Microelectronics Journal 108 (2021) 104964

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

Design of a reconfigurable network-on-chip for next generation FPGAs using Dynamic Partial Reconfiguration

Ahmed Ramy [a,b], Hassan Mostafa [b,c,*], A.H. Khalil [b]

[a] *A Siemens Business Corporation, Egypt*

[b] *Electronics and Communications Engineering Department, Cairo University, Giza, 12613, Egypt*

[c] *University of Science and technology, Nanotechnology and Nanoelectronics Program, Zewail City of Science and Technology, October Gardens, 6th of October, Giza 12578, Egypt*

ARTICLE INFO

Index Terms:
Network on chip
Dynamic partial reconfiguration
Field programmable gate arrays

ABSTRACT

Introducing the reconfigurability concept into one of the most ramping and trending design platforms like the NoC is considered a good opportunity for gaining the most out of them. The high flexibility and full customization of the reconfigurable NoC could open the door for a completely adaptive NoC that suits a large number of benchmarks according to runtime needs and requirements.

The main objective of this work is to present the Dynamic Partial Reconfiguration (DPR) support to CONNECT Network-on-Chip (NoC) for Field Programmable Gate Array (FPGA) applications. It also analyzes the effect of this reconfigurability on the performance of the network and how reconfigurability could lead to area and power saving. Reconfigurability during runtime leads to more flexible NoCs and enables full customization for dynamic reconfigurable applications. In comparison with static NoCs, dynamically reconfigurable NoCs achieve more area utilization by reusing a part of the network area resources when it is not required during runtime. A reconfiguration tool is developed helping the designer to decide the optimal network structure for every application used. The reconfiguration tool requires as inputs the minimum needed throughput and the expected traffic load. Those inputs are used to decide the best network configuration and the minimum area that achieves those requirements.

1. Introduction

Design area and throughput of System-on-Chips (SoC) are among the most important metrics that need to be considered while planning an architecture for a SoC. With the complexity of designs, NoC design architecture has appeared as an optimum candidate for an on-chip platform that can be customized according to the application requirements. NoC design approach is preferred over conventional bus-based communication for its improved modularity, scalability, and performance [1].

Meanwhile, the advancement in dynamically reconfigurable FPGAs lets the hardware designs to be reconfigured during runtime. DPR offers more flexibility to hardware modules and allows better area utilization and more power optimization. Moreover, using DPR results in the development of adaptive hardware algorithms depending on each benchmarks constraints and requirements [2].

Studying reconfigurability techniques offered by FPGA providers (Xilinx and Altera) helps in analyzing strengths and weaknesses of different methods with different design sizes. Moreover, taking into consideration the reconfigurability constraints helps in the SoC design phase. This leads to maximizing the gain earned from targeting reconfigurable architectures.

Introducing the DPR capability to the NoC offers new opportunities for network topology customization according to the runtime requirements of the system. Runtime self-adaptive NoCs are beneficial when used with a configurable hardware design. The configurable hardware has multiple applications and various performance requirements, and this flexibility is not applicable in fixed NoCs.

This paper provides an example to a static NoC that is converted into a dynamically reconfigurable NoC. This dynamic NoC can be adapted during runtime to fit the current application within the DPR flow. Therefore, the user can apply the DPR into the NoC and the rest of the system as well.

This paper is organized as follows: In Section II, the previous work in dynamic reconfigurable NoCs is reviewed. Section III presents an

[*] Corresponding author. Electronics and Communications Engineering Department, Cairo University, Giza, 12613, Egypt.
E-mail address: hmostafa@uwaterloo.ca (H. Mostafa).

https://doi.org/10.1016/j.mejo.2020.104964
Received 7 May 2020; Received in revised form 6 November 2020; Accepted 7 December 2020
Available online 5 January 2021

overview of CONNECT NoC platform used and introducing the dynamic reconfiguration capability to it. Section IV describes the test environment and the reconfiguration tool for evaluation. The results and performance study are discussed in Section V. Section VI describes a case study highlighting the main benefits of reconfigurable NoCs. Section VII provides design guidelines and recommendations for designers of reconfigurable NoCs. Finally, Section VIII provides the conclusion of this work and future work proposed.

2. Previous work

The value of NoC to the hardware implementations can appear in the work presented in Ref. [3]. In this work, the well-known k-means clustering algorithm is built over a NoC-based architecture. Many other adaptive algorithms require the existence of NoC in their hardware implementations.

Several architectures of different NoCs are achieving a high level of reconfigurability. However, most of the reconfigurable NoCs related work is addressing design time not runtime reconfigurability. Nevertheless, some reconfigurable NoCs face different dynamic communication issues such as bypassing or surrounding the obstacles at runtime and how to dynamically manage the routing adaptation.

The proposed work in Ref. [4] introduces a NoC that manages circuit routing for the dynamically reconfigurable devices and explains how this approach is more effective than the legacy bus-based communication architectures. The work described in Ref. [5] introduces a CoNoChi NoC with a minimal area overhead and number of switches with a unique deadlock-free mechanism. In addition, the authors offered two reconfiguration methods, with and without the NoC stalled.

In [6,7], the authors presented DyNoC that offers advanced dynamic capabilities and a deadlock-free routing mechanism in order to guarantee the reachability to all the pins and blocks. The work achieved this by expanding the well-known XY routing algorithm to an S-XY (Surrounding XY) routing algorithm which is depending on surrounding the existing obstacles horizontally or vertically at runtime for reaching a deadlock-free routing mechanism.

The reconfigurability of the NoC in Ref. [8] is achieved by routers placing or removal, using adaptive routing tables to guarantee full connectivity, which enables every router to be accessed by its neighbor routers. Moreover, network updates are transmitted through unique packets from a central control unit. In Ref. [9], the reconfigurability of NoC is based on monitoring the local traffic and calculating the path weight. The calculations are given to a global arbiter for the minimum cost path selection.

ReNoC presented in Ref. [10] is considered as a generic structural form that can be combined with NoC routers and topology. ReNoC reconfigurability is achieved by a using packet switching and physical circuit switching. Moreover, the work in Refs. [11,12] addresses the reconfigurability of application-specific NoCs and how topology synthesis is performed in dynamically reconfigurable FPGAs. On the other hand, some efforts were exerted in power reduction area like the work in Ref. [13] which proposes a NoC power saving example through using application-specific Radio Frequency (RF) interconnects.

