

# Parallel and Fault-Tolerant Routing in Nanoscale Spin-Wave Architectures<sup>1</sup>

Mary M. Eshaghian-Wilner<sup>2</sup>  
Electronics Practice, Foley & Lardner LLP  
and Electrical Engineering Dept., UCLA  
California, USA

Shiva Navab  
Broadcom Corporation  
and Electrical Engineering Dept., UCLA  
California, USA

**Abstract** - *In this paper, we present a number of parallel and fault-tolerant routing schemes for a set of nanoscale spin-wave architectures. The architectures considered here have several features, including the ability to simultaneously transmit multiple data on the same spin-wave bus using different frequencies, as well as the capability to perform concurrent writes. These parallel features result in several parallel and fault-tolerant routing schemes that are investigated here. By alternating paths to transmit data, the spin-wave architectures can be reconfigured to avoid various faults present in the underlying switches, hence rendering a set of fault-tolerant architectures.*

**Keywords:** parallel routing, fault-tolerance, nanoscale architectures.

## 1 Introduction

Emerging nanoscale device technologies such as carbon nanotubes, quantum dots, molecular crossbars, and single electron transistors have been proposed with the aim of increasing the densities of integrated circuits [1]. However, these nanoscale designs suffer from dramatically increased permanent and transient failure rates. These failures are mainly due to the quantum nature of the devices as well as the fundamental limitations of the fabrication processes [2]. Fault tolerance will be one of the main concerns in the adoption of new approaches in nanotechnology.

One well-known approach for developing reliable nano architectures that deals with manufacturing and transient defects incorporates spatial and/or temporal redundancy [3]. In recent years, different tools have been developed to evaluate certain design trade-offs in the nanoscale

architectures. Some of the most important trade-offs are between granularity and reliability [4], and between redundancy and reliability [5]. Towards that, researchers have applied different degrees of redundancy to different granularity levels (i.e., gate and reconfigurable logic block levels) [5].

Many techniques have been studied to increase the tolerance of nanoscale architectures to both transient and fabrication defects [6-15]. For instance, a number of redundancy schemes, including Von Neumann's multiplexing logic, N-tuple modular redundancy, and interwoven redundant logic have been presented in [12]. In addition, a new fault-tolerant design approach based on coding theory has recently been proposed at HP Labs [13]. In their approach, by using a crossbar architecture and adding 50% more wires, nano-electronic circuits with impressive yields can be fabricated.

To implement fault-tolerant quantum computers, quantum error correcting codes are being developed and elementary quantum gates are being constructed to form the basic building blocks of these computers [14]. Furthermore, a number of logic-mapping algorithms with defect avoidance have been presented in [15] to circumvent clustered defective crosspoints in nanowire reconfigurable crossbar architectures.

Fault tolerance is achieved in all these architectures by adding some level of redundancy. While redundancy is needed for reliable computation, in choosing the redundancy factor, economic constraints also need to be considered [3]. One of the advantages of the fault-tolerant schemes presented in this paper as compared to the other methods mentioned previously is in the smaller degree of redundancy that is required.

---

<sup>1</sup> Authors are listed in alphabetical order

<sup>2</sup> Mary M. Eshaghian-Wilner is a Patent Agent at the Electronics Practice Group of Foley & Lardner, LLP. She is also an Adjunct Professor of Electrical Engineering at the University of California, Los Angeles.

In this paper, we concentrate on the parallel and fault-tolerance features of a set of spin-wave nanoscale architectures [16, 17, 18]. We show that by employing the parallel features of these architectures, the amount of the spatial redundancy required for a fault-tolerant design is significantly reduced.

These spin-wave architectures have several features, including the ability to simultaneously transmit multiple data on the same spin-wave bus using different frequencies as well as the capability of performing concurrent writes. These features result in parallel routing schemes such as multiple arbitrary permutations, broadcasting, and data transmission from multiple inputs to a single output. By alternating paths used to transmit data, the spin-wave architectures can be reconfigured to avoid the faults present in underlying switches, hence rendering fault-tolerant architectures.

The rest of the paper is organized as follows: in Section 2, we present three spin-wave architectures. Several of their parallel and fault-tolerant routing schemes are discussed afterwards in Sections 3 and 4, followed by the concluding remarks in Section 5.

