

Efficient Simulation and Modelling of Non-rectangular NoC Topologies

Ji Qi and Mark Zwolinski

Electronics and Computer Science, University of Southampton, UK
{jq2e09,mz}@ecs.soton.ac.uk

Abstract—With increasing chip complexity, Networks-on-Chips (NoCs) are becoming a central platform for future on-chip communications. Many regular NoC architectures have been proposed to eliminate the communication bottlenecks of traditional bus-based networks. Non-rectangular and irregular architectures have also been proposed to increase performance. However, the complexity of designing custom non-rectangular networks leads to a rapid increase in design and verification times. To alleviate the conflict between performance and efficiency, this paper proposes a novel method that efficiently constructs virtual non-rectangular topologies on a mesh network by using time-regulated models to emulate irregular patterns. Data routings on virtual hexagonal and two irregular geometries validate the proposed method. An MPEG-4 decoder is used to exemplify the proposed method for media applications. Results analysis shows the virtual topologies emulated by the proposed method can provide precise timing and energy performance.

I. INTRODUCTION

With IC feature size continually scaling down and increasing transistors counts, on-chip communication becomes more difficult as traditional bus-based architectures cannot cope with increasing system complexity. Networks-on-chip, in which routers are connected between pre-designed building blocks, are considered a promising platform for future systems. Designing suitable NoCs is of high priority because of their impact on system performance and design cost. As chip complexity increases, however, higher module density and specialization makes it harder for modular NoC designs to achieve the target performance and for custom NoC systems to be implemented in acceptable design times. The gap between design for performance and efficiency is thus widening.

Efficient latency and low energy consumption are important to NoCs. Such design aims have been well supported by many NoC simulators in constructing rectangular topologies, such as mesh and torus, due to their high regularity and scalability. Yet non-rectangular topologies such as hexagon, though also popular in wireless communication fields, are often built in custom ways such that modules and links are specifically designed [1], [2]. Such methods for non-rectangular topological construction are not cost efficient with increasing network complexity. To our best knowledge, there are no general methods for the construction of non-rectangular topologies, with accurate timing and energy performance. We present a novel method to form virtual non-rectangular and irregular routing structures based on a rectangular mesh network. The method models accurate timing and energy performance in the NIRGAM simulator [3]. To validate the proposed method, a honeycomb hexagonal topology [1] and two different irregular geometries

were constructed in the conventional custom manner and virtually using the proposed method. The data routing latency and energy costs using the two methods are compared. An MPEG-4 decoder is also used to validate the proposed method.

II. RELATED WORK

A number of papers on constructing irregular network topologies [4], [5], [6], [7] and developing NoC simulators [8], [9], [10], [11], [12] have been published in the past decade. Several design methods were proposed for constructing either special irregular topologies: Ring, Octagon and 2-hop or custom irregular networks tailored to specific applications. A number of NoC simulators were also developed to offer self-built regular NoC topologies and some of them can integrate user-designed network topologies as extensions. These works present various ways of constructing non-rectangular and irregular NoC topologies. Yet none of them offers a productive method with accurate performance measurements in terms of timing and energy. Inspired by their work, configurable time regulated models have been designed in our proposed method to efficiently establish virtual non-rectangular and irregular routing geometries on a mesh network. The NIRGAM simulator was chosen as the platform for our method because of its extensibility and performance measurement capability.

III. PROPOSED MODEL AND METHOD

A network topology defines the interconnection of processing nodes and is significant for network efficiency. A rectangular mesh is one of the most commonly used NoC topologies. Its mutually independent structure can be easily expanded to incorporate new models without changes to existing parts. In our proposed method, a data transceiving model called the *time-regulated model* has been designed and mapped to network nodes in the NIRGAM simulator. The principle is to adjust the data processing time to account for unequal transmission times between adjacent nodes. This is achieved by regulating the data sending time and injection rate at the outputs of a node router. Specifically, when a data flit (a small unit that forms a packet) is arbitrated by a node router and sent to an output port, the *time-regulated model* can precisely change the waiting time by inserting longer or shorter time periods as time delays or time advances in the write cycles of node buffers. Since the inserted time period is user-defined and derived from the global clock, clock jitter can be avoided and different flit processing time periods can be adjusted to form specific transmitting distance. Hence, any desired topological shapes can be emulated *virtually* on a rectangular mesh.

