is the cluster level, shown in Figure 2b. In this case, the system establishes multiple regions. During the active state, the router frequencies in the region where transmission of flits occurs (R1, R2, R4, R5, R7, R8) are switched to the operational frequency. This is shown in region 1 of Figure 2. The router frequency in the region where no flit transmissions occur remain at a lower frequency. However, during the idle state, all the router frequencies are kept at a low frequency. The third approach is the route-based level, shown in Figure 2c. In this case, the TTFS switches only the router frequency in the flit path during the active state. The router frequencies (R7, R4, R1) are switched to 100 MHz in the example, while the other routers remain at 3MHz. However, during the idle state, all the routers are switched back to 3MHz.



(a) Global level

(b) Cluster level

(c) Route-based level

Fig. 2: Operation model for the TTFS

V. Use Case Description of Proposed Method

The building blocks of the TTFS are illustrated in Figure 3. The NoC implementation is an extension of lisNOC [1]. The added features are the source-based routing scheme and an AXI interface. Source-based routing is selected in this research since it can provide configurable flit navigation from source to sink compared to distributed routings such as XY routing and YX routing. Source-based routing can be deterministic, if the behavior of the NIs are deterministic. The NoC is used to connect different tiles. The master and slave module communicate through the NoC. The $TTNI$ module is the network interface between the core and the NoC. It is equipped with time-triggered capabilities such that messages between the master and slave module can be exchanged according to a global time. Several sub-modules are implemented to realize the $TTNI$. These modules include an NI memory Map region, TTscheduler, $Flitizer$, $Deflitizer$, and a frequency controller.

The NI memory map is used to map the cores to the NoC. The TX_AREA is used to buffer the message sent from a master. The TX_DATA holds a signal to trigger when the data contained in the TX_AREA should be sent. This trigger signal is received from the $TTScheduler$ module. When the TX_DATA is triggered, flits from the TX_AREA passes through the $Flitizer$, which converts the message into flits. The flits are then injected into the NoC. At the receiving end of the NoC, the flits are converted back into messages by the $Deflitizer$ module, after which the messages are stored in the RX_AREA of the NI Memory Map module. The cores can then access the messages following a global pre-computed schedule. The trigger signal for the message is fed by the $TTScheduler$ to the RX_READ_DATA area of the NI Memory Map. During the operation described above, the $FrequencyController$ configures each individual NoC routers' frequency according to a time-triggered schedule provided by the $TTScheduler$. The $TTScheduler$ is the core of the TTFS operation. Time-triggered schedules are pre-computed offline to ensure that there are no collisions between

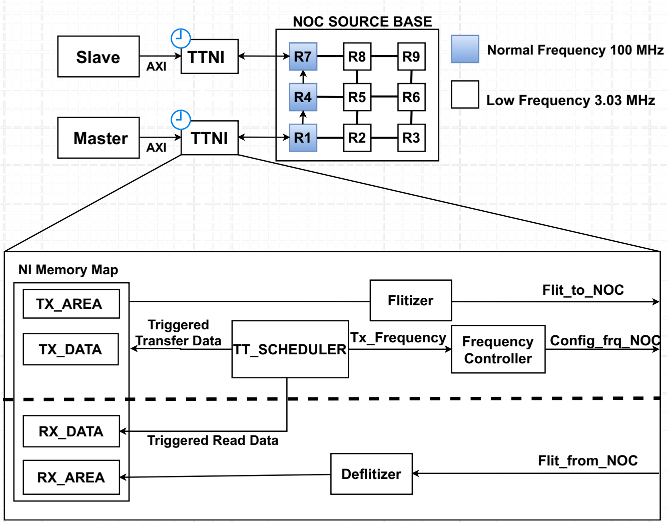


Fig. 3: 3 x 3 NoC along with Time Triggered Frequency Scaling

the exchanged time-triggered messages. In this way, safe and efficient resource usage is achieved. In Figure 3, the $Master$ is connected via the $TTNI$ to the router $R1$ of the NoC, and the $slave$ is connected to the router $R7$. To configure each router, the $FrequencyController$ in the example uses nine bits, one bit for each router. The frequency trigger signal is used to scale up and down the frequency of each individual routers. Two frequencies are used in this example, 100 MHz for the normal frequency and 3 MHz for the low frequency.

We evaluate the proposed TTFS scheme using three scenarios that reflect the global level, cluster level, and route-based level. According to a given time-triggered schedule described by the events in Figure 3, the investigations target the dynamic adjustment of the router's frequency to improve the power consumption of subsystems communication in the NoC.