## 2 Spin-Wave Architectures

In this section, we provide an overview of three spin-wave architectures: a spin-wave crossbar, a spin-wave reconfigurable mesh, and a spin-wave fully interconnected cluster. For more detailed explanations on these architectures, please refer to [16, 17, 18] respectively. In the spin-wave architectures presented here, spin waves are used for both information transmission and information processing [19]. A spin wave is a collection of precession of an electron’s magnetic moment about a magnetic field. In the spin-wave crossbar architecture, the classical type of computing is employed, as opposed to quantum, and this architecture can operate at room temperature.

### 2.1 Spin-Wave Crossbars

Crossbars are attractive architectures because they can realize any permutations of  $N$  inputs to  $N$  outputs. However, their main shortcoming is that  $N^2$  switches are used to transmit only  $N$  pairs of data. The architecture described here, while requiring the same number of switches as standard crossbars, is capable of transmitting  $N^2$  data elements. This is because each spin-wave bus is capable of carrying multiple waves at any given time by using different frequencies. Therefore, each of the  $N$  inputs in parallel can essentially broadcast its data to all of the  $N$  outputs. As compared to molecular nanoscale crossbars, this design is more fault-tolerant, as shown in the next section, because if there is a failure in one of the  $N$  channels, other channels can be used to transmit the data. This is possible because all the channels are accessible by

all the ports and each channel can handle multiple data. An example of the proposed spin-wave cross-bar architecture is shown in Figure 1.

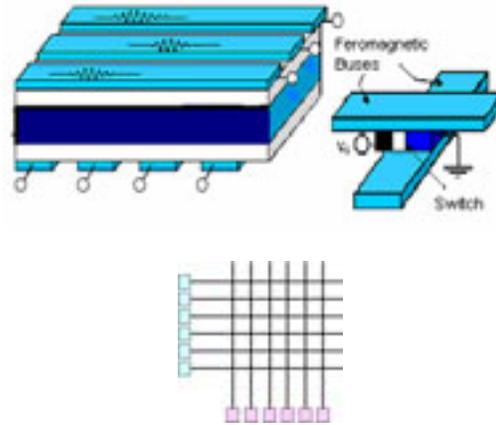


Figure 1 - Spin-Wave Crossbar Architecture

Note that a set of column spin-wave buses on the bottom and a set of row spin-wave buses on the top are connected via the vertical spin-wave switches. A spin-wave switch is a device that has an externally controllable magnetic phase. In the “On” state, the switch transmits spin waves, while in the “Off” state it reflects any incoming spin wave. As described in [16], the ferromagnetic film is divided by a region of diluted magnetic semiconductor (DMS), and it is used as a magnetic channel. The magnetic phase is controlled by the applied electric field via the effect of hole-mediated ferromagnetism. A negative gate bias increases the hole concentration in the DMS region, resulting in the paramagnetic-to-ferromagnetic or Off-to-On transition, whereas a positive bias has the opposite effect.

### 2.2 Spin-Wave Reconfigurable Mesh

A nanoscale reconfigurable mesh of size  $N^2$  consists of an  $N \times N$  array of processors connected to a reconfigurable spin-wave bus grid, where each processor has a locally controllable bus switch. An example of the proposed spin-wave reconfigurable mesh architecture is shown in Figure 2. Note that the column spin-wave buses and the row spin-wave buses are connected via the spin-wave switches.

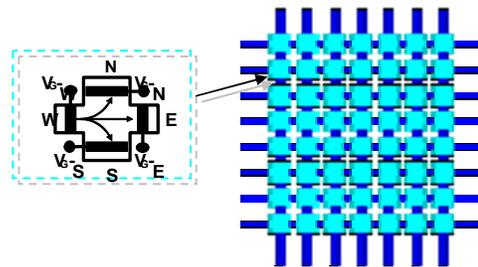


Figure 2 - The Nanoscale Reconfigurable Mesh

Each switch is placed at the grid point of the mesh. The switches allow the broadcast bus to be divided into sub-buses providing smaller reconfigurable meshes. These switches are similar to crossbar switches, except each reconfigurable mesh switch has four controllable gates to route the signal in different directions.

