

Interconnect Architecture Exploration for Low-Energy Reconfigurable Single-Chip DSPs

Hui Zhang, Marlene Wan, Varghese George, Jan Rabaey
University of California at Berkeley

ABSTRACT

In this paper, we present and analyze a number of interconnect architectures for reconfigurable systems targeting applications in the areas of wireless communication and multimedia processing. Several interconnect architectures suitable for heterogeneous elements are proposed and then a methodology to evaluate the architectures is described. The results indicate that the hierarchical generalized mesh structure shows the most promise in terms of energy efficiency, as it can optimize both local and global connections.

1. INTRODUCTION

Combining flexibility, performance and energy-efficiency is an important design aspect of future system-on-a-chip implementation platforms, targeting applications in the areas of wireless communication and multimedia processing. Reconfigurable systems present an attractive solution, offering both high-performance [5] and energy-efficiency [2] in a domain-specific DSP context. The Pleiades architecture [1] (shown in Figure 1) presents such a reconfigurable solution by combining pre-designed computation modules of different programming granularity (microprocessor, ASIC, FPGA, embedded memory) through a reconfigurable network. The interconnect network is the key component for providing flexibility in such reconfigurable systems. Yet, the high energy consumption of interconnect in related implementations (e.g. FPGA architectures [7]) reminds us that flexibility often comes at the cost of high energy consumption. Therefore, careful design of the interconnect architecture is needed to ensure both low energy and flexibility. The work presented here is the first work (to the authors' knowledge) that provides a comparison and analysis of reconfigurable interconnect architectures using energy as a metric.

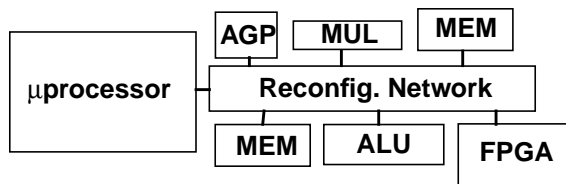


FIGURE 1. An Architectural template for reconfigurable DSPs

In the remainder of this paper, we will first introduce the concept of reconfiguration. Next, various reconfigurable interconnect schemes and their implementations are presented. We then introduce the concept of heterogeneity and its repercussions on the interconnect requirements. The core of the paper is formed by a detailed comparison of the various reconfigurable architectures, based on a well-defined evaluation methodology. Finally, we draw some conclusions and discuss future work.

2. RECONFIGURATION

The main idea behind (re)configurable computing is to build a computational engine through a spatially-programmed connection of processing elements. The interconnect model to be supported by such a system is depicted in Fig. 2. Each bar on the time-axis represents a set of inter-module connections that has to be realized simultaneously. Note that a configuration period can vary from one single clock cycle, over a single duration-limited function, to the entire duration of an application in a multi-function system [6]. The reconfiguration requires the underlying interconnect architecture to support sufficient concurrency within a configuration, while allowing for time-sharing of resources across configurations. An efficient network hence must have sufficient flexibility to support the required interconnect patterns, while still maintaining good performance and efficiency for each configuration.

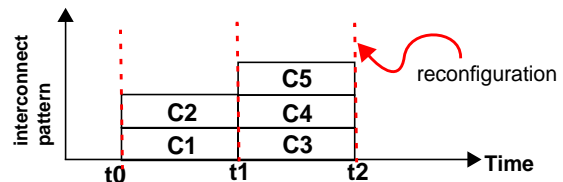


FIGURE 2. Model of Reconfiguration

3. RECONFIGURABLE INTERCONNECT ARCHITECTURES

Having in mind the reconfiguration paradigm presented in the previous section, we now study and extend some existing interconnect architectures for heterogeneous systems. We classify the architectures under investigation into two broad categories. The architectures in the first category (*global interconnect networks*) provide routes with identical costs for all connections between any pair of modules. Many

(DSP) algorithms however display locality and regularity in the required interconnect patterns. In order to take advantage of the spatial locality of the connections, we also investigate architectures that can optimize the costs of local connections. We classify these architectures as *local interconnect networks*.