Functional validation and performance measurement for the proposed method are considered in the NIRGAM simulator. NIRGAM [3] is a discrete-event and cycle-accurate simulator that supports extensions of user-designed topologies, routing algorithms and applications. A dynamic trace-based mechanism is used for timing measurement. The Orion power model [13], [14] incorporated in NIRGAM provides power estimation by extracting architectural parameter statistics for network routers and links. Since power and timing are modelled, modifications to Orion are made to estimate energy.

Between any node pair in a network, a traversal flit has the relationship: $Throughput = Velocity \times Latency$. $Throughput$ indicates the total number of flit bits traversing between node pairs over a specific time period. $Velocity$ indicates the number of bits of a flit traversing per second and $Latency$ indicates the time consumed between the first bit of a flit leaving one node and the last bit leaving the paired node. The flit velocity is only determined by the channel capacity between node pairs. Once the network channel bandwidth is fixed, the number of bits traversing per second is constant so the flit velocity is fixed. In this case, the traversal distance of flits between node pairs is only determined by the throughput, which is in turn determined by the flit latency. Since a mesh network has the same flit velocity over the whole network, the flit traversal distance is determined by the routing latency, which can be precisely adjusted by using the time-regulated model.

Having different time periods enables flits traversing in given patterns to form specific *virtual* topological shapes, such as SPIN, Spidergon, Hexagon and Octagon. If the values of time spans are set more randomly, virtual pseudo-irregular routing geometries can be emulated. As an example, the virtual formation of a hexagonal network using time-regulated models is given in Figure 1(a). Assume each hexagon edge has a length of r units (data traversal distance). Flits travelling between nodes R_0 and R_2 should spend the same time as between R_2 and R_3 , which implies the same traversal distance. If we denote the flit traversal times between node pairs $R_0 - R_1$, $R_1 - R_2$, $R_0 - R_2$ and $R_2 - R_3$ as t_1, t_2, t_3 and t_4 , respectively, the time relation should be: $t_1 + t_2 = t_3 = t_4$. To form virtual hexagonal edges, time regulated models are inserted at nodes R_0 , R_2 and R_3 . Figure 1(b) gives the timeline of a flit traversal. In a regular mesh network, the flit traversal time period from node R_0 to R_2 (from f_S at R_0 to f_S at R_2) is bigger than that between $R_2 - R_3$: ($t_1 + t_2 > t_3$), but their in-node processing time spans (from f_R to f_S in the same nodes) are equal: ($ts_1 + ts_2 = ts_3$). To form equal hexagonal edges, a shorter wait time is set at node R_2 ($ts'_2 < ts_2$), which causes a send time advance and reduces the value of t_2 in the time line ($t'_2 < t_2$). A longer wait time is set at node R_3 ($ts'_3 > ts_3$), which causes a send time delay and increases the value of t_3 ($t'_3 > t_3$). Thus, the latency at node pairs $R_0 - R_2$ and $R_2 - R_3$ has been equalised ($t_1 + t'_2 = t'_3$) while their total traversal latency remains unchanged ($t_1 + t_2 + t_3 = t_1 + t'_2 + t'_3$). Similarly, values of the flit traversal time in node pairs $R_3 - R_4$, $R_4 - R_5$, $R_5 - R_6$ and $R_6 - R_0$ are also equalised manually. With a fixed flit velocity, the flit traversal latency between those node pairs defines equal traversal distances, which forms a virtual hexagonal topology. Using the same principle, other non-rectangular and irregular routing patterns can also be emulated virtually.