Basically, except for the spin-wave buses, the nanoscale reconfigurable mesh with spin-wave buses is similar to the standard reconfigurable mesh. It is worth noting that, similar to the reconfigurable mesh (and standard mesh), the nanoscale reconfigurable mesh of size  $N \times N$  occupies  $N \times N$  area, under the assumption that processors, switches, and a link between adjacent switches occupy unit area. However, the main difference in term of area here is that the unit of area is at nanoscale level, as opposed to the standard reconfigurable meshes that are currently available at micro-scale level of integration.

### 2.3 Spin-Wave Fully Interconnected Cluster

A fully interconnected architecture consists of  $N$  computing nodes, all of which intercommunicate with spin waves. Figure 3 shows the top view of the architecture in which the  $N$  computing nodes are placed around a circle on a magnetic film. The area requirement of this architecture is  $O(N^2)$ , as opposed to the  $O(N^4)$  area requirement if electrical interconnects were to be used. We should also note that all the distances in this architecture are at the nanoscale level.

Unlike electrical interconnection networks, in which only one transmission can be done at a time, here multiple simultaneous permutations are possible by transmitting the spin waves over different frequencies. The information is coded into the phase of the spin waves in the sender and is detected by the receivers. In addition, within each frequency, data can be sent to one or more other nodes from each node.

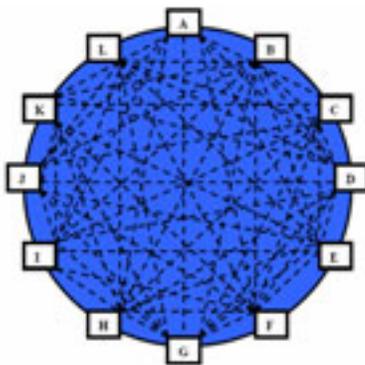


Figure 3 - The Top View of a Spin-Wave Fully Interconnected Cluster

Normally, in architectures where the phases of the waves are the means of information transmission, the exact location of the nodes is an important design issue. The distance between the sender and receiver has to be at a length that is a multiple of the wave's wavelength; otherwise, the receiver might receive the wave with a  $\pi$  radian phase-shift, which is a "0" instead of a "1" or vice versa. However, in our design, this is not an issue because the wavelengths of spin waves are considerably larger than the distance between the nodes. The speed of spin waves is around  $10^5$  m/s. Assuming the input frequency range of 1-10 GHz (as in our experiment), the wavelength will be in the order of  $10^{-4}$  to  $10^{-5}$  m, while the distances are nanoscale or  $10^{-9}$  m. In other words, the wavelengths of the spin waves are some orders of magnitude greater than the distances between the nodes. Therefore, all the nodes receive the same phase regardless of their location, and there is no need to place the nodes in specific distance relative to the other ones.

## 3 Parallel Routing and Broadcasting

In the following, we illustrate the routing features of our three spin-wave architectures. We focus primarily on the routing on a spin-wave crossbar, because the routing on the other two architectures is similar to the routing on the crossbar. We also discuss an enhanced multiple multicasting feature on the fully interconnected cluster.

### 3.1 On a Spin-Wave Crossbar

Our spin-wave crossbar has several parallel and fault tolerant routing features. In the following we concentrate the routing features of this architecture in three different scenarios. These techniques are then compared with those for the reconfigurable spin-wave architecture and the fully connected spin-wave architecture presented later in this paper.

It is well known that all crossbars are capable of realizing any arbitrary one-to-one permutation. In a standard VLSI crossbar, however, unless there are broadcasting buses on each row, at any single point in time, only one switch is turned on in each row and each column. Spin-wave crossbars, on the other hand, support additional features such as broadcasting and concurrent receiving as described below.

#### 3.1.1 Arbitrary Permutations

Similar to any standard crossbar, a spin-wave crossbar realizes arbitrary permutations. This section illustrates how this is done. In the crossbar architecture, the signals are directed in each row and each column through spin-wave buses. As an example of a one-to-one permutation realization, assume that input 3 needs to send a message to output 6. In that case, the switch in row 3 and column 6,

represented as  $s(3,6)$  should be set to “on”. In addition, the receiving frequency of node 6 should be tuned to sending frequency of node 3. The switches can be set to “on” according to the following mechanism: A fixed frequency is assigned to each column, and on top of each switch there is a receiver that is tuned to the frequency assigned to its column. As soon as the switch receives a signal on its frequency, it activates and routes the data. For instance, switch  $s(3,6)$  is tuned to the frequency assigned to column 6,  $f_6$ . Input node 3 sends a signal on frequency  $f_6$  on row 3, which turns on  $s(3,6)$ . Now, the third row is connected to the sixth column, and permutation  $(3,6)$  is realized. Figure 4 shows this communication on a crossbar of size 6. Note that there is a switch located on each of the grid points, but here we are just showing the one that is used.

