An Asynchronous, Low Power and Secure Framework for Network-On-Chips

M. Mirza-Aghatabar[*], A. Sadeghi[†]

[*]Sharif University of Technology, Tehran, Iran
[†]Maleke Ashtar University of Technology, Tehran, Iran

Summary
Network-on-Chip (NoC) is an approach to handle huge number of transistors by virtue of technology scaling to lower than 50nm. The issue of security has been always controversial to many designers. Among the attackers, one of the most important of them is power attacker which uses statistical techniques to determine the secret keys by observing power consumption. The power consumption spurs during runtime (1 to 0 and 0 to 1 switching) is based on charging and discharging of capacitors. In this paper we will introduce an asynchronous framework for NoC which is based on QDI, 4-Phase handshake signaling. We will show that our framework has lower power consumption compared to traditional synchronous router due to its asynchronous nature.
We will also show that our framework is more secure against the power attackers due to its dual rail encoding style. We will synthesis our framework with *Persia* synthesis tool and will compare it with synchronous router from power consumption point of view under different traffic models, one and three number of virtual channels and AFBAR routing algorithm.

Key words:
Network-on-Chip, Router, Asynchronous, Low Power, Security

1. Introduction

The number of transistors has increased beyond billions in a technology less than 50nm [1]. System-on-Chip (SoC) and Network-on-Chip (NoC) are two main implementation approaches that are used to manage these enormous numbers of transistors. SoCs have some disadvantages such as: (1) non-reusability and (2) low scalability, (3) complex design, and (4) long time to market [2].

Traditionally, communication between processing elements was based on buses; however, for large multiprocessor SoCs with many processing elements, it is expected that the bus will become a bottleneck from a performance, scalability and power dissipation point of views. Therefore, the idea of networks on chip has evolved. NoCs enable integration of considerable number of computational and storage blocks on a single chip. They are structured, reusable, scalable, and have high performance.

Due to importance of NoCs and their usages, we should consider the security issue of these circuits. There are many attackers that try to infect the systems and determine their secret keys. One of the most common ways is power consumption analysis. These attackers use statistical techniques to determine the secret keys by observing power consumption [3]. As we know, each transition in a circuit causes charging or discharging of a capacitor which leads to power consumption. These transitions are inevitable, and different data coding just can increase the complexity of determining the secret keys by attackers.

Asynchronous circuits design with dual rail encoding is the best solution in order to remove the power consumption spurs. We have two bits for each value in dual rail encoding, and when there is a transition between 0 and 1 we have two capacitors charging and discharging concurrently. Therefore, there would be no power consumption spur.

In recent years, a number of methods based on different timing models have been proposed to develop practical asynchronous circuits [5] such as delay insensitive. Regarding the delay insensitive models, synchronization between different sections is performed by generating and detecting request and acknowledgement signals.

Advantage of asynchronous circuits are as follows: eliminating the clock skew problem, modularity, lower power consumption, applying average delay instead of worst case delay, quick adaptation to newer technologies, and less vulnerability to changes in voltage and other environmental parameter such as temperature. As a result, an asynchronous router has the potential to consume less power, to expose better performance, be more secure and more flexible to adapt to voltage and other parameter changes compared to a synchronous router.

In contrast to the mentioned advantages, there are some drawbacks regarding asynchronous circuits such as complex design procedures, and larger number of transistors. As a result, having an automatic asynchronous design tool is extremely helpful in popularizing asynchronous design methods. We will use *Persia*, an asynchronous design synthesis tool based on the QDI[1]

[1] Quasi Delay Insensitive

Manuscript received July 5, 2008.
Manuscript revised July 20, 2008.

[6][7] timing model, and it can support GALS[2] design as well.

We claim that an asynchronous design of a traditional router can decline the power consumption and also improve the security of NoCs against the power analyzer attackers. Hence, in this paper we will introduce a new framework for routers and NoC with asynchronous design in order to reduce the power consumption and increase their security. We will evaluate our framework under different traffic models such as Uniform, Local and Hotspot, AFBAR as the best routing algorithm, and different number of virtual channels. Our experimental results show the efficiency of our proposed framework.

In section 2, we take a look at related work. Section 3 introduces the *Persia* synthesis tool. Section 4 presents the design of an asynchronous router. In section 5 we describe our motivation and in section 6 we show our experimental results. Finally, in section 7 we will conclude our work.

2. Related Work

The security issue of NoCs is completely new and there is lack of variety in the papers related to security of NoC. For the first time a framework for security on NoC at both the network level (or transport layer) and at the core level (or application layer) is proposed in [8]. To protect encrypted private and public keys, the authors included a key-keeper core and security wrapper to each IP core at the network layer. At the core level (application layer) the security framework is illustrated with software modification for resistance against power attacks with extremely low overheads in energy.

Another work is [9] which addresses a new kind of security vulnerable spots introduced by Network-on-chip (NoC) use in System-on-Chip (SoC) design. This study is based on the experience of a CAD framework for NoC design and proposes a classification of weaknesses with regard to usual routing and interface techniques. The authors of [9] proposed a design strategies and a new path routing technique in order to enforce the security.

None of this works considered the power attackers at hardware level. The first work considered the power attackers with software modifications. However, this assumption is not very effective and the system is yet exposed to these attackers. In this paper we introduce a new framework which is independent of inputs and completely secure against of power attackers. Our framework is based on asynchronous design and has lower power consumption compared to traditional synchronous router.

3. Persia Synthesis Tool

Persia is an asynchronous synthesis toolset developed for automatic synthesis of QDI[3] asynchronous circuits with adequate support for GALS[4] [12] systems. The structure of Persia is based on the design flow shown in Figure 1 which can be considered as the following four individual portions: QDI synthesis, GALS synthesis, layout synthesis, and simulation at various levels. QDI and GALS synthesis flows are joint together in the layout stage. The simulation flow is intended to verify the correctness of the synthesized circuit in all levels of abstraction.