We demonstrate the implementation with a 2D 3X3 mesh network consisting of five input port and five output port. The wormhole routing scheme was used and a 32-bit flit size. The voltage, virtual channel, and buffer input/output port is 1.5V, 1, 32-bit width, respectively.

VI. Result and Discussion

ORION3.0 tools is an open-source tool used to estimate the power consumption of micro-architecture, implementation, and operational parameters as well as multiple router RTL generators [4]. This tools is used here to simulate the NoC using the router features as described in section IV-B. The results demonstrate that power is saved using the three approaches (Global level, Regional level, Route-based Level) previously described. Figure 4 shows the result obtained from the simulation of the global level scenario. Without the TTFS scheme, the average power is 40.76mW. In applying the TTFS in a global level configuration, we obtained the static power, dynamic power, and average power as 10.92mW, 15.36mW, and 26.29 mW, respectively. This concept amounts to a 35.5% reduction in power consumption.

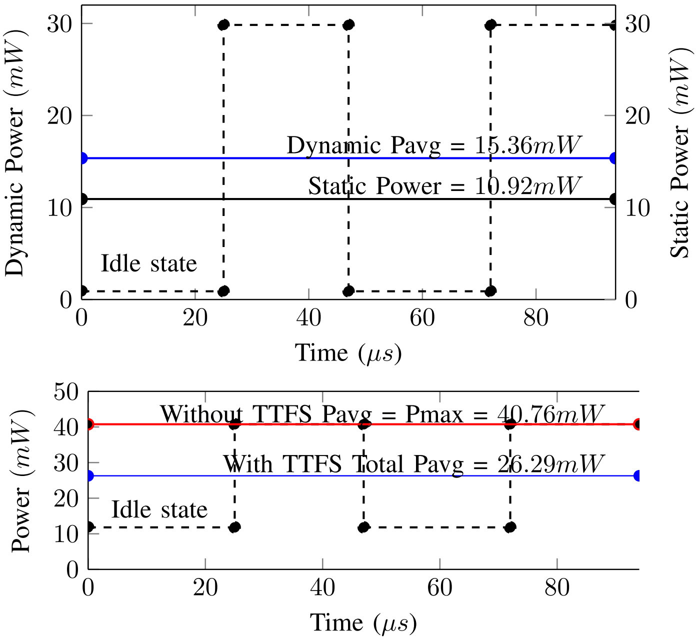


Fig. 4: TTFS result using the Global technique

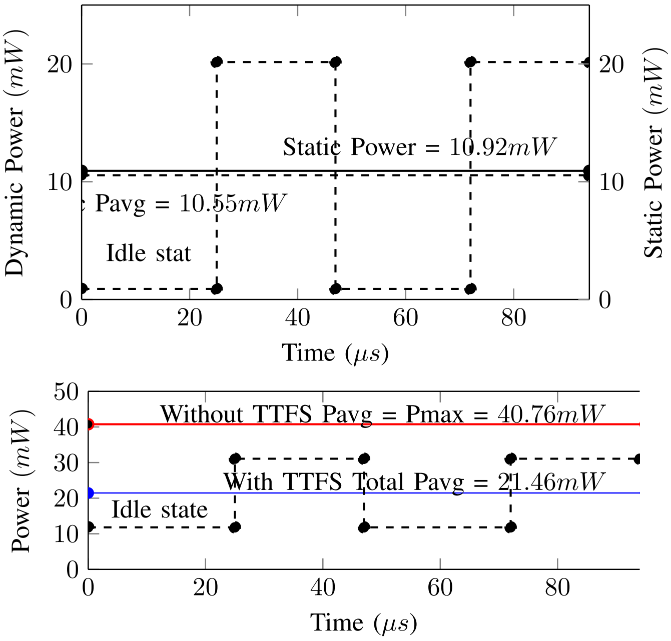


Fig. 5: TTFS result using the Cluster technique

Figure 5 shows the result obtained from the simulation of the cluster-based scenario. Without the TTFS scheme, the average power is 40.76mW. In applying the TTFS in a cluster-based configuration, we obtained the static power, dynamic power, and average power as 10.92mW, 10.55mW, and 21.46 mW, respectively. This concept amounts to a 47.3% reduction