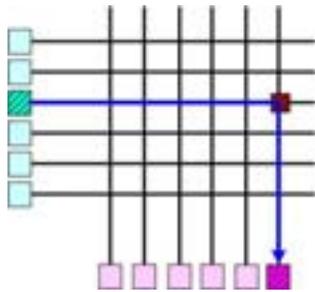


Figure 4 - Arbitrary Permutation

### 3.1.3 Broadcasting Feature

Broadcasting at a node happens when that node sends a single message to multiple receivers. Realizing broadcasting in a spin-wave crossbar is slightly different from realizing concurrent receive. In this case, a fixed frequency is assigned to each row (each sender), and the receivers tune their receiving frequency to that. As explained earlier, one of the most important advantages of a spin-wave crossbar is that one input can broadcast to multiple outputs simultaneously. For instance, node 3 can broadcast a message to output 2, 4, 5, and 6 at the same time, as shown in Figure 6. The only constraint is that the receiver nodes should be tuned to the sender's frequency.

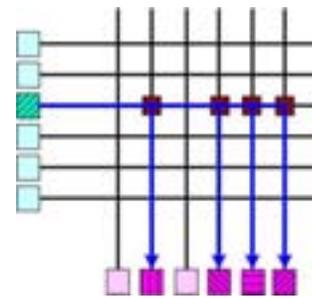


Figure 6 - Broadcasting Feature

### 3.1.2 Concurrent Receive Feature

Realizing the concurrent receive feature is similar to realizing the one-to-one permutation described above. A fixed frequency is assigned to each column (each receiver), and the senders tune their sending frequency to that frequency. One of the important features of a spin-wave crossbar is that it allows concurrent write. For instance node 2, 3 and 4 can all send a message to node 5, as shown in Figure 5. Due to the superposition property of waves, output 5 receives a signal that is the sum of these three waves. In a standard VLSI crossbar, it is not possible to perform these three communications simultaneously because such a situation will cause a conflict on column 5.

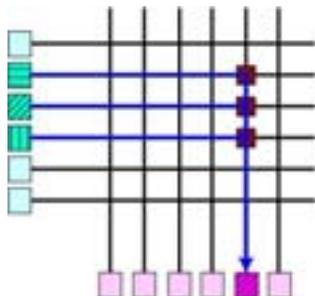


Figure 5 - Concurrent Receive Feature

### 3.2 On a Spin-Wave Reconfigurable Mesh

The routing on a reconfigurable mesh is similar to a crossbar. However, this routing can be from any of the  $N^2$  processing elements to any other, so there can exist up to  $N^2 \times N^2$  different routing schemes.

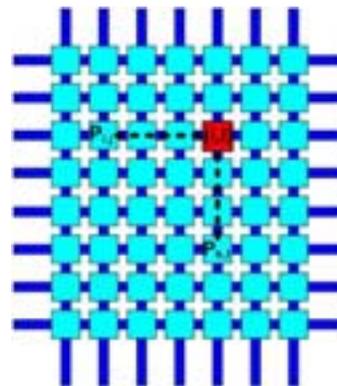


Figure 7 - Routing in Reconfigurable Mesh

The routing mechanism in a spin-wave reconfigurable mesh is as follows: To send information from  $P_{i,j}$  to  $P_{k,l}$ , the sender,  $P_{i,j}$  sends the signal to switch  $s(i,l)$  to be routed to  $P_{k,l}$  as shown in Figure 7.

As mentioned in the previous section, the significance of spin-wave architectures is that multiple waves on different frequencies can pass through the same bus without any conflict. For instance  $P_{3,2}$  can send a signal to  $P_{6,5}$ , while  $P_{3,3}$  is sending a signal, on the same row and column, to  $P_{5,5}$ .

### 3.3 On a Fully Interconnected Cluster

The routing on a spin-wave fully interconnected cluster is similar to the routing on a spin-wave crossbar, except there are not any switches on this architecture. In addition, the fully interconnected cluster has an extra feature, which we explain later in this section.