CSP is a well-known language for description of concurrent systems which is accepted as a good description language for asynchronous systems. A Circuit in CSP is described as the composition of distinct processes that run in parallel and communicate with each other on channels by message passing. Persia uses Verilog-CSP[5]0 [13], an extension of the standard Verilog which supports asynchronous communication as the hardware description language for all levels of abstractions except the net list which uses standard Verilog. This way the Verilog is powered by some READ and WRITE PLI[6] macros to emulate CSP language communication actions on the channels. The input of Persia is a Verilog description of a circuit that includes READ and WRITE macros for sending and receiving data via communication channels. This description will be converted to a netlist of standard-cell elements through several steps of QDI synthesis flow. For simpler synthesis first arithmetic operations are extracted from the code and the major steps of synthesis only works on the codes without any arithmetic operations. This is done by the AFE which also replaces the arithmetic functions by standard library modules. The two major steps in Persia synthesis are Decomposition and TSYN. For more information you can see [4].

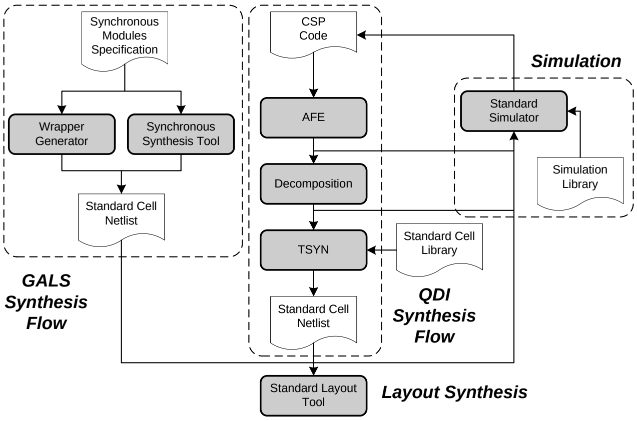


Figure 1: QDI and GALS design flow

4. Asynchronous Router for NoC

We implemented our router for a 2-dimentional topology; therefore, each router has four dual ports to

[2] Globally Synchronous Locally Asynchronous

[3] Quasi Delay Insensitive

[4] Globally-Asynchronous Locally-Synchronous

[5] Communicating Sequential Processes

[6] Programming Language Interface

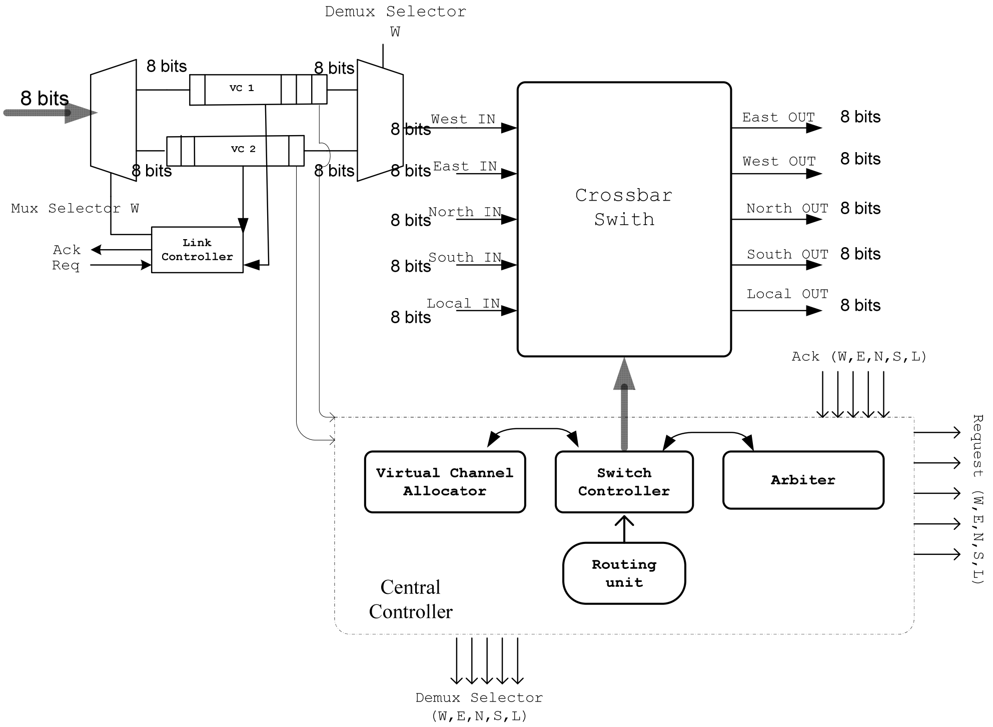


Figure 2: An asynchronous Router with 2 virtual channels

communicate with its neighbors. It has also one local port; we name the router’s ports as: East, West, South, North and Local. Local port is used for injection and ejection of packets and this port will be used in source and destination nodes. Other ports are used to transmit the intermediate packets.

In asynchronous circuits we do not have Clock, and the data communication will be done by handshake signaling. Of the many transactional protocols widely used in electronics to transfer data, the four-phase handshake is about the most common. The protocol provides rate adaptation, in that it has an idle state that it starts in and resets in when the sender is not ready to send, and in that it has backpressure or flow-control that prevents the sender sending further data when the receiver has no accommodation. The protocol is defined at the net level using a request and an acknowledge signal and one or more data signals that carry the data: {req, ack}.

Figure 2 shows the high level design of an asynchronous router. This is clear that we do not have any Clk signal and each router will communicate by its neighbors with handshake signaling. Each router needs a switching and routing mechanism to route packets in the network. There are many switching mechanisms such as packet switching, circuit switching, virtual cut through, and wormhole (WH) switching [2]. Among them, WH switching is very popular due to its pipeline nature, also it needs least buffer size which leads to lower power consumption and less hardware complexity.

Another important part of the router is its routing function. The routing function uses the address of source, destination and current node to route the packet into its destination. There are many routing functions. One way to characterize routing algorithms is based on their amount of adaptivity. In this way, deterministic algorithms determine the path based only on the source and destination node addresses, which leads to minimum adaptivity results in fast and simple design of routers [18][19]. Adaptive routing algorithms on the other hand, can react to network conditions as they allow packets to be routed along alternate paths [19].