As a reference, a hexagonal topology was also constructed

using the conventional method. Using a mesh coordinate system, the homogeneous nodes in Figure 1(a) are re-written to give 3 directional links (60° North of East, 60° South of East, West or 60° North of West, 60° South of West and East) instead of the default 4 links (North, South, West and East) to neighbouring nodes. Other mesh nodes, not used by an irregular topology, are disconnected and the bandwidth values on remaining node pairs are specified to yield node links with various channel capacity ratios.

IV. EXPERIMENTAL EVALUATION

A. Hexagonal and Irregular Routings

To validate the functionality of the proposed method, hexagonal and irregular networks were constructed on a 9 by 9 mesh in the NIRGAM simulator. A manifest architectural difference between hexagonal and mesh networks is the smaller diameter (hops of traversing the network critical path) of a hexagon, if they have the same connected nodes [15]. The network diameter directly affects the flit transmission time. Hence, a source-destination node pair having a fixed Euclidean distance is designed on the virtual hexagon and mesh geometries. Two different irregular routings with the same nodes as the hexagonal network are also implemented to verify the feasibility of proposed method in forming irregular geometries. The time and energy performance accuracy of the proposed method is validated by comparing with “physical” networks constructed using the conventional method.

Data routings in virtual hexagonal and mesh geometries are shown in Figure. 1(c). The hexagonal edge length (radius) is equal to r units. The source and destination coordinates are set to $[0, 1]$ and $[8, 5]$ respectively. The hexagonal routing distance is then derived in (1). On a rectangular mesh, flits need to travel $\frac{\sqrt{3}}{2}r$ units vertically and $\frac{1}{2}r$ units horizontally to reach the first Euclidean point as in the hexagon. Then they need to traverse r units horizontally to reach the second Euclidean point. After repeating the above process 4 times they reach the destination. The Manhattan distance is given in (2). It is clear the routing distance between fixed source and destination nodes in a mesh is longer than in a hexagon. Equation (3) gives the Euclidean distance (straight line from source to destination), which is also the shortest routing path.

Figure 1(d) shows two irregular routings with the same number of nodes as the hexagonal routing, for comparison. The node coordinates are set using the time-regulated models as shown in the figure. In both irregular networks, the black dash lines, red dotted lines and red solid lines indicate the straight line, hexagonal routing and virtual irregular routings respectively. The irregular 1 path is closer to the Euclidean distance than the hexagonal routing, while the irregular 2 path is further. This suggests a shorter distance for irregular 1 and a longer distance for irregular 2 than the hexagonal routing. In the rectangular mesh coordinate system, their distances are calculated in (4) and (5), respectively.