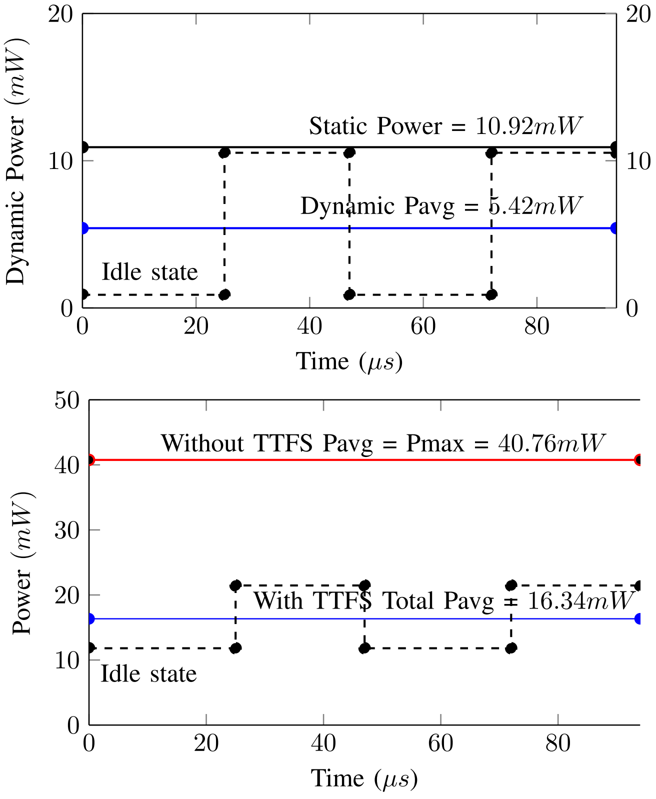


Fig. 6: TTFS result using the Route technique

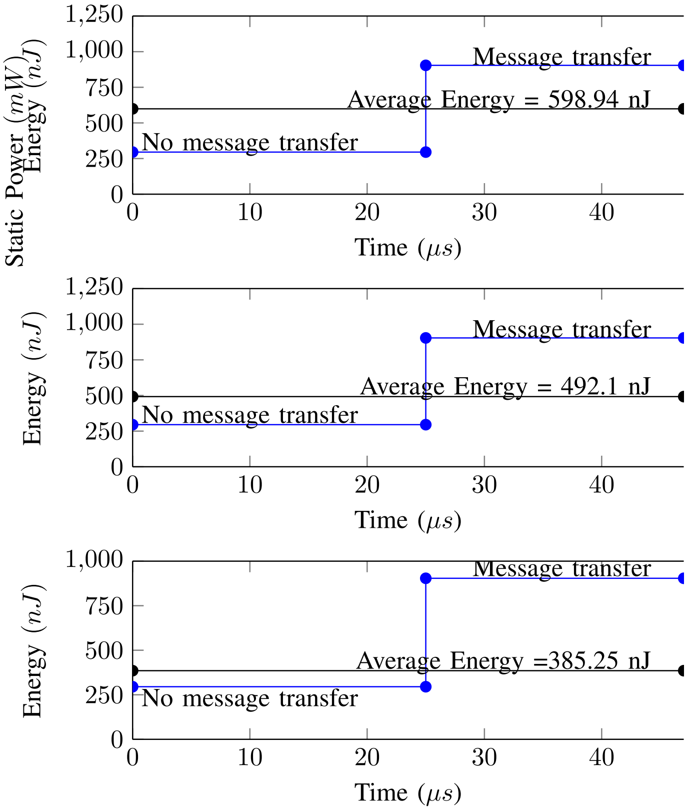


Fig. 7: Energy consumption vs time with TTFS implemented at different levels

in power consumption.

Figure 6 shows the result obtained from the simulation of the Route-based scenario. Without the TTFS scheme, the average power is 40.76mW. In applying the TTFS to the route-based configuration, we obtained the static power, dynamic power, and average power as 10.92mW, 5.42mW, and 16.34 mW, respectively. This concept amounts to a 59.9% reduction.

We plot the energy consumption to compare all three concepts in Figure 7. The average energy consumed by the global, cluster-based, and route-based technique is 598.94 nJ, 492.1 nJ, and 385.25 nJ, respectively. The route-based technique is the most energy-efficient from the results obtained.

VII. Conclusion and Future Works

This work proposes a novel technique to realize energy-efficient on NoC based on a Time- Triggered schedule. The proposed TTFS is simulated on ORION 3. The proposed models reduced the power consumption and preserved the system's safety by providing a temporal guarantee for critical messages. The simulation result shows that our proposed method reduces the dynamic power consumption up to 59% for a route-based model. Regarding future works, we intend to go beyond the route-based model to a path-based model where the TTFS scheme scales up individual router's frequency just before flit transmission. In addition, the granularity of TTFS will be considered to determine the impact on power consumption.