Another way to characterize routing algorithms is based on their strategy to handle deadlocks: deadlock avoidance (Deterministic and Adaptive) and deadlock recovery routing algorithms.There are some deadlock recovery routing algorithms such as: CR [21], SW_TFAR [14], AFBAR [10], Disha [17][16][15][14].

To implement a routing function for an asynchronous router we use Verilog-CSP language. To illustrate, Figure 3 shows the asynchronous implementation of XY routing algorithm with CSP.

```
-- =========== Physical Channel Selection
'READ (HEADER_FLIT, Receiver)
CurX := To_Integer(CurNode Mod ColNo); --x
CurY := To_Integer(CurNode / ColNo); --y
DestX := To_Integer(Receiver Mod ColNo); --x
DestY := To_Integer(Receiver / ColNo); --y
XDiff := DestX - CurX;
YDiff := DestY - CurY;
If (XDiff < 0) Then  PhyChAssigned := 0; -- "000"
```

```
ElsIf (XDiff > 0) Then PhyChAssigned := 2; -- "001"
ElsIf (YDiff < 0) Then PhyChAssigned := 1; -- "010"
ElsIf (YDiff > 0) Then PhyChAssigned := 3; -- "011"
Else                   PhyChAssigned := 4; -- "100"
End If;
'WRITE (OutPackPhCh , PhyChAssigned)
-- ============ Virtual Channel Selection
IsInpChAssigned := '0';
OutPackViCh := -1;
For j In 0 To ViCh-1 Loop --loop vi outp ch
   Ind := PhyChAssigned*ViCh+j;
   If(IsOutpChBusy(Ind)='0')And
     (IsInpChAssigned='0')Then
         IsInpChAssigned := '1';
         OutPackViCh := j;
   End If;
End Loop;
'WRITE (InpChAssigned , IsInpChAssigned)
```

Figure 3: Asynchronous implementation of XY routing algorithm

As it is clear from figure 3 there are two new macro functions in Verilog-CSP. These functions are ‘READ and ‘WRITE. Different modules are related with each other by asynchronous channels which are connected to their ports. Data communication is done by writing data to the ports and read it from the corresponding port on the other side of the channel. For write we use write macro: **‘WRITE (Port name, value).** If the sender wants to write another data on that port, it would be suspended until the last data read from that port: **‘READ (Port name, value).** The receiver module also remains suspended until a data is written on its counterpart port. Read and Write operation will be implemented by 4 phase handshake signaling.

5. Motivation

The issue of power consumption has been always controversial to many designers. Asynchronous circuits, due to clock elimination, have less power consumption compared to synchronous circuits. This is due to the fact that idle modules are off; while, in synchronous circuits they are on. It means there is no doubt that the dynamic power consumption in these circuits is lower than synchronous ones. However, asynchronous circuits like synchronous circuits suffer from static power consumption. Although technology scaling led to more static power consumption but dynamic power consumption is more critical than static power consumption. Our experimental results prove this claim.

One of the most common methods for attackers is power consumption analysis. Theses attackers use statistical techniques to determine the secret keys by observing power consumption. In a typical attack, an attacker samples the target device’s power consumption and builds a power trace. A high-speed analog-to-digital converter can be used to create these power traces. These measured power traces are compared with predicted power consumptions. To make a prediction a guess on the secret key is used. Several statistical and mathematical techniques are available to correlate the predictions and measurements. Based on these analyses the secret key can be found.

Our motivation is based on the encodings of the channels in asynchronous circuits. The encodings of the channels can be in a variety of ways. We use a dual rail encoding here. The data channel contains a valid data (token) when exactly one of 2 wires is high. When the two wires are lowered the channel contains no valid data and is called to be neutral (Figure 4).

	d.t	**d.f**
Neutral(“E”)	0	0
Valid ‘0’	0	1
Valid ‘1’	1	0
Not Used	1	1

Figure 4: Dual rail coding

In synchronous circuits when there is a switch between 0 to 1 and 1 to 0, we have power consumption spur due to charge and discharge of a capacitor. Attackers use this fact to determine the secret keys. We claim that asynchronous circuits will tackle these power consumption spurs based on their dual rail encoding. A transition from 0 to 1 in asynchronous circuits with dual rail encoding means a switch between “01” to “10” and vise versa. Figure 5 shows this fact that due to concurrent charging and discharging of capacitors we would not have considerable power consumption spur.

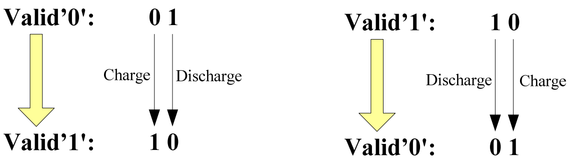


Figure 5: Transition from 0 to 1 and I to 0 causes no power spur in asynchronous circuits