Similar to a crossbar, concurrent receive feature applies here as well. At a given frequency, a node can listen to multiple waves simultaneously. Using the superposition property of waves, that node receives the sum of all waves destined to it. For instance, multiple senders send data to G at the same time, and G receives the sum of those signals. In this case, the requirement is that all the nodes should transmit at the same frequency that is also the frequency at which G's receiver is tuned.

Multiple broadcasting is possible here too. To distinguish the data being transmitted to different nodes, transmissions are done at distinct frequencies, using frequency division multiplexing. In a way, this is similar to having various radio stations, each broadcasting at a different frequency. To listen to a specific station, one tunes to the corresponding frequency. Figure 8 shows an example, where node A is sending to a set of nodes, while C is sending to another set.

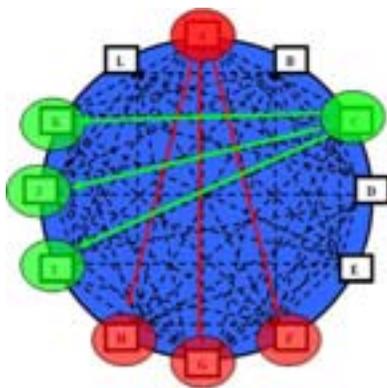


Figure 8 - Multiple Broadcasting on Disjoint Sets of Receivers

Note that since different senders broadcast to different sets on different frequencies, the sets must be disjoint. However, as pointed out earlier, the fully interconnected network has an additional feature comparing to the other two architectures. The feature allows multiple broadcasting to sets that are not disjoint. This is basically the combination of concurrent receive and multiple broadcasting as shown in Figure 9.

In the scenario shown in Figure 9, one of the A and C destinations is the same (node H). This means that here the sending frequency of A and C are the same as the receiving frequency of H. Consequently, the receiving frequency of K, J, G, and F is the same too, which causes each of these nodes to receive the superposition of the signals sent by A and C.

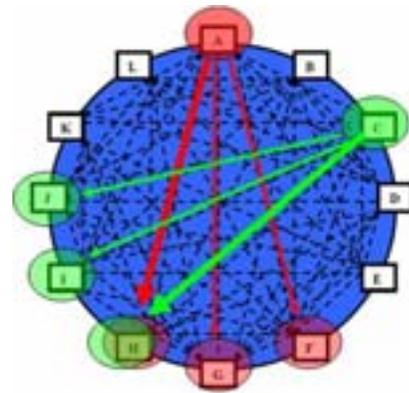


Figure 9 - Multiple Broadcasting with overlapping frequencies

One approach to designing fully interconnected clusters with overlapping frequencies, would be to use phased array techniques explained in [20] to direct the waves to specific locations. It is also possible to combine the phased array technique with multiple frequencies. This way, for each frequency, some of the waves are transmitted only to desirable directions and are received by the intended sources.

## 4 Fault-Tolerant Routing

As discussed previously, fault tolerance is one of the most important requirements for nanometer scale devices and architecture. The scale of elementary logic devices and the number of devices integrated in a circuit create a great demand in a fault-tolerant architecture. For such a densely integrated circuit to perform a useful computation, it has to deal with the inaccuracies and instabilities introduced by fabrication processes and transient faults that may spontaneously occur during circuit lifetimes. As mentioned, one solution is in the use of redundant components to obtain reliable synthesis from unreliable components. However, the use of the redundant components increases the number of devices per logic circuit and smashes the

advantage of high-density nanoscale logic circuits. The use of waves as a physical mechanism for information transmission lets us utilize fewer devices for redundant components in comparison to the electron-based devices. In this section, we focus on the fault-tolerance features of the spin-wave crossbar. We first briefly talk about fault diagnosis and then present a simple fault recovery scheme.

### 4.1 Fault Diagnosis

There are many different ways in literature to detect a defective switch [21-24]. Here we choose a very simple method, in which an acknowledgment is sent from the receiver back to the sender for each transmission. If the sender does not receive an acknowledgment from the receiver after a fixed amount of time, existence of a fault has been determined. The sender then tries to resend the message through another route.