3. Reconfigurability of CONNECT NoC

The proposed work is achieved by using CONNECT as an FPGA-oriented on-chip network [14]. The low latency of CONNECT results in achieving an enhanced network performance for FPGA-based designs. Although, this single clock latency requires execution of all the different network stages in a single clock cycle which will create a critical path during compilation. Moreover, CONNECT provides a fully customizable router design and simple network allocation, routing, and flow control mechanisms. The network designer can generate the required NoC using a Register Transfer Level (RTL) configurator where several network configurations are supported. These diverse configurations depend on network size, network topology, buffer depth, number of virtual channels, flow control type, allocation type, and data width [15].

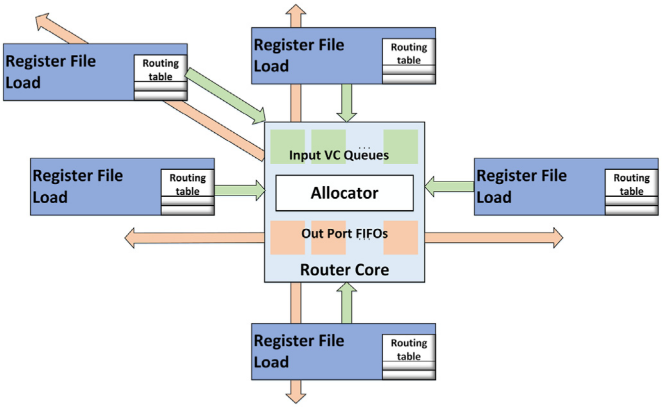


Fig. 1. CONNECT NoC router core internal structure.

The internal structure of CONNECT is based on optimized RTL implementation for different modules in a scalable manner in order to satisfy its customizable requirements. Fig. 1 shows the main internal structure for CONNECT router core. Each router core can communicate with four neighbor routers in addition to the outside user port which makes them five open communication channels. The core contains an input handler for routing the input packets and input queues for internal storage. Output port FIFOs are also required for dealing with neighbor routers availability.

The used network has the characteristics of a 4 × 4 mesh network with credit-based flow control, two virtual channels, and a buffer depth ranging from 4 to 64. The 4 × 4 mesh CONNECT is recommended by the authors in Ref. [16] to be with two virtual channels. The authors in Ref. [16] recommend not using more than two virtual channels as the third and fourth virtual channels impact the performance very slightly. This recommended network is achieving optimal performance with the best possible throughput and the lowest possible latency at dense injection rates.

The network reconfiguration at runtime requires dynamic routing adaptation; every router node needs to be aware of the availability of the neighbor routers. Accordingly, the used CONNECT RTL is changed for introducing the runtime reconfigurability of each router node. The RTL changes are as follows:

- An interface input configuration is added for indicating the current network status of each router. Each bit corresponds to the status of a single router whether it is active and normally functioning with the neighbor routers or inactive and being bypassed by the other nodes.
- The routing technique used in all the routers is adapted considering the chance that the neighbor router might be inactive. In this case some packets might be required to change the route. This change adds some area overhead by adding new routing tables in addition to preserving the original routing tables. All routing potentials are considered covering the four routers possible reconfigurations.
- Switches around every router for connectivity are added in order to disconnect the inactive neighbor routers in case of reconfiguring any of them. These switches are responsible for connecting with the nearest active router node.

Those changes enable the runtime reconfigurability of the CONNECT NoC. The global input decides the desired configuration while the switches adapt the connections between every router and its surroundings. The routing adaptation helps in solving any possible deadlock situation caused by the absence of any router when reconfigured.

The ordinary XY routing algorithm is suitable for static network configurations. The dynamically changed network could face the sudden appearance or absence of any router. These dynamic changes to the

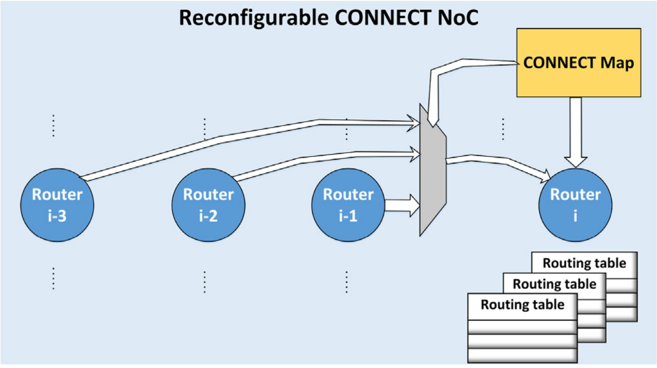


Fig. 2. Reconfigurable CONNEC NoC structure after the RTL changes.

network would lead to some deadlock situations with the XY algorithm. The routing technique changes are made to handle those possible deadlock situations. These changes reflects into the routing tables to dictate the available routing path from the current configuration.

The RTL changes added an area overhead to the CONNECT RTL implementation. The area overhead is for the added network map module, connectivity switches, and the routing adapted Look up Tables (LUTs). The new structure for the Reconfigurable CONNECT NoC is shown in Fig. 2. All these RTL changes are described and discussed in Ref. [17]. This paper completes the analysis provided in Ref. [17] with respect to flow control and area. Moreover, this paper introduces the reconfiguration tool development and provides a case study to highlight the benefit from this work.

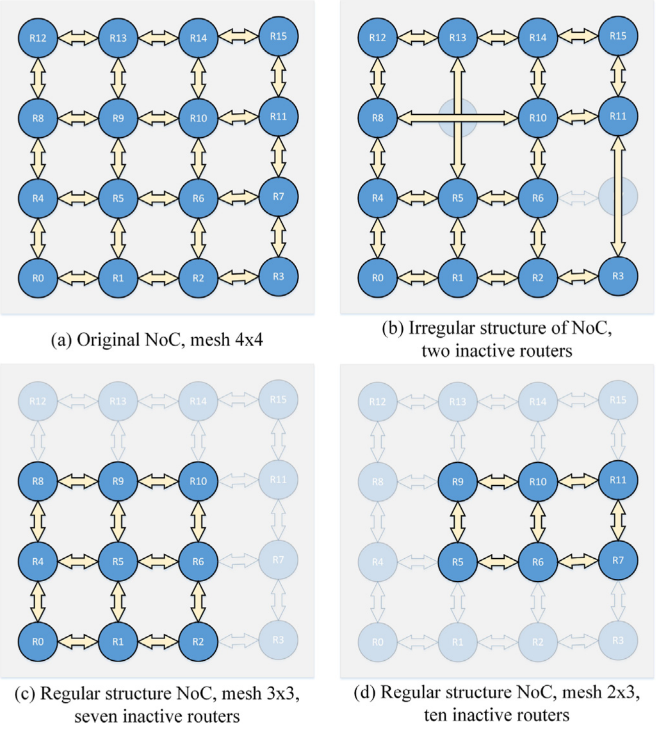
Those changes allowed the network to be completely reconfigurable based on the needed runtime requirements. In Fig. 3(a), the 4 × 4 mesh network is completely functioning and could be reconfigured at runtime to any desired network structure like the 3 × 3 mesh network in Fig. 3(c) by disabling the 7 routers on the edge, or the 2 × 3 mesh network like in Fig. 3(d) by disabling the 10 routers on the edge, or even any irregular or non-uniform structure like in Fig. 3(b).