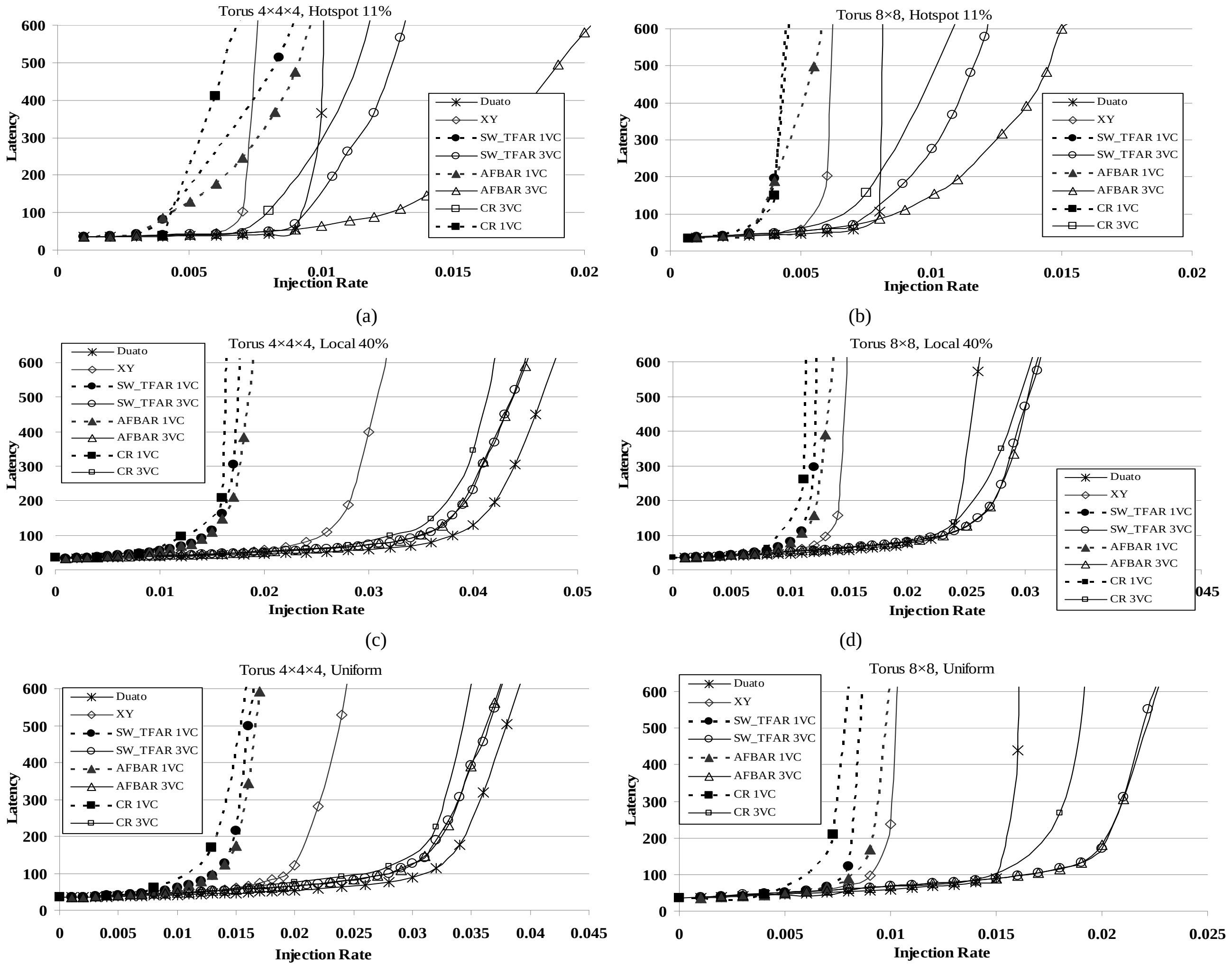
We implement our motivation in order to gain a secure and lower power NoC.

6. Experimental Results

In this section, first we will select the best routing algorithm which has the best performance (latency) in comparison with the other routing algorithms. We showed in [20] that performance and power consumption has a direct relation. The better performance leads to more power consumption and distribution in the network; hence, better routing algorithm can increase the complexity of power analysis attackers in order to determine the secret keys. Then, we will compare the power consumption spurs of this routing algorithm in an asynchronous and synchronous router under different traffic models.

6.1 Routing Algorithm Selection

In this section, the performance and latency of re-injection based deadlock recovery algorithms will be presented then we will compare their latencies with



(a) (b)

(c) (d)

(e) (f)

Figure 6: Latency comparison of deadlock recovery and deadlock avoidance routing algorithms

deadlock avoidance routing algorithms (i.e. Duato and XY).

In next step, we will analyze the energy consumption of mentioned routing algorithms under three traffic models, i.e. uniform, local 40% and hotspot 11. Our traffic models are combined with uniform traffic model.

As an exemplification, the local40% means that, 40% of the messages are distributed locally and the remaining 60% are distributed uniformly.

6.1.1 Latency Analysis

Figure 6 shows the latency comparison of deadlock recovery and avoidance algorithms. We know that a routing algorithm with more virtual channels usually gains better performance [20]. It is obvious from Figure 6 that all deadlock recovery routing algorithms with 3 virtual channels have a better performance than the ones with 1 virtual channel under all traffic models.

A key point is the dependency of latency to the network's diameter. We know that the diameter of a torus topology is calculated as follow:

$$\text{Torus Topology Diameter} = \sum_{i=1}^{n} \left\lfloor \frac{k_i}{2} \right\rfloor \qquad (6.1)$$

In (6.1), k_i is the number of nodes in i-th dimension and n is the number of dimensions of torus. Due to (6.1), the diameter of torus 4×4×4 topology is 6 and the diameter of torus 8×8 topology is 8. Although both topologies have 64 nodes, we claim that unequal diameter lengths lead to different latency behaviors in some situations.

To illustrate, it has been shown [10] that the AFBAR and SW_TFAR have a better performance than Duato routing algorithm with 3 virtual channels in a 2 dimensional torus topology under all traffic models. This claim is traceable from Figure 6(b,d,f) in 8×8 topology.

But, in 4×4×4 topology this is not correct and the best latency belongs to Duato with 3 virtual channels under local and uniform traffic models (Figure 6(c,e)). Shorter diameter in 4×4×4 topology leads to less blocking time; furthermore, each node has 6 input/output ports which enhance the adaptivity of each node to route packets. On the other hand, the killing or ejection and re-injection procedures of deadlock recovery routing algorithms lead to performance degradation. In fact, Duato routing algorithm dose not suffer from this process and also benefits from high node adaptivity in 4×4×4 topology which lead to slightly better performance.

Under hotspot traffic model, due to high traffic around the hot node and pipeline nature of wormhole switching, the blocking time is high and there are lots of packets in block chain. Hence, Duato can not benefit from more adaptivity of each node in 4×4×4 topology and it acts similar to XY routing algorithm. Therefore, Duato cannot use its virtual channels efficiently as opposed to deadlock recovery routing algorithms. Figure 6 (a) shows the better performance of deadlock recovery routing algorithms in comparison with Duato with 3 virtual channels. Unlike 4×4×4 topology, in 8×8 topology each node has 4 input/output ports that cause lower adaptivity for routing. Hence, the more efficient usage of virtual channels in deadlock recovery routing algorithms leads to their better performance in comparison with Duato routing algorithm in 8×8 topology.