$$S - D_{[H]} = 8r \quad \text{units} \quad (1)$$

$$S - D_{[M]} \approx 9.5r \quad \text{units} \quad (2)$$

$$S - D_{[E]} \approx 6.9r \quad \text{units} \quad (3)$$

$$S - D_{[I1]} \approx 7.5r \quad \text{units} \quad (4)$$

$$S - D_{[I2]} \approx 11.6r \quad \text{units} \quad (5)$$

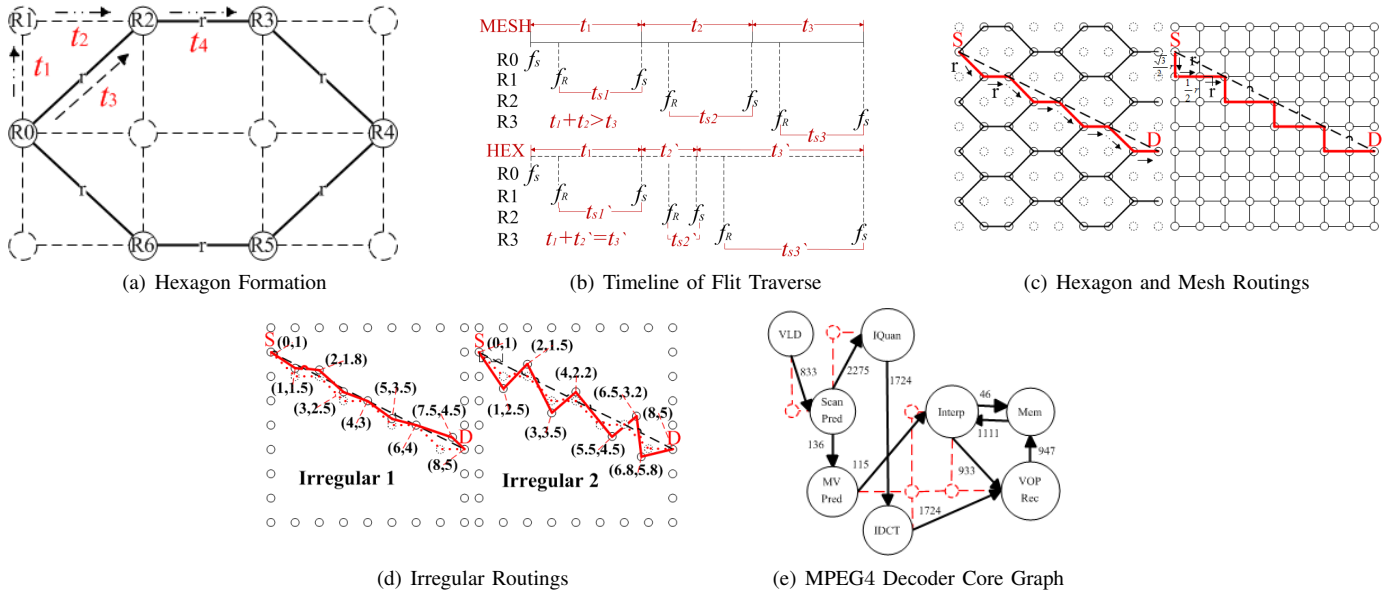


Fig. 1: Construction of Mesh-based Hexagon, Hexagonal, Mesh and Irregular Routings and MPEG4 Decoder

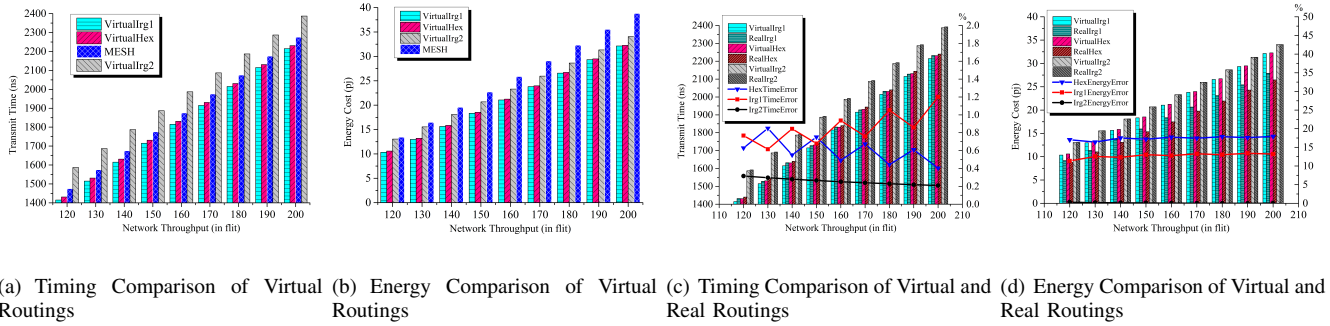


Fig. 2: Routing Comparisons of Virtual and Real Topologies

B. MPEG-4 Decoder Simulation

To evaluate the potential of the proposed method for real multimedia applications, an MPEG-4 decoder was mapped onto a custom network and a mesh network with virtual irregular routings respectively. Both networks are constructed on a 9 by 9 mesh. The timing and energy performance of the decoding process on both networks are compared in NIRGAM.