The original routing algorithm implemented inside CONNECT Mesh NoCs was based on aligning the horizontal destination first before the vertical destination. The modified routing algorithm in this work handle the possibility that the horizontal path might be reconfigured and directs the packets into the alternative direction (the vertical direction in this case) till the packet reaches its destination.

The newly introduced changes in the CONNECT NoC are considered the most valuable contribution in this work. Enabling the reconfiguration of part of the network during runtime leads to power and area saving during the low traffic periods. The global interface input provides a user interface which decides the desired network configuration. The added switches around each node maintain the connectivity of each node under any possibility. Moreover, the modified routing technique adds the flexibility to a static NoC hardware designed for FPGA without degrading the performance.

4. Reconfiguration tool and environment structure

The modified CONNECT RTL is tested against an environment used in Ref. [16] which has the structure shown in Fig. 4. Every router inside the network is attached to a packet generation module and a credit handler module. The packet generator essentially provides every router with input packets going to a completely randomized destination and virtual channel. The packet generator takes into consideration the required traffic load, the currently active routers, and the free virtual channels. The credit handler module monitors the incoming and outcoming packets into and from each virtual channel. This credit is considered as an indicator to the storage available inside every channel buffer. The environment in Fig. 4 is built to ensure that the NoC is fully functional and the developed RTL is behaving correctly.



(a) Original NoC, mesh 4x4

(b) Irregular structure of NoC, two inactive routers

(c) Regular structure NoC, mesh 3x3, seven inactive routers

(d) Regular structure NoC, mesh 2x3, ten inactive routers

Fig. 3. CONNEC NoC reconfigurable samples at runtime.

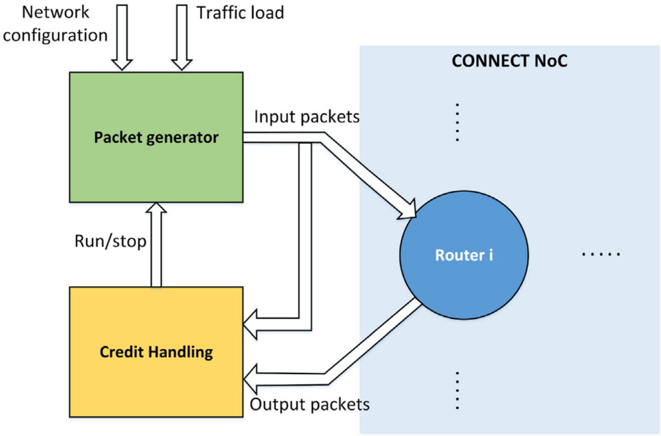


Fig. 4. CONNECT NoC reconfigurable environment components.

Generally, the aim of all the described RTL updates is to build a reconfiguration tool providing the endorsed network structure according to the intended benchmarks to be used at runtime. The referred reconfiguration tool is developed as shown in Fig. 5. The reconfiguration requires the traffic density and performance requirement corresponding to each input user application. The output of the reconfiguration tool is the endorsed network structure for every application to be switched to while reconfiguration is taking place. Table 1 represents a use case showing how the reconfiguration tool recommends different networks which could lead into area optimization when switching from an application to another.

The reconfiguration tool is developed using Python as a scripting language. The criteria are based on searching for the best fitting network within the evaluation results. The required throughput and the expected traffic load are the two inputs to the reconfiguration tool from

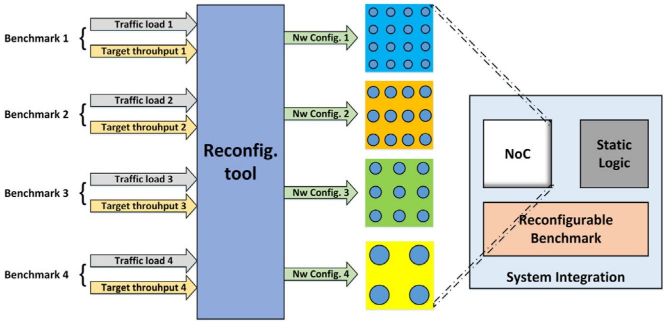


Fig. 5. Reconfiguration tool use case for user benchmarks.

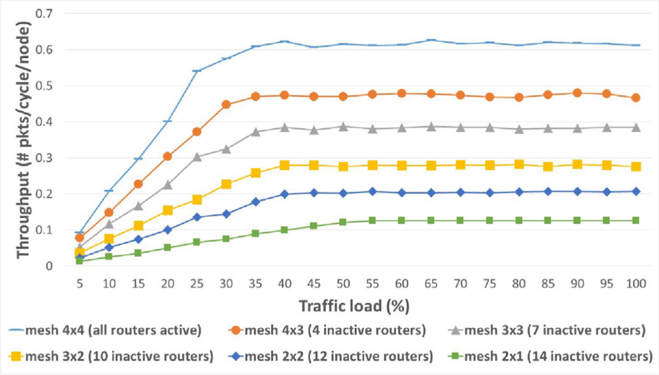


Fig. 6. Throughput of regular networks with a buffer depth of 4.