We now analyze the deadlock recovery routing algorithms. In all conditions including topology, traffic model and 3 virtual channels, the best performance belongs to AFBAR. The performance gap of AFBAR and SW_TFAR is negligible under uniform and local traffic models. This is based on close number of deadlock detections, but this gap is considerable under hotspot traffic model, since AFBAR uses a more efficient deadlock detection mechanism (Table I).

TABLE I: COMPARISON OF NUMBER OF DETECTED DEADLOCKS [10]

Traffic Patterns	**Rate**	Num. of Detected Deadlock	
		3Virtuall Channel	
		SW_TFAR	AFBAR
Uniform	---	0	0
Local	*20%*	618	282
	40%	1596	984
	60%	3124	2002
Hotspot	*5%*	5	0
	10%	414	35
	15%	541	162

Another key point is the worse performance of CR in comparison with AFBAR and SW_TFAR with 3 virtual channels in all circumstances. Since CR kills the deadlock detected packets and re-injects them from source node again, its deadlock recovery overhead is more costly than AFBAR and SW_TFAR, which re-inject the deadlocked packets from the intermediate node that has detected the deadlocked packet. This manner is correct for deadlock recovery routing algorithms with 1 virtual channel, too.

6.1.2 Energy Analysis

The issue of energy consumption has been controversial to many designers and fabricators due to extension of portable products such as mobile phones or laptops. In this section, we analyze the energy consumption of deadlock recovery routing algorithms and compare their energy consumption with deadlock avoidance routing algorithms.

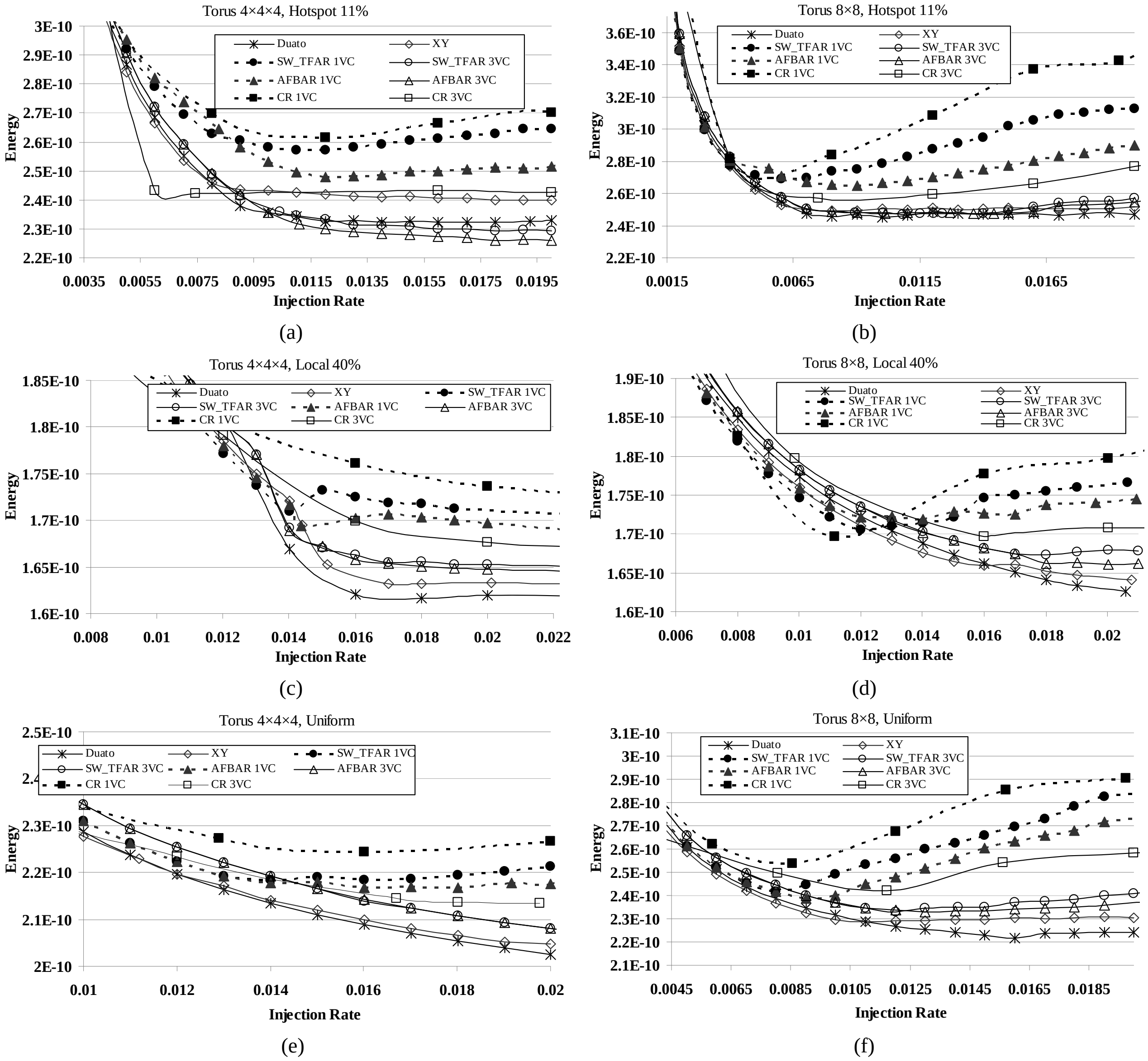
Figure 7 shows the energy consumption of the mentioned routing algorithms under hotspot, local and uniform traffic models in torus 4×4×4 and torus 8×8. The important point that the authors would like to make is that the worst energy consumption belongs to CR with 1 virtual channel, since the worst delay is associated with CR (Figure 6). In addition, CR deadlock recovery procedure leads to more number of transferred flits per cycle, which increases the power consumption as well.