Figure 1(e) gives the core graph of the MPEG-4 decoder and the bandwidth requirements of each core. Five core pairs ([VLD-Scan], [Scan-IQuan], [MV-Interp], [IDCT-Rec] and [Interp-Rec]) in the decoder core map have irregular routings. To evaluate the proposed method, both custom network and virtual mesh network with irregular routings are designed according to the decoder mapping. For the custom network, a customised mesh with 8 instead of the default 4 directional links to neighbouring nodes was designed, which enables decoder cores to be mapped as per the core graph. The required values of transmitting bandwidth between core pairs are modelled by configuring consistent ratios of channel capacity to related node links. Unused mesh nodes and links are disconnected. The mesh network with virtual irregular routings

using our proposed method was also designed. Red dashed circles in Figure. 1(e) represents the nodes attaching time-regulated models. Red dashed lines show the routing paths on the mesh for emulating the decoder irregular routings. Five nodes with time-regulated models are placed on Manhattan routing paths between those cores.

Compared to the custom network, the virtual irregular routing between the five core pairs on the mesh consume at least 5 more hops (in the case of the [MV-Interp] core pair, 2 more hops) due to the Manhattan routing of the mesh networks. Hence, the cores in the MPEG-4 decoder are mapped more compactly onto the mesh compared with the custom network. The global channel capacity of the mesh is also reduced proportionally to ensure the decoder bandwidth usage is consistent with that used in the custom network.

C. Results Analysis

Constant flit traffic at 1 flit per 10 clock cycles was injected into all topological routings to avoid potential transmit interruptions and packet drops. Data packets containing 1 flit per packet with packets from 120 to 200 (incremental intervals

TABLE I: Performance Comparisons of MPEG-4 Decoding

Performance	Mesh with Virtual Routings	Custom Network	Error (%)
Time (μs)	4131.795	4137.157	0.13
Energy (pJ)	154.59	149.982	3.07

of 10 packets) were routed on all networks. The different flit injections aim to eliminate any artefacts in the results.

Figure 2(a) shows the time comparisons of data routings on the virtual hexagon, two irregular and mesh geometries. The time cost in all virtual topologies increases linearly with the packet increase. Irregular 1 routing has the least time cost, followed by hexagonal, mesh and irregular 2 routings. Figure 2(b) shows the energy cost of the four virtual routings. Virtual irregular 1 routing consumes the least energy followed by hexagonal, irregular 2 and mesh routings, respectively. Less time but more energy is consumed by the mesh routing than the virtual irregular 2 routing is due to more flit hops in mesh routing. Since the two irregular routings use the same nodes as the hexagonal routing, they will have the same dynamic energy but different static energy which is proportional to their routing time. Thus, though the irregular 2 routing costs more static energy due to longer routing time, extra dynamic energy cost by more mesh hops outweighs the static energy gap.

The above experiments verify that virtual routing geometries emulated by the proposed method have timing performances consistent with the geometric distances. The following experiments are designed to validate the accuracy of proposed method. Figure 2(c) and Figure 2(d) compare the timing and energy performance with the “physical” hexagonal, irregular 1 and irregular 2 networks, as implemented using the conventional method. As shown, the timings of the virtual routings are, respectively, 0.6%, 0.9% and 0.3% different from the “physical” routings. The energy of the virtual routings are about 17%, 12% and 0.2% different from the “physical” routings, respectively. The energy inaccuracy of hexagonal and irregular 1 routing pairs is due to their different flit hop costs; that is, the virtual routings require more flit hops than the “physical” routings, which costs more dynamic energy. The smaller energy difference of irregular 1 (12%) than the hexagonal routing pair (17%) is due to the fewer *extra hops* used in the virtual irregular 1 routing (2 extra compared with 4 extra hops). The number of hops between the virtual and real irregular 2 routings is equal (both 8 hops), giving an accurate energy performance (0.2% error). The energy results indicate the virtual routings emulated by the proposed method can provide reasonable energy performance accuracy.

For the multimedia application, the encoded video bitstream of a QCIF-format MPEG-4 video clip with 1s of content and 25fps frame rate is decoded on the two networks. Table I lists the performance comparisons for decoding on the “physical” custom and mesh networks. By using the proposed method to model irregular routings on a mesh, the performance is emulated with 0.13% timing error and 3.07% energy error compared to the performance of realistic case. The accurate timing emulation of the MPEG-4 decoder given by the proposed method is due to the precise timing regulation of the irregular routings and configuration of the channel capacity. Although an approximation of the total decoding hops on the

virtual mesh to those on the custom network has been made, there is still 1 extra hop between the [Interp-Rec] core pair on the mesh, which results in the 3% error in energy.