References

[1] lisnoc. http://www.lisnoc.org/. (accessed: 01.03.2021).

[2] Versal ACAP Programmable Network on Chip and Integrated Memory Controller v1.0. https://www.xilinx.com/support/documentation/ip_documentation/axi_noc/v1_0/pg313-network-on-chip.pdf. Accessed: 2021-02-09.

[3] H. Ahmadian and R. Obermaisser. Time-triggered extension layer for on-chip network interfaces in mixed-criticality systems. In *2015 Euromicro Conference on Digital System Design*, pages 693–699, 2015.

[4] Andrew B. Kahng, Bill Lin, Siddhartha Nath. ORION3.0: A comprehensive noc router estimation tool. *IEEE Embedded Systems Letters* , June 2015.

[5] Ashwini M. Dahule, P. Indurkar and Prof. S. D. Kamble. 2D Mesh Topology for Routing Algorithms in NoC Based on VBR and CBR. *International Research Journal of Engineering and Technology (IRJET)*, 2(5)(190), 2015.

[6] M. Clark, R. Bunescu, A. Kodi, and A. Louri. Lead: Learning-enabled energy-aware dynamic voltage/frequency scaling in nocs. In *2018 55th*

ACM/ESDA/IEEE Design Automation Conference (DAC), pages 1–6, 2018.

[7] T. P. F. e. Fizardo and R. Z. Dias. State of art Network on Chip. In *2nd International Conference on Signal Processing and Communication (ICSPC)*, pages 64–69, 2019.

[8] A. Kalita et al. A topology for network-on-chip. In *International Conference on Information Communication and Embedded Systems (ICICES)*, pages 1–7, 2016.

[9] N. Y. Phing et al. Towards high performance network-on-chip: A survey on enabling technologies, open issues and challenges. In *3rd International Conference on Electronic Design (ICED)*, pages 259–263, 2016.

[10] Gaillardon, Pierre-Emmanuel ; Beigné, Edith ; Lesecq, Suzanne ; De Micheli, Giovanni. A Survey on Low-Power Techniques with Emerging Technologies: From Devices to Systems. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, pages 12:1–12:26, 2015.

[11] George Papadimitriou, Athanasios Chatzidimitriou,Dimitris Gizopoulos. Adaptive Voltage/Frequency Scaling and Core Allocation for Balanced Energy and Performance on Multicore CPUs. *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2019.

[12] Raghu Prasad Gudla. Design and implementation of clocked open core protocol interfaces for intellectual property cores and on-chip network fabric. May 2011.

[13] S. Ilager, R. Muralidhar, K. Rammohanrao, and R. Buyya. A data-driven frequency scaling approach for deadline-aware energy efficient scheduling on graphics processing units (gpus). In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pages 579–588, 2020.

[14] Soininen JP Forsell M Millberg M Oberg J Tiensyrja K Hemani A Kumar S, Jantsch A. A network on chip architecture and design methodology. In *IEEE computer society annual symposium on VLSI*, page 105–112, 2002.

[15] N. Matveeva, Y. Sheynin, and E. Suvorova. Qos support in embedded networks and noc. In *Proceedings of 16th Conference of Open Innovations Association FRUCT*, pages 51–59, 2014.

[16] Wooshik Myung, Zhao Qi, and Ma Cheng. Performance analysis of routing algorithms in mesh based network on chip using booksim simulator. In *2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE)*, pages 297–300, 2019.

[17] Rym Chéour , Sabrine Khriji , Martin Götz , Mohamed Abid , Olfa Kanoun . Accurate Dynamic Voltage and Frequency Scaling Measurement for Low-Power Microcontrollors in Wireless Sensor Networks. *Microelectronics Journal 105 (2020) 104874*, 11 August 2020.

[18] Vaibhav Sundriyal , Kristopher Keipert, Masha Sosonkina and Mark S Gordon. Effect of frequency scaling granularity on energy-saving strategies. *International Journal of High performance computing applications)*, 2019.

[19] Vikrant A. Bute et al. Review on Network on Chip Router Design. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 5(3), 2014.

[20] Vitor R.G.Silva,Alex F.A.Furtunato,Kyriakos Georgiou,Samuel Xavier-de-Souza, Carlos A.V.Sakuyama,Kerstin Eder. Energy-Optimal Configurations for Single-Node HPC Applications. *2019 International Conference on High Performance Computing & Simulation (HPCS))*, 2018.

[21] Zhimin Wang, Qinglin Zhao,Li Feng , and Fangxin Xu. How Much Benefit Can Dynamic Frequency Scaling Bring to WIFI. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, March 2021.