Another important point is that in most cases, the best energy consumption is gained by Duato routing algorithm among all routings algorithms. Using three virtual channels by Duato leads to better performance and lower delay. Additionally, this algorithm does not kill or re-inject any packets to the network due to its deadlock avoidance nature. Hence, this algorithm has lower power consumption as it would not increase the number of transferred flits per cycle in comparison with deadlock recovery routing algorithms.

There is an exception were Duato does not have the best energy consumption. This case appears in torus 4×4×4 with hotspot traffic (Figure 7 (a)). As aforementioned, in this topology, the lower diameter (i.e. 6) reduces the number of blocking packets and so SW_TFAR and AFBAR will detect lower number of packets engaging in deadlock cycle. Therefore, the number of re-injected packets and number of transferred flits per cycle will reduce which leads to lower power-delay product or energy consumption.

Deadlock recovery routing algorithms with 1 virtual channel consume more energy compared to deadlock recovery routing algorithms with 3 virtual channels in all cases (Figure 7). Although more virtual channels cause more power consumption, it does not increase energy consumption. It means more efficient usage of virtual channels by deadlock recovery routing algorithms, which leads to better performance, overcomes the more power consumption based on more virtual channels. On the other hand, this fact is not applicable to deadlock avoidance routing algorithms [20].

It was mentioned in [10] that AFBAR has a lower delay in comparison with SW_TFAR which is verified in Figure 6. However, our experimental results show that less



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7: Energy consumption comparison of deadlock recovery and deadlock avoidance routing algorithms

energy is consumed by AFBAR than SW_TFAR. Therefore, we can come into conclusion that AFBAR is more efficient than SW_TFAR due its better performance and lower energy consumption. The energy consumption gap for AFBAR and SW_TFAR is more obvious with 1 virtual channel than 3 virtual channels, as the better deadlock detection mechanism of AFBAR is more efficient in lower number of virtual channels.

In a nutshell, we can say that whenever the energy consumption is a critical parameter for a designer, the Duato deadlock avoidance routing algorithm is a better selection, and whenever the performance or delay is the critical parameter, the AFBAR deadlock recovery routing algorithm is the best choice. In this paper the latency is more critical for us because asynchronous circuits have lower power and energy consumption; therefore, we will use AFBAR in the next section as it has the best performance among other routing algorithms.

8.2 Security Validation of an Asynchronous NoC

We had mentioned before that one of the most common methods for attackers in order to determine the secret keys is power consumption analysis. We had claimed that asynchronous circuits have the least power consumption spurs during the transitions in comparison with synchronous one. In this section we will follow up the correctness of this claim and we will prove it. We will run different traffic models in an asynchronous and synchronous router and show the power consumption

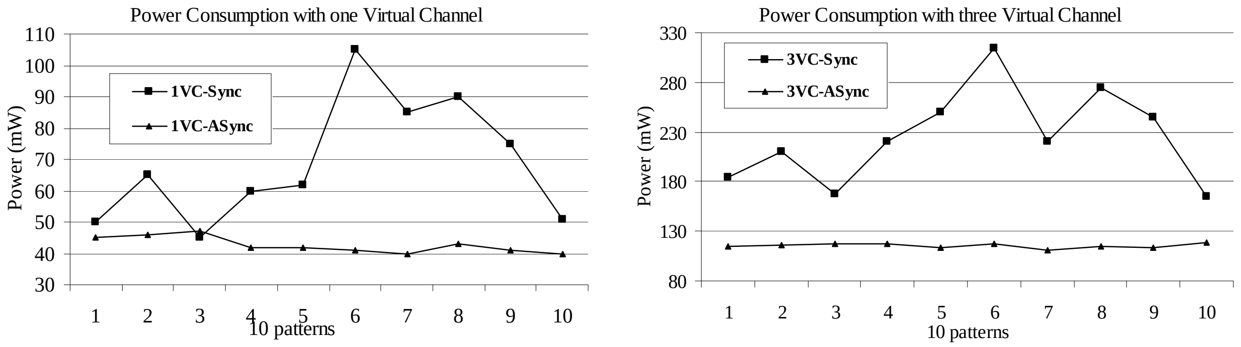
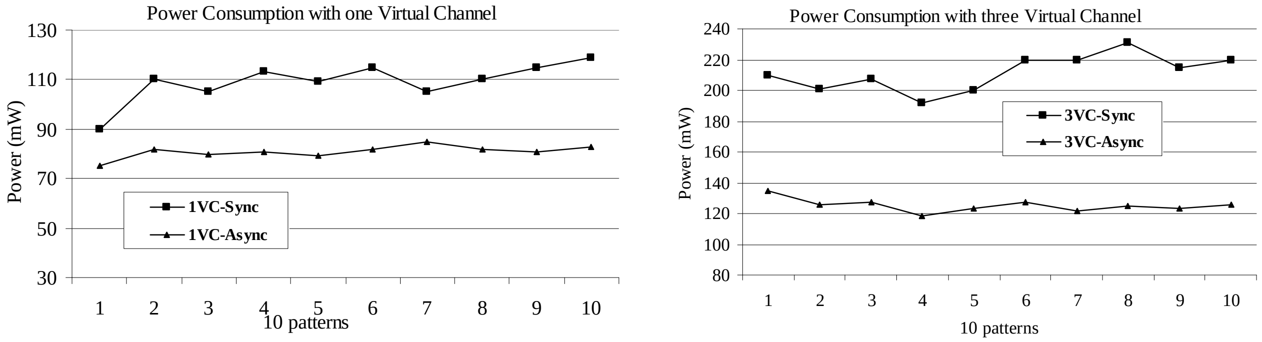


Figure 8: Power consumption spurs in Uniform traffic model



(c) (d)