V. CONCLUSION

A method has been proposed to construct custom topologies based on a mesh network. The method virtually emulates a hexagon and two irregular topologies by using time-regulated models to configure the flit send time at nodes of a mesh network. Conventional hexagonal and irregular networks are also constructed in NIRGAM to validate our method. Fixed source-destination data routings in all virtual and “physical” networks are compared in respect of their timing and energy performance. Irregular core deployment of MPEG-4 decoder is virtually emulated and “physically” implemented to validate our method for media applications. The results show that our proposed method can efficiently construct virtual non-rectangular and irregular geometries with accurate timing. It can also provide accurate energy estimates if the nodes of the virtual networks are reasonably close to their real counterparts.

REFERENCES

- [1] I. Stojmenovic, “Honeycomb networks: Topological properties and communication algorithms,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 8, no. 10, pp. 1036–1042, 1997.
- [2] A. W. Yin, T. C. Xu, P. Liljeberg, and H. Tenhunen, “Explorations of honeycomb topologies for network-on-chip,” *NPC’09*, pp. 73–79, 2009.
- [3] L. Jain, B. M. Al-Hashimi, M. S. Gaur, V. Laxmi, and A. Narayanan, “NIRGAM: a simulator for noc interconnect routing and application modeling,” *DATE’07*, 2007.
- [4] W. Chen, D. Jin, and L. Zeng, “Heterogeneous design methodology with configurable regular topology set for salable network-on-chip designs,” *ASICON’07*, pp. 1293–1296, 2007.
- [5] T. A. Bartic, J. Y. Mignolet, V. Nollet, T. Marescaux, D. Verkest, S. Vernalde, and R. Lauwereins, “Topology adaptive network-on-chip design and implementation,” *IEEE Proceedings of Computers and Digital Techniques*, vol. 152, no. 4, pp. 467–472, 2005.
- [6] C. Neeb and N. Wehn, “Designing efficient irregular networks for heterogeneous systems-on-chip,” *Journal of Systems Architecture*, vol. 54, no. 3-4, pp. 384–396, 2008.
- [7] N. Choudhary, M. S. Gaur, and V. Laxmi, “Irregular noc simulation framework: IrNIRGAM,” *ETNCC’11*, pp. 1–5, 2011.
- [8] J. Xu, W. Wolf, J. Henkel, and S. Chakradhar, “A design methodology for application-specific networks-on-chip,” *TECS’06*, vol. 5, no. 2, pp. 263–280, 2006.
- [9] L. Bononi and N. Concer, “Simulation and analysis of network on chip architectures: ring, spidergon and 2d mesh,” *Proceedings of DATE: Designers’ forum*, pp. 154–159, 2006.
- [10] M. Karl, “A comparison of the architecture of network simulators ns-2 and tossim,” *Proceedings of Performance Simulation of Algorithms and Protocols Seminar*, 2005.
- [11] G. N. Khan and V. Dumitriu, “A modeling tool for simulating and design of on-chip network systems,” *Microprocessors and Microsystems*, vol. 34, no. 2-4, pp. 84–95, 2010.
- [12] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch, “The nostrum backbone—a communication protocol stack for networks on chip,” *VLSID’04*, pp. 693–696, 2004.
- [13] H. S. Wang, X. Zhu, L. S. Peh, and S. Malik, “Orion: a power-performance simulator for interconnection networks,” *MICRO-35*, pp. 294–305, 2002.
- [14] A. B. Kahng, B. Li, L. S. Peh, and K. Samadi, “Orion 2.0: a fast and accurate noc power and area model for early-stage design space exploration,” *DATE’09*, pp. 423–428, 2009.
- [15] Y. Xie, Z. Li, and F. Li, “Hexagon mesh interconnection networks,” *SPIE’05*, vol. 5626, pp. 991–998, 2005.