3.1 Global Interconnect Networks

3.1.1 Crossbar

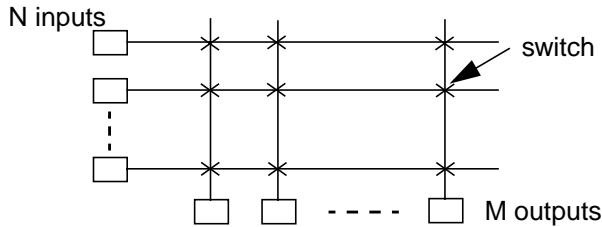


FIGURE 3. Conceptual crossbar network

A crossbar interconnect network allows simultaneous connections from any input port to any output port. Figure 3 shows a conceptual view of such a network. One possible implementation of the crossbar network is to assign a global bus to each input, and each of these busses can be connected to any outputs through a switch. Observe that such a network requires only one switching stage, that is, every input and output pair is connected through a single switching element. This architecture provides full connection flexibility, but suffers from a large area overhead (wasting wire resources) and a high energy consumption (due to the long global buses and the large number of switches).

3.1.2 Multi-stage Interconnect Networks

To reduce the number of switches while still providing the full connection flexibility, a variety of multistage interconnection networks have been proposed in the literature [9]. In general, the number of switches can be reduced from $O(N^2)$ to $O(N \log N)$, while the number of buses is still $O(N)$. Implementing such a multi-stage network on a chip is not straightforward, as these networks require intricate interconnect patterns between the switches. One possible solution is to provide a single, centralized multi-stage switch box with its inputs and outputs connected to the respective ports of the modules. This certainly results in a global interconnect network, and the routing between switchbox and modules becomes prohibitively expensive and difficult.

3.1.3 Multiple-Bus Network

Another way to improve upon the crossbar network is to reduce the number of buses to the maximum number of interconnections B that has to be concurrently supported at

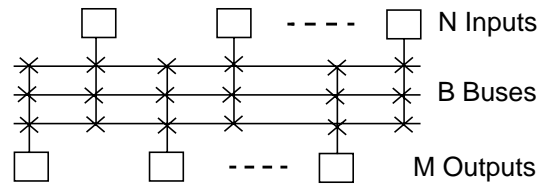


FIGURE 4. Multiple-bus network

any time over all required configurations. An example of such an $N \times B \times M$ “multiple-bus” interconnect network is depicted in Fig. 4.

3.1.4 Summary and Comparison of Global Interconnect Networks

Table 1 summarizes the features of the three presented global interconnection networks. We use an $N \times N$ Omega network [9] as an example of a multistage interconnect network.

Table 1: Comparison of Global Interconnect Networks

	Crossbar	Multistage (Omega: $N=M=2^k$)	Multiple-Bus
Total No. of switches	$N * M$	$2 * N * k$	$B(N+M)$
No. of buses	N	N	B
Loads per connection	M	$2 * k$	$N+M$

The first two rows correspond to the hardware cost of each architecture. The “loads per connection” (3rd row) is defined as the total number of switches connected to each bus. Overall, global interconnection networks tend to present a major hardware and energy cost for systems with large N and M .

3.2 Generalized Mesh

More energy-efficient architectures can be conceived by taking advantage of the locality of most of the connections. In the multi-processor world, a plethora of network topologies [9] have been proposed that provide point-to-point links between neighboring network nodes at the expense of connections between distant nodes. Examples are the ring, the 2-D mesh, the generalized hypercube, and the torus. The mesh structure of the interconnect network utilized in most contemporary FPGAs, is an example of how these network concepts can be applied to the single-chip reconfigurable interconnect problem. Fig. 5 shows a simple FPGA interconnect architecture. It consists of switch matrices with limited connectivity (also called switch-boxes) connected by wiring channels. C(onnexion)-boxes provide the connec-

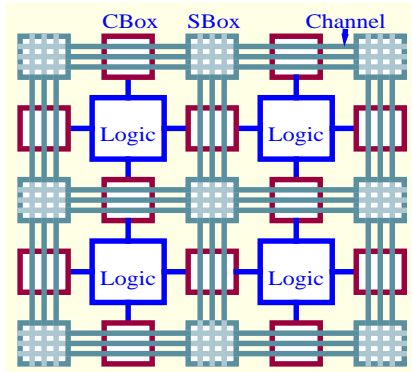


FIGURE 5. Simple FPGA switchbox mesh structure

tions to the ports of the computational elements. The mesh network has the advantage that local interconnections are implemented efficiently due to the segmented nature of the network: only the segments really necessary to provide a connection are activated.