Figure 9: Power consumption spurs in Hotspot 20% traffic model

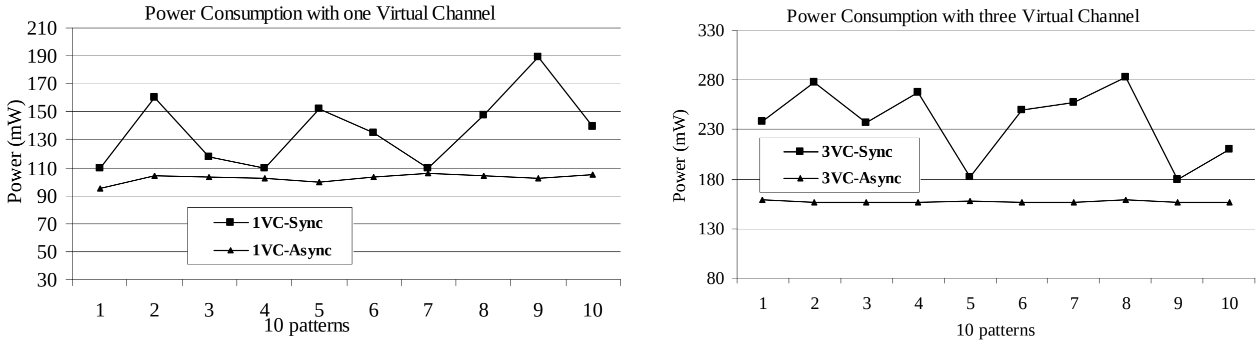


Figure 10: Power consumption spurs in Local 40% traffic model

spurs. The traffic models have different patterns. To illustrate, we will generate different patterns of 0 and 1 in a packet, but the number of 1s and 0s is constant in all of the packets. It means, if we have 1000 packets and each packet is 32 bytes then we will have 1000×32×4 ones and 1000×32×4 zeros.

Figure 8, 9, and 10 show the power consumption comparison of an asynchronous and synchronous router under uniform, hotspot 20%, local 40% traffic models respectively, and 10 different patterns, AFBAR routing algorithm, one and three number of virtual channels. To determine the power consumption and latency of the asynchronous router, we used Persia to synthesis the asynchronous router and then used HSPICE to determine the power consumption and latency. As we know there is no any commercial asynchronous circuits based on QDI synthesis tool. For synchronous router we used Leonardo tool and then HSPICE to determine the power consumption and latency.

It is clear from figure 8, 9 and 10 that the power consumption of an asynchronous router is always lower than synchronous one. This is due to the fact that in asynchronous circuits the idle models are off and have no dynamic power consumption. The first precious aspect of this framework is its low power design.

Another important point is that, asynchronous router under all traffic models, diverse patterns and different number of virtual channels has no considerable power consumption spur. As mentioned before, different patterns have just different number of transitions between 1s and 0s but the number of 1s and 0s during one runtime is equal. However, synchronous router has many power consumption spurs which help the attackers to determine the secret keys by analyzing the power consumption. In continue, we will analyze the behavior of power consumption diagram of a synchronous router under different traffic models.

8.2.1 Uniform Traffic Model

In uniform traffic model, each node sends its messages to any other node with equal probability. For example in a torus with 9 nodes the probability will be 0.11. Synchronous router has some spurs under this traffic model. This id due to the fact that uniform traffic model distribute the traffics uniformly which leads to more switching activity and power consumption. In [11], four important factors were introduced in order to evaluate the traffic pattern effect on power consumption in Mesh and Torus Network-on-Chips. These factors are: **F1:** Length of the link between adjacent routers, **F2:** Distance between source and destination in terms of number of links, **F3:** Number of channel monitorings in a cycle and **F4:** blocking time. Due to good traffic distribution in this traffic model and synchronous nature of the router, it is completely logical to have some spurs under different patterns of packets.

8.2.2 Hotspot Traffic Model

In hotspot traffic model, each node sends specific portion of its generated messages to the hot node. As an illustration, in hotspot 20%, each node sends 20% of its generated messages to the hot node and we suppose that they distribute the other messages uniformly in the network. An important key point is that, the power consumption spurs in hotspot traffic model is not more than uniform or local. This fact is clear from figure 9 that F4 or blocking time reduces the switching activity in the network; hence, different patterns have no considerable different power consumption and the most part of the power consumption dedicated to channel monitorings or F3. We can say that hotspot traffic model is to some extent complex for attackers to determine the secret keys.

8.2.3 Local Traffic Model

In local traffic model, each node sends specific portion of its generated messages to its neighbors within a predefined distance, called neighborhood radius. In this paper we used Local 40% with radius one. We can recognize from figure 8 to 10 that the most power consumption is dedicated to local traffic model, this due to F2 factor. More explanations are available in [16]. However, another important factor is that the most number of spurs are also dedicated to this traffic model. It means this the worst traffic model from security point of views. The reason is that, in local traffic model many of messages deliver to neighbor node and due to F2 factor we have short traveling length for each message which leads to least blocking time (F4) in the network among other traffic models. Hence, we will have many transitions in each node and as mentioned before in a synchronous router each transition will show its effect on power consumption; therefore, this traffic model will have the most power consumption spurs among other traffic models.

But we can see that our asynchronous router is completely independent of traffic model, number of virtual channels and patterns of 0s and 1s in each packet.

10. Conclusions

In this paper we introduce a new framework for Network-On-Chips which was asynchronous, secure and low power. First, we introduce the asynchronous circuits, and Persia as an automatic synthesis tool for asynchronous circuits based on QDI and PCFB and PCHB templates. We mentioned that Persia has three important stages AFE, Decomposition and TSYN and described all of them.

Then we explained architecture of an asynchronous router based on QDI, PCFB template and four-phase handshaking. We also described the different aspects of an asynchronous router, such as links, internal buffers, Arbiter, Virtual Channel Allocator, Crossbar Switch, Switch Controller, Link Controller and Routing Unit.

We considered the different routing algorithms and compared the most famous ones such as: XY, Duato as deadlock avoidance routing algorithms and AFBAR, SW_TFAR and CR as re-injection based deadlock recovery routing algorithms. We showed that the best routing algorithm when latency is critical is AFBAR and when energy consumption is critical, Duato routing algorithm is the best one.