### 4.2 Fault Recovery

In the example presented earlier where input 3 was sending to output 6, assume that the switch  $s(3,6)$  is defective. So now after node 3 sends a message to node 6, it does not receive the acknowledgement. Therefore, sender 3 will attempt to resend the message to node 6 through a new path. There are several schemes to reroute the message. The method we employ to reroute the path is simple. It is basically performed by adding an extra column of switches to the crossbar. In case of a fault on any of the switches on row  $i$ , the input, using the extra column, connects with the spin-wave bus path on row  $i+1$  (or row  $i-1$  if in the last row), and then from there goes to its destination, as shown in the example below. We refer to the switches in the extra routing column as switches in column 0. In our example, node 3 connects to row 4 via this switch and reroutes the path through  $s(4,6)$ , as shown in Figure 10.

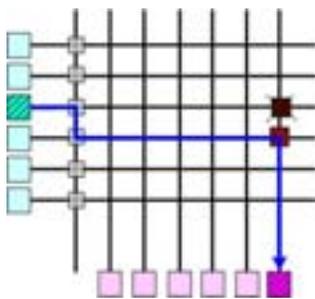


Figure 10 - Fault Recovery Example

The significant advantage of a spin-wave crossbar over a standard VLSI crossbar is that this simple scheme for rerouting does not collide with other intercommunications along the same row or column. For instance, as shown in Figure 11, in the same example, node 4 can still send a message to node 5 while 3 sends a message to 6 via  $s(4,6)$  in row 4. Although the rerouted path passes through row 4,

these communications can be done in parallel with no conflict. This is due to the fact that nodes 3 and 4 use different frequencies, so their signal waves pass through each other without interference.

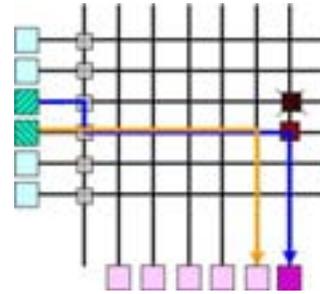


Figure 11 - Parallel Communications on a Spin-Wave Bus

Note that in this example, the two signals from input 3 and 4 go through row 4, as well as both columns 5 and 6. So the two input messages reach both outputs 5 and 6; however, output 5 detects the message from input 4 on its tuned frequency, while output 6 receives the signal from input 3.

Fault-tolerant routing in a spin-wave reconfigurable mesh can be performed in the exact same fashion as described for a spin-wave crossbar. Fault-tolerant routing discussion is not applicable to the fully interconnected architecture since there are no underlying switches in that architecture; hence, there are no faults in the communication medium to be diagnosed or recovered.

## 5 Conclusion

In this paper, we presented a number of parallel and fault-tolerant routing schemes for a set of nanoscale spin-wave architectures. As discussed, these architectures have several features, including the ability to simultaneously transmit multiple data on the same spin-wave bus using different frequencies as well as the capability of performing concurrent writes. These parallel features result in concurrent and fault-tolerant routing schemes such as multiple arbitrary permutations, broadcasting, and data transmission from multiple inputs to a single output. By alternating the paths to transmit the data, the spin-wave crossbar and reconfigurable mesh can be reconfigured to avoid the faults present in underlying switches, hence rendering fault-tolerant architectures. The key advantage of these architectures over other nanoscale architectures is that they hardly require any additional hardware, and yet they are comparable to other fault-tolerant architectures that use different types of redundancy.