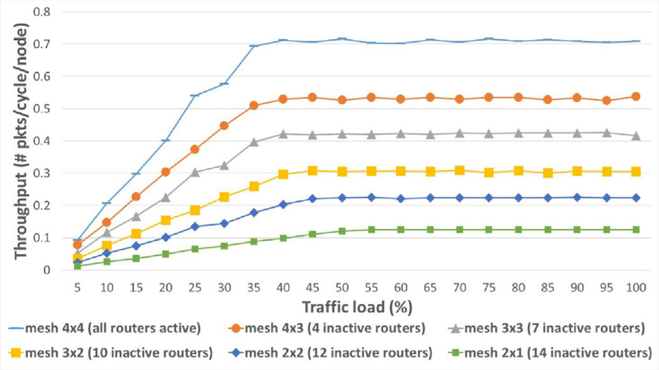


Fig. 7. Throughput of all regular networks with a buffer depth of 64.

which the reconfiguration tool can list all the fitting networks in an output file.

The reconfiguration tool interface with the user is available in the batch mode and the GUI mode as well. In Batch mode, the python script "search.py" is called through the shell passing the expected traffic and required throughput. The output of the reconfiguration tool in both modes is a list of all network configurations that are fitting the requirements.

5. Results and discussion

The previously specified environment of CONNECT is generalized with the ability to examine various network configurations at any traffic load ranging from 5% to 100%. The environment can deal with different virtual channels ranging from 2 to 8, different buffer depths ranging from 4 to 64. Moreover, the DPR evaluation is applied to all the possible networks nodes starting from a complete network functioning (4 × 4 mesh) up to reconfiguring 14 nodes and preserving two nodes only functioning (2 × 1 mesh).

Network evaluation is based on the output throughput as a performance metric and plays an important role in the decision of the optimal network configuration. The output throughput is examined by injecting the network with input packets based on the desired traffic load. Then, calculating the number of output packets per node per cycle.

When some routers are reconfigured and removed during runtime, the network becomes smaller in size. This leads into area saving at the cost of performance degradation because the neighbor routers are supposed to manage higher traffic load. The network reconfiguration can vary from a fully functioning network with the best possible performance to a smaller network with the worst performance. Also, the benchmark target performance plays an important role in the decision of the best network configuration which meets this requirement. The selection criteria is prioritizing the lowest area size which corresponds to the smallest possible network.

The following subsections discuss the impact of different configurations on the performance and provides an in-depth view on how each configuration works individually on boosting the network overall performance.

- Network topology/size DPR

From the early evaluation results, with the same number of routers, the regular forms of the network give better results than the irregular forms. For the 3 × 3 mesh network, for instance, it contains 9 routers and gives better results than any irregular network with the same number of routers (9 routers). So, the results shown mainly correspond to regular network forms: 2 × 1 mesh, 2 × 2 mesh, 3 × 2 mesh, 3 × 3 mesh, 4 × 3 mesh, and 4 × 4 mesh.

Fig. 6 presents the specified networks performance under a traffic load ranging from 5% to 100% and with a buffer depth of 4. Obviously, using reconfiguration in reducing the network size helps in getting lower throughput. The reason behind the lower throughput rates is that the rest of the active nodes after reconfiguration are forced to handle a higher load. This higher load appears after compensating for the absence of the reconfigured inactive nodes.

In Fig. 7, the buffer depth of every router is expanded to 64 instead of 4 and this affects the performance of all the network configurations positively. However, still the same effect of the network topology/size is dominant.

In general, shrinking the network size by means of DPR into a smaller topology results in degrading the network performance. However, DPR leads to saving some area for other logic to be used. This could be beneficial with applications that do not require high performance at the moment and could switch into smaller network topology.

- Buffer Depth DPR

Table 1
Reconfiguration tool Output for different User Benchmarks.

Benchmark	Target Throughput	Expected Traffic	Network Config.	Virtual Ch.	Buffer Depth
Application 1	0.68	80%	4 × 4 mesh	2	64
Application 2	0.53	55%	4 × 3 mesh	2	32
Application 3	0.4	40%	3 × 3 mesh	2	16
Application 4	0.1	20%	2 × 2 mesh	2	4

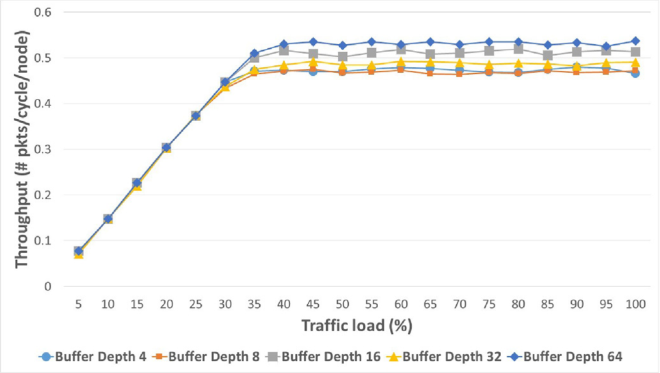


Fig. 8. Throughput of a 4 × 3 mesh network with different buffer depths.

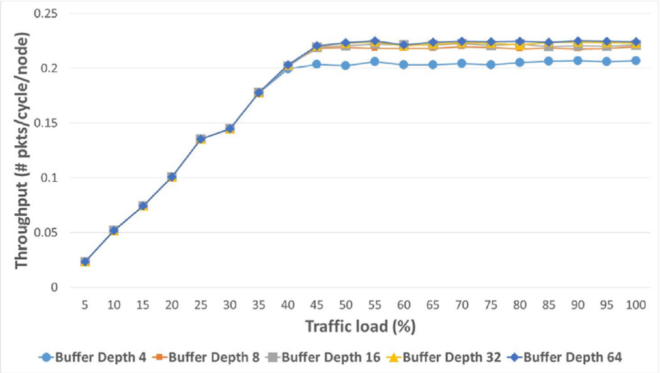


Fig. 9. Throughput of a 2 × 2 mesh network with different buffer depths.

Applying DPR into the buffer depth as a network parameter impacts the network performance differently. The performance shown here is for the buffer depth of each router inside the 4 × 4 mesh specified network with credit-based flow control, 2 virtual channels, a buffer depth of 4, and a traffic density ranging from 5% to 100%.

Fig. 8 presents the impact of the buffer depth as a parameter on a 4 × 3 mesh network. The 4 × 3 mesh network is achieved by reconfiguring 4 nodes out of the 4 × 4 mesh original network. At low traffic loads (lower than 30%), the different buffer depths have the same effect on the performance because the network is not fully loaded. At high traffic loads, a large buffer can handle more packets which helps in improving the network performance.