Finally we compare our asynchronous framework with traditional synchronous one under AFBAR routing algorithm and different famous traffic models such Uniform, Hotspot and Local with different number of virtual channels. Our asynchronous framework has lower power consumption in comparison with a traditional synchronous router. It was due to the fact that asynchronous circuits have no global clock; hence, the idle modules are off and will no more consume the dynamic power consumption. Figure 8 to 10 show that an asynchronous router under all conditions has lower power consumption. Another important factor was that our frame work due to its encoding style (Dual Rail) has no power consumption spurs under different input patterns. It means our framework was completely input independent which means this is a secure framework and power analyzer attackers can not infect to our framework in order to determine the secret keys.

In a nutshell, in this paper we introduced an asynchronous framework for routers and links which was low power and secure compared to its traditional synchronous router.

References

[1] A. Allen, D. Edenfeld, W.H. Joyner, A.B. Kahng, M. Rodgers, and Y. Zorian, "2001 Technology Roadmap for Semiconductors," IEEE Computer, pp. 42-53, Jan. 2002.

[2] J. Duato, S. Yalamanchili, and L. Ni, "Interconnection Networks—An Engineering Approach," Morgan Kaufmann, 2002.

[3] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Proc.19th Intl. Advances in Cryptology Conference-CRYPTO'99,* Aug. 1999,pp. 388–397.

[4] M. Mirzaaghatabar, M. Najibi, K. Saleh, H. Pedram, "Exploring the Design of A RISC Asynchronous Processor Using Persia Asynchronous Toolset", *Proceedings of the 14th Iranian Conference on Electrical Engineering (ICEE2006),* May 2006

[5] Scott Hauck, "Asynchronous design methodologies: An overview", *Proceedings of the IEEE*, 83(1):69-93, January 1995

[6] J. Sparso, S. Furber, "*Principles of Asynchronous Circuit Design – A System Perspective*", Kluwer Academic Publishers, 2002.

[7] Alain J. Martin, "Synthesis of Asynchronous VLSI Circuits",*Caltech*, *CS-TR-93-28*, 1991.

[8] Catherine H. Gebotys, Robert J. Gebotys: A Framework for Security on NoC Technologies. ISVLSI 2003: 113-120

[9] Evain, S. Diguet, J.-P , "From NoC Security Analysis To Design Solutions", IEEE Workshop on Signal Processing Systems Design and Implementation, 2005, pp: 166-171

[10] M. Mirza-Aghatabar, A. Tavakkol, H. Sarbazi Azad, "An adaptive software-based deadlock recovery technique," IEEE International Conference on Advanced Information Networking and Applications (AINA Workshop 2008) in Japan, pp:514-519.

[11] S. Koohi, M. Mirza-Aghatabar, S. Hessabi, "Evaluation of Traffic Pattern Effect on Power Consumption in Mesh and Torus-based Network-on-Chips," International Symposium on Integrated Circuits, Singapore, 2007

[12] J. Muttersbach, T. Villiger, and W. Fichtner. Practical Design of Globally-Asynchronous Locally-Synchronous Systems. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, April 2000.

[13] A. Seifhashemi, H. Pedram, "Verilog HDL, Powered by PLI: a Suitable Framework for Describing and Modeling Asynchronous Circuits at All Levels of Abstraction", *Proc. Of 40th DAC, Anneheim, CA, USA*, June 2003

[14] J.M. Martinez, P. Lopez, J. Duato and T.M. Pinkston, "Software-Based Deadlock Recovery Technique for True Fully Adaptive Routing in Wormhole Networks," 1997 International Conference Parallel Processing, Aug. 1997

[15] K. V. Anjan and T. M. Pinkston and J. Duato, "Generalized theory for deadlock-free adaptive routing and its application to Disha Concurrent," In Proc. of the 10th International Parallel Processing Symposium, April 1996

[16] K. V. Anjan and T. M. Pinkston, "An Efficient, Fully Adaptive Deadlock Recovery Scheme: DISHA," In Proc. of the 22nd International Symposium on Computer Architecture, pages 201-210, June 1995

[17] K. V. Anjan and T. M. Pinkston, "DISHA: A Deadlock Recovery Scheme for Fully Adaptive Routing," In Proc. of the 9th International Parallel Processing Symposium, pp. 537-543, April 1995

[18] L. M. Ni and P. K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," IEEE Computer, Vol. 26, No. 2, pp. 62-76, February 1993

[19] W.J. Dally and C.L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," IEEE Trans. Computers, Vol. 36, No. 5, pp. 547-553, 1987

[20] M. Mirza-Aghatabar, S. Koohi, S. Hessabi, and Massoud Pedram, "An empirical investigation of Mesh and Torus NoC topologies under different routing algorithms and traffic models," in Proceedings of the 10th IEEE Euromicro Conference on Digital System Design (DSD) , pp. 19-26, 2007

[21] J. Kim, Z. Liu and A. Chien, "Compressionless Routing: A framework for adaptive and fault-tolerant routing," In Proc. of the 21st International Symposium on Computer Architecture, pages 289-300, April 1994.

Mohammad Mirza-Aghatabar received his B.Sc. degree from AmirKabir University of Technology in Hardware and Software Computer Engineering in 2005 as a Top Student. He received his M.Sc degree from Sharif University of Technology in Computer Architecture in 2008. Now, he is PH.D candidate in University of Southern California (USC). His research interest includes Network-on-Chip (NoC) circuit design, Low Power circuit design, Asynchronous circuit design, Fault Tolerant system design and Design for Testability.

Abolghasem Sadeghi received his B.Sc. degree from Sharif University of Technology in hardware computer engineering in 1999. He received his M.Sc degree from Sharif University of Technology in Computer Architecture in 2001. Now, he is faculty member of ICT department in Malek Ashtar University of Technology. His research interest includes Network Security, Security Auditing and Analyzing, Image Processing and Signal Processing.