## 6 References

- [1] M. M. Eshaghian-Wilner, A. H. Flood, A. Khitun, J. Fraser. Stoddart, and K. L. Wang., "Molecular and Nano-scale Computing and Technology," to appear as a book chapter in the edited volume by Albert Zomaya, entitled "Handbook of Innovative Computing," Springer-Verlag (USA), 2006.
- [2] C. Constantinescu, "Trends and Challenges in VLSI Circuit Reliability," IEEE Micro, vol. 23, pp. 14–19, Jul.–Aug. 2003.
- [3] S. Roy and V. Beiu, "Majority multiplexing—Economical redundant fault-tolerant design for nano architectures," IEEE Trans. Nanotechnol., vol. 4, no. 4, pp. 441–451, Jul. 2005
- [4] D. Bhaduri and S. K. Shukla, Nanoprism, "A tool for evaluating granularity vs. reliability trade-offs in nano architectures," in GLSVLSI, ACM, Boston, MA, April 2004
- [5] S. K. Shukla, Gethin Norman, David Parker and Marta Kwiatkowska, "Evaluating the Reliability of Defect-Tolerant Architectures for Nanotechnology with Probabilistic Model Checking," In proceedings of the International Conference on VLSI design 2004.
- [6] P.T. Gauehan, B.V. Dao, S. Yalamanchili, D. E. Schimmet, "Distributed, Deadlock-Free Routing in Faulty, Pipelined Direct Interconnection Networks," IEEE Transactions on Computers, vol.6, pp, 651-665, 1996.
- [7] B. Almohammand, B. Bose. "Fault-Tolerant communication algorithms in toroidal networks," IEEE Transaction on Parallel and Distributed System. vol. 10, pp.976-983, 1999.
- [8] V.P. Roychowdhury, D.B. Janes, S. Bandyopadhyay and X. Wang, "Collective computational activity in self-assembled arrays of quantum dots: a novel neuromorphic architecture for nanoelectronics," IEEE Transactions on Electron Devices 43, 1996
- [9] R. M.P. Rad, M. Tehranipoor, "A Reconfiguration-based Defect Tolerance Method for Nanoscale Devices," 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06), pp. 107-118, 2006.
- [10] K. Nikolic, A. Sadek, M. Forshaw, "Fault-tolerant techniques for nanocomputers," Nanotechnology, Volume 13, Number 3, pp. 357-362(6), 2002.
- [11] J. Byunghyun, Y. Kim, F. Lombardi, "Error Tolerance of DNA Self-Assembly by Monomer Concentration Control," 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06), pp. 89-97, 2006.
- [12] Jie Han , Jianbo Gao , Yan Qi , Pieter Jonker , Jose A. B. Fortes, Toward Hardware-Redundant, Fault-Tolerant Logic for Nanoelectronics, IEEE Design & Test, v.22 n.4, p.328-339, July 2005
- [13] <http://www.extremetech.com/article2/0,1558,1826021,00.as>
- [14] D.P. Vasudevan, P.K. Lala and J.P. Parkerson, "Fault Tolerant Quantum Computation with new Reversible Gate," Proceedings of the NSTI Nanotechnology Conference, pp 744 – 747, 2005.
- [15] Y. Yellambalase, M. Choi, and Y. Kim, "Inherited Redundancy and Configurability Utilization for Repairing Nanowire Crossbars with Clustered Defects," Proceedings of the 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06), pp 98-106, 2006.
- [16] M. M. Eshaghian-Wilner, A. Khitun, S. Navab, and K. L. Wang, "A Nanoscale Crossbar with Spin Waves," Proceedings of the 6th IEEE Conference on Nanotechnology, Ohio, USA, July, 2006.
- [17] M. M. Eshaghian-Wilner., A. Khitun, S. Navab, and K.L. Wang, "A Nano-Scale Reconfigurable Mesh with Spin Waves," The ACM International Conference on Computing Frontiers, Italy, May 2006.
- [18] M. M. Eshaghian-Wilner., A. Khitun, S. Navab, and K.L. Wang, "Nano-Scale Modules with Spin-Wave Inter-communications for Integrated Circuits," The NSTI Nanotech 2006, Boston, MA, May 2006.
- [19] A. Khitun, and K. L. Wang, "Nano Scale Computational Architectures with Spin-wave Bus, Superlattices & Microstructures. 38(3), 184-200, 2005.
- [20] R.C. Hansen, Ed., "Significant Phased Array Papers," Artech House, Norwood, MA, 1973.
- [21] G. Wang, J. Chen, "A new fault-tolerant routing scheme for 2-dimensional mesh networks," Proceedings of the Fourth International Conference on Parallel and Distributed Computing, 2003.
- [22] X. Fan, W. Moore, C. Hora, M. Konijnenburg, G. Gronthoud, "A gate-level method for transistor-level bridging fault diagnosis," 24th IEEE Proceedings of VLSI Test Symposium, 2006.
- [23] M. B. Tahoori, S. Mitra, "Fault Detection and Diagnosis Techniques for Molecular Computing", In NanoTech Conference, 2004.
- [24] J. Zhou, F.Lau, "Adaptive Fault-Tolerant Wormhole Routing with Two Virtual Channels in 2D Meshes ," International Symposium on Parallel Architectures, Algorithms and Networks, p.142, 2004