In general, expanding the buffer depth impacts positively the network configurations performance. This is due to the extended capability of each node to receive and handle more packets at the cost of more area overhead.

In Fig. 9, the low traffic impact is expanded to reach 40% traffic load instead of 30%. On the other hand, large buffer depths have an insignificant effect on the performance of the noticeable small networks (2 × 2 mesh – 12 reconfigured routers). This is because of the noticeable low latency which requires small storage (buffer depth) inside every router.

However, the buffer depth effect in Fig. 10 becomes unnoticeable with shrinking the network size (2 × 1 mesh – 14 routers reconfigured). This is because the major part of the buffer depth is not utilized efficiently especially with small network sizes. Generally, packets inside small networks suffer lower latency than inside large networks as the packets do not consume much time till reaching the destination.

- Virtual Channel DPR

Applying DPR into the virtual channel as a network parameter impacts the network performance in a different way than the buffer depth. The performance shown here is for virtual channels inside the 4 × 4 mesh network with peek-based and credit-based flow controls, different buffer depths, and at a traffic density ranging from 5% to 100%.

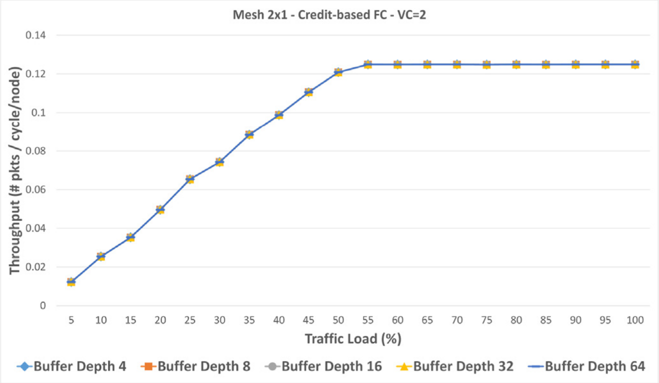


Fig. 10. Throughput of a 2 × 1 mesh network with different buffer depths.

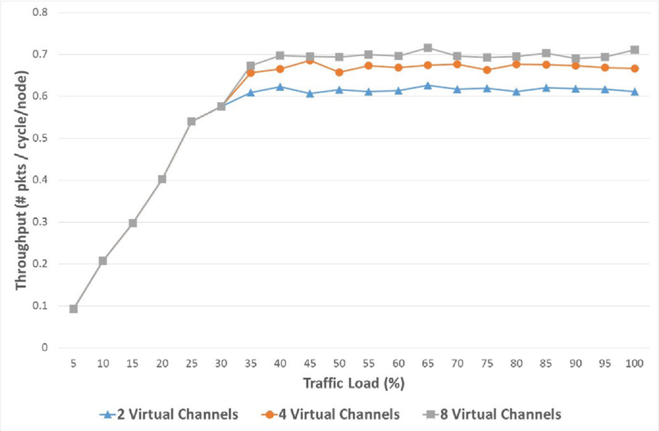


Fig. 11. Throughput of a 4 × 4 mesh network with a buffer depth of 4 and different virtual channels.

In general, increasing the virtual channels enhances all the network configurations performance. This is due to creating new routing paths in parallel with the original network paths. Accordingly, expanding the network capability to receive and handle more packets at the cost of more area overhead.

Fig. 11 presents the effect of the virtual channel as a parameter on the performance of the reconfigurable 4 × 4 mesh network. The existence of additional virtual channels presents alternative paths to packets resulting in throughput enhancement. However, the 3 × 3 mesh network throughput in Fig. 12 is not affected by the virtual channel because of the large buffer depth which prevents the utilization of the additional virtual channels.

The virtual channel impact becomes unnoticeable with the shrinking of the network size to a 2 × 1 mesh network. This is because of the low latency of small networks which lowers the probability of congestion even with high injection rates.

It is noticeable in all the results that the positive impact of virtual channel DPR is valuable only with relatively large networks with small buffer depths. Investing in virtual channel DPR in small networks or large buffer depths leads to a waste of area.

- Buffer Depth vs Virtual Channel

Applying DPR into the buffer depth or the virtual channel impacts the network performance in nearly the same way. The performance

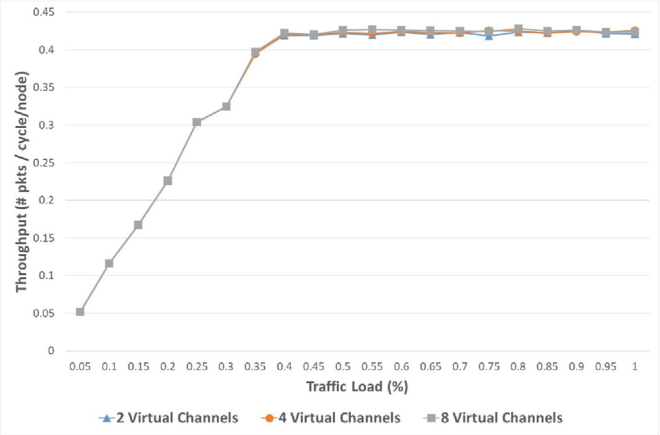


Fig. 12. Throughput of a 3 × 3 mesh network with a buffer depth of 32 and different virtual channels.

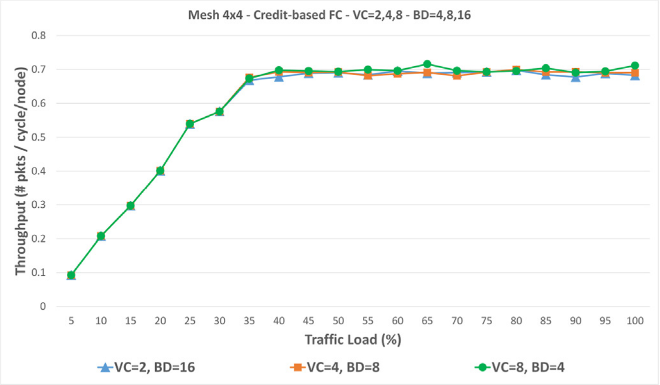


Fig. 13. Throughput of a 4 × 4 mesh network with buffer depths of 4, 8, and 16 and virtual channels of 2, 4, and 8 respectively.

shown here is for buffer depths vs virtual channels inside a 4 × 4 mesh network using credit-based flow control, at a traffic load ranging from 5% to 100%.

It is obvious in Fig. 13 that using virtual channel/buffer depth gives nearly the same effect with the same network configurations. This is proven using 4-virtual channels and 8 buffer depth versus 8-virtual channels and 4 buffer depth. Note that the condition of a small networks still applies.

- Flow Control DPR

The network flow control mechanism defines the feedback technique while communicating with neighbor routers. The credit-based flow control provides detailed feedback when space is emptied in each virtual channel. On the other hand, the peek-based flow control just provides a busy signal indicating the availability or not of each virtual channel. Accordingly, the peek-based flow control is much simpler and allows maximizing the use of network resources but credit-based flow control provides more intelligence into the network in the case of choosing different routing paths.

Fig. 14 corresponds to a 4 × 4 mesh network with a buffer depth of 4 and 2 virtual channels. It shows that the impact of the Flow control mechanism is very slight. It is only noticeable with small buffer depths that give some small advantage to the peek-based flow control mechanism over the credit-based flow control mechanism. Note that the condition of a small networks still applies.

- Area Evaluation

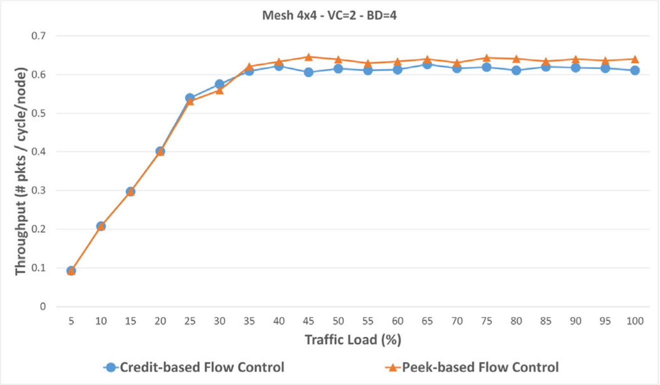


Fig. 14. Throughput of a 4 × 4 mesh network with a buffer depth of 4, a virtual channel of 2, and different flow-control mechanisms.

Table 2
Estimated Virtex 5 Xilinx FPGA resource area [18].

Resource	Equivalent number of gates	Silicon Area in mm2
Register	7	0.000341
LUT	24	0.001171
IO	100	0.004882
BRAM	–	0.025436

As a different performance metric, the area score of different networks is considered the most important factor in deciding the most suitable network with every benchmark. This metric is mainly used by the reconfiguration tool in order not to let the throughput mislead the network selection criteria. In this subsection, an example highlights how the area could be a valuable gain by using reconfigurable NoCs over static NoCs.

The static CONNECT and the reconfigurable CONNECT are synthesized on Virtex-5 xc5vlx110tff1136-1 FPGA. At the synthesis level, the runtime reconfigurable 4 × 4 mesh CONNECT achieves more area score. The area overhead is because of the introduced switches and adapted routing.

For the area resources in Virtex 5 Xilinx FPGAs, Table 2 shows estimates for the area score for each Virtex 5 resource with respect to the comparable number of gates and the absolute area in millimeters square. This area score is used by the reconfiguration tool as a metric for the best fitting Network. This criterion is used as the main aim of the reconfigurable NoC to allow area saving when switching between different networks.

The reconfiguration tool needs a single area score in order to compare the different networks with respect to area. The minimum area score is used to decide the minimum area network configuration. The area estimations in Table 2 are used to merge the different FPGA resources into a single area score. The silicon area scores presented in Ref. [18] are used as a reference in order to be able to calculate an overall area score by combining the different FPGA area resources. This evaluated area score is used by the reconfiguration tool to compare multiple configuration area by comparing the overall area score only.

- Reconfiguration Time

The reconfiguration time for the static CONNECT is dependent on configuring all the frames reserved by the whole NoC. On the other hand, the dynamic CONNECT reconfiguration time can be reduced by the factor of the difference between the two NoCs being configured. However, this work does not address this aspect in details. This work focuses on the impact of reconfigurability on the NoC performance not the other way.

Table 3
Case study benchmarks with Static NoC and Reconfigurable NoC approaches.

Benchmark	Static NoC approach	Reconfigurable NoC approach
Application A	4 × 4 mesh	4 × 4 mesh
Application B	4 × 4 mesh	4 × 3 mesh
Application C	4 × 4 mesh	3 × 3 mesh
Application D	4 × 4 mesh	3 × 2 mesh
Application E	4 × 4 mesh	2 × 2 mesh

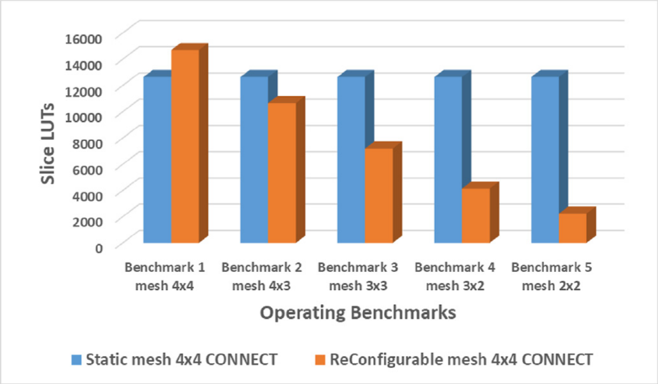


Fig. 15. Slice LUTs of Reconfigurable 4 × 4 mesh CONNECT vs Static 4 × 4 mesh CONNECT using Virtex 5 xc5vlx110tff1136-1.

6. Case study

This is a study highlighting the effect of reconfigurable NoC. The five benchmarks listed in Table 3 are designed using the Static NoC approach and the Reconfigurable NoC approach. These five benchmarks require uniform traffic loads among mesh-shaped NoCs.

This study uses different virtual application covering the minimum and maximum performance requirements. The benchmarks are selected to highlight the maximum advantage from using the reconfigurable network over the static one.

The various reconfigurable applications force selecting the static NoC which satisfies the worst case requirements and consumes the largest area all the time. Nevertheless, the reconfigurable NoC allows switching to the best fit structure satisfying the application requirements without consuming unneeded area. This unneeded area is eliminated when the current application performance requirements are relaxed.

Fig. 15 represents a usage model example of two NoCs switching between the set of listed benchmarks. These benchmarks are varying from a 4 × 4 mesh network into a 2 × 2 mesh network. The first NoC is a static NoC which shows a constant area resource usage even with moving from an application to another.

However, the second NoC which is a reconfigurable NoC shows a variable area with each new reconfiguration. The switching and hence usage area are according to the current application used. Generally, the reconfigurability of NoC is very powerful when the network usage model does not require a high throughput all the time.

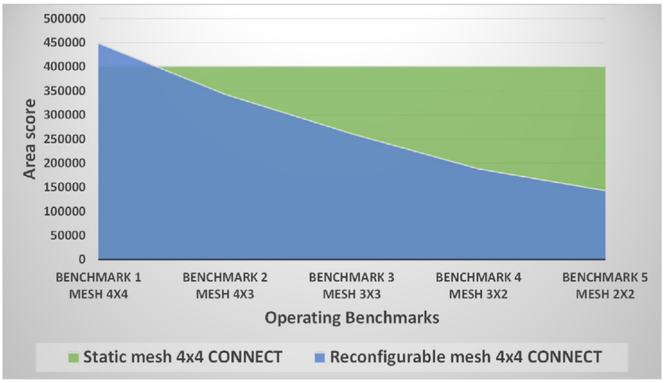


Fig. 16. Area scores of Reconfigurable 4 × 4 mesh CONNECT vs Static 4 × 4 mesh CONNECT using Virtex 5 xc5vlx110tff1136-1.

The area resources of all the static and reconfigurable networks are listed in Table 4. The area reduction is noticed with the reduction of the network size when switching between different benchmarks.

Assuming that the five benchmarks are operating with equal times, when using reconfigurable NoCs including the added RTL for reconfiguration, the overall area saving in this case study is as follows:

- Saving in Slice Registers: 44.67%
- Saving in Slice LUTs: 38.4%
- Saving in LUT-FF pairs: 44%
- Saving in IOs: 18.6%
- Saving in BUFG/BUFGCTRL: 40%

Fig. 16 shows the overall area score of the five benchmarks using the two approaches, the static NoC approach and the reconfigurable NoC approach. The area score is calculated using the Virtex 5 resources data listed in Table 2. In this case study, the reconfigurable NoC approach achieves average saving of 30.89% over the static NoC one.

Fig. 17 represents the power scores of the same case study synthesized at clock frequency of 20 MHz, The reconfigurable NoC achieves an average power saving of 38.26% over the static NoC. The different designs normally achieve clock frequency more than 100 MHz. However, some other design combinations suffer some critical paths during compilation that need to be divided. Therefore, all the designs used are compiled at 20 MHz in order to unify the results with the minimum frequency guaranteed among all designs during compilation for fair comparison.

These results are assuming that the time multiplexing will be shared equally among all the benchmarks which allows the calculations of average area and power savings. Furthermore, the benchmarks selection in this case study is highlighting the best case of using the reconfigurable NoCs through a variety of the different applications requirements.

On the other hand, it is obvious that moving from a 4 × 4 network to a 2 × 2 network would be at the expense of throughput as the throughput would degrade by nearly 37% on average. However,

Table 4
Area resources for different networks corresponding to every benchmark.

Benchmark	VC	BD	Slice Regs.	Slice LUTs	LUT-FF pairs	IOs	BUFG
4 × 4 mesh (Static)	2	4	3758	12644	2287	914	2
4 × 4 mesh (Reconfig.)	2	4	3758	14696	2276	930	2
4 × 3 mesh (Reconfig.)	2	4	2728	10660	1668	818	1
3 × 3 mesh (Reconfig.)	2	4	1976	7202	1231	733	1
3 × 2 mesh (Reconfig.)	2	4	1207	4145	766	648	1
2 × 2 mesh (Reconfig.)	2	4	726	2238	459	590	1

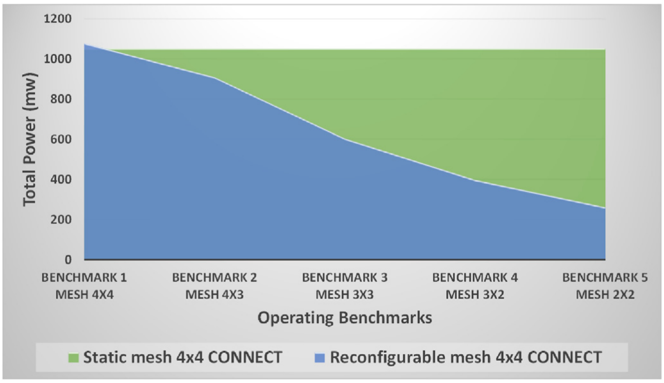


Fig. 17. Total power of Reconfigurable 4 × 4 mesh CONNECT vs Static 4 × 4 mesh CONNECT using Virtex 5 xc5vlx110tff1136-1.

this low throughput can be acceptable by design during the low traffic periods in the network.

This case study highlights the best usage model of the reconfigurable NoCs. The usage model is based on the reconfiguration by the user to the network designed previously. The planned applications require different networks to be reconfigured during runtime. The user and the network designer need to consider the switching time and its effect of the overall task completion time.

7. Design recommendations

From the previous evaluations, some recommendations could help when planning to use NoCs in your design and whether you are going to use static or reconfigurable NoCs. These design recommendations are listed below:

- Reconfigurable NoCs are preferred when there are multiple benchmarks going to run with different requirements (traffic load, throughput). The reconfigurable NoCs will give the design the required adaptability to runtime requirements with area and power gains.
- Static NoCs are preferred when only a single benchmark is running. Additionally, it is suitable for multiple benchmarks with the same performance requirements. The adaptability here has no meaning as runtime requirements do not need a lot of variations.
- The main gain behind applying PDR to the NoC topology is the area and power saving. Removing a set of routers and changing the network topology during runtime degrades the performance while saving the area of the reconfigured nodes. This area saved is going to be used by the rest of the design.
- Applying DPR to the network buffer depth is beneficial with high traffic loads and relatively large networks. It has its minimal effect with small networks or low traffic loads.
- Applying DPR to the network virtual channel has its highest effect with high traffic loads and relatively large networks plus a small buffer depth. Large buffer depth networks could prevent making the most of the available virtual channels.
- Choosing a buffer depth DPR or a virtual channel DPR depends mainly on the usage model of the design of the specified benchmarks. The virtual channel is preferred when planning to use parallel loading and packet injection. However, the buffer depth is preferred when the internal router storage is more important than routing resources.
- Flow control mechanism DPR could help when fine tuning the network parameters during DPR selection. The peek flow control is much simpler in implementation which means less area and power. However, the credit flow control gives the router a more detailed information about the traffic going through the neighbors and gives a smarter insight with possible routing paths.

8. Conclusion and future work

The main value added by this work is the reconfigurability of the CONNECT NoC for FPGA applications and how this reconfigurability could result in reducing area and power. This work shows a study on the reconfiguration effect on the NoC performance. Moreover, it focuses on how scaling down the network size during runtime can result in area and power saving. This saving is at the expense of degrading the performance when the high performance is not a priority. In general, a low performance can fit with some benchmarks under certain traffic loads. Moreover, other network configuration parameters are studied and their impact on the network performance is analyzed.

Since a large network requires holding packets for a longer duration than a small network, the buffer depth contributes to improving the performance with large network sizes. The Virtual channel acts also as a booster for the performance especially with large networks and small buffer depths. The flow control mechanism impact can also be noticeable with small buffer depths. Finally, the area metric plays a very important role in the best fit network selection. The area score is considered the main advantage that can be gained from applying DPR into NoC.

The reconfigurable 4 × 4 mesh CONNECT had a detailed analysis of the throughput and area as performance metrics of the network. This analysis can be extended in the future to include the following:

- Comparing the performance against the work done in Refs. [11,12].
- Evaluating larger networks like 6 × 6 mesh and 9 × 9 mesh networks for providing more configuration options to the user and more detailed analysis on large scale NoCs.
- Evaluating other network topologies like Ring and Star networks and providing a detailed analysis and a comparison between them.
- Proposing a technique for estimating traffic load for every user benchmark instead of expecting it as an input.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was funded by Mentor, a Siemens business, and ONE Lab at Cairo University and Zewail City of Science and Technology

References

[1] M.S. Gaur, V. Laxmi, M. Zwolinski, M. Kumar, N. Gupta, Ashish, Network-on-chip: current issues and challenges, in: VLSI Design and Test (VDAT), 2015 19th International Symposium, June 2015, pp. 1–3.

[2] P. Lysaght, B. Blodget, J. Mason, J. Young, B. Bridgeford, Enhanced architectures, design methodologies and CAD tools for dynamic reconfiguration on XILINX FPGAS, in: Proceedings of the 16th International Conference on Field Programmable Logic and Applications (FPL06), Madrid, Spain, August 2006.

[3] S.G. Khawaja, M.U. Akram, S.A. Khan, A. Shaukat, S. Rehman, Network-on-Chip based MPSoC architecture for k-mean clustering algorithm, in: Microprocessors and Microsystems, vol. 46, 2016, pp. 1–10.

[4] A. Ahmadinia, C. Bobda, J. Ding, M. Majer, J. Teich, S.P. Fekete, J.C. van der Veen, A practical approach for circuit routing on dynamic reconfigurable devices, in: Proceedings of RSP, 2005, pp. 84–90.

[5] T. Pionteck, R. Koch, C. Albrecht, Applying partial reconfiguration to networks-on-chips, in: Proc. Int. Conf. Field Programmable Logic and Applications (FPL), 2006, pp. 1–6.

[6] C. Bobda, A. Ahmadinia, Dynamic interconnection of reconfigurable modules on reconfigurable devices, in: Design & Test of Computers, vol. 22, 2005, pp. 443–451. no. 5.

[7] C. Bobda, A. Ahmadinia, M. Majer, J. Teich, S. Fekete, J. van der Veen, DyNoC: a dynamic infrastructure for communication in dynamically reconfigurable devices, in: Proc. Int. Conf. Field Program. Logic Appl., Aug. 2005, pp. 153–158.

[8] T. Pionteck, C. Albrecht, R. Koch, A dynamically reconfigurable PacketSwitched network-on-chip, in: Proceeding of the Conference on Design, Automation and Test in Europe, DATE'06, vol. 1, March 2006, pp. 8–9.

[9] M. Modarressi, H. Sarbazi-Azad, A. Tavakkol, An efficient dynamically reconfigurable on-chip network architecture, in: Proc. of the 47th Design Automation Conference (DAC 2010), 2010, pp. 310–313.

[10] M.B. Stuart, M.B. Stensgaard, J. Sparsø, The ReNoC reconfigurable network-on-chip: architecture, configuration algorithms, and evaluation, in: ACM Trans. Embed. Comput. Syst., vol. 10, 2011, p. 45. no. (4).

[11] M. Modarressi, A. Tavakkol, H. Sarbazi-Azad, Application-aware topology reconfiguration for on-chip networks, in: IEEE Transactions on VLSI Systems, vol. 19, Nov. 2011, pp. 2010–2022. no. 11.

[12] J. Huang, X. Xu, N. Wang, S. Chen, Reconfigurable topology synthesis for application-specific NoC on partially dynamically reconfigurable systems, in: Integration-the VLSI Journal, vol. 65, March 2019, pp. 331–343.

[13] M. Chang, J. Cong, A. Kaplan, C. Liu, M. Naik, J. Premkumar, G. Reinman, E. Socher, S.-W. Tam, Power reduction of CMP communication networks via RF-interconnects, in: Proceedings of the 41st Annual IEEE/ACM International Symposium on Microarchitecture, 2008, pp. 376–387.

[14] M.K. Papamichael, J.C. Hoe, The CONNECT network-on-chip generator, in: Computer, vol. 48, Dec. 2015, pp. 72–79. no. 12.

[15] http://users.ece.cmu.edu/~mpapamic/connect/.

[16] K.A. Helal, S. Attia, T. Ismail, H. Mostafa, Comparative review of NoCs in the context of ASICs and FPGAs, in: International Symposium on Circuits and Systems (ISCAS), 2015, pp. 1866–1869.

[17] R. Ahmed, H. Mostafa, A.H. Khalil, Impact of dynamic partial reconfiguration on CONNECT Network-on-Chip for FPGAs, in: 2018 13th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS), Taormina, 2018, pp. 1–5.

[18] F. Arnaud, et al., A Functional 0.69 Embedded 6T-SRAM bit cell for 65nm CMOS platform, in: The Digest of Technical Papers of the Symposium on VLSI Technology, 2003, pp. 65–66.