While attractive due to its simplicity, the mesh network is not directly applicable to reconfigurable systems in the style of Figure 1. Since the system is composed of heterogeneous modules of different shapes and sizes, no regular 2-D grid in the style of Figure 5 can be found. We therefore propose a “generalized mesh” structure, shown conceptually in Fig. 6. Given a placement of the modules, wiring channels are created along the sides of each module. Wherever channels meet or cross, a switchbox is provided. The design parameters of such a network are the number of buses in each channel, which can be optimized individually according to the connectivity requirements, and the connectivity provided by the S-boxes.

3.3 Hierarchical Interconnect Networks

While being very effective for local interconnections, the mesh network has the disadvantage that distant communica-

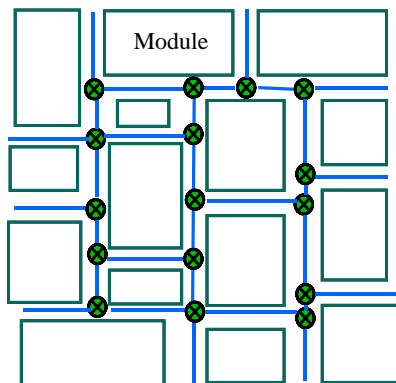


FIGURE 6. Generalized mesh structure

tions become slow and expensive since a large number of programmable switching elements (proportional to the Manhattan distance between the source and the destination) have to be traversed.

This leads to the concept of the hierarchical interconnect network, which continues to exploit locality while reducing the cost of the global connections. In this section, we will first introduce two homogeneous hierarchical interconnect network architectures. A hierarchical interconnect network for heterogeneous systems will then be derived.

3.3.1 Hierarchical Segmented Interconnect Network

Fig. 7 demonstrates the concept of segmentation hierarchy. While a traditional mesh is segmented with a granularity of single cell, a larger granularity of segmentation can be easily envisioned with channel segments spanning more than one logic block (and hence requiring less switching elements per connection). For instance, Figure 7 shows both a single and a double-segmented mesh structure overlaid on top of each other. Based on the required connectivity, nets can be routed using various channel segments to enhance both the energy efficiency and resource utilization. The Xilinx 4000-series FPGAs [11] present a typical example of such a hierarchical segmented interconnect network.

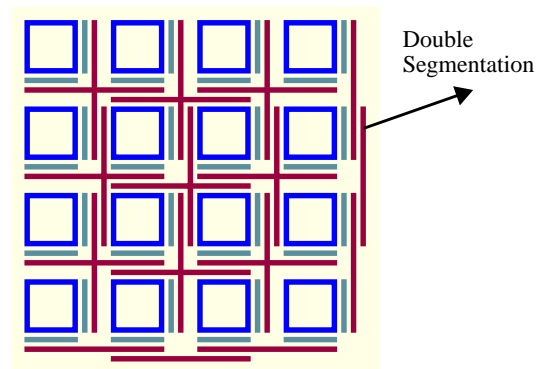


FIGURE 7. Segmented hierarchical mesh structure

3.3.2 Fat-Tree Interconnect Network

An alternative to the segmented hierarchical mesh is presented by the “fat tree”, which is a network based on a complete (binary) tree (see Fig. 8). Logic blocks are located at the leaves of the network tree. The number of buses per channel increases as going up the tree. An example of such a network can be found in the CM-5 supercomputer [8] and the Hierarchical Synchronous Reconfigurable Array used in the BRASS project [4].

3.3.3 Hierarchical Generalized Mesh

Due to the irregularity of the placement and the different

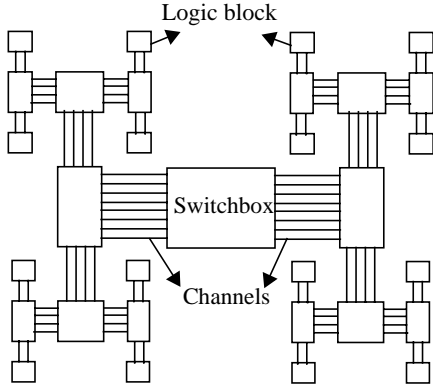


FIGURE 8. Fat-tree interconnect network

interface connectivity of the modules, it is hard to apply the concepts of segmented hierarchy and fat-tree directly to heterogeneous reconfigurable systems. To effectively utilize this approach requires a knowledge of the application domain and its associated benchmark sets. Based on interconnect requirements extracted from these applications, one can partition the modules into clusters of tightly-connected components (Fig. 9). Within each cluster, a generalized mesh network provides the intra-cluster connections. A second larger-granularity mesh is superimposed on top of this local network to provide energy-efficient and high-performance inter-cluster connections. Obviously, this architecture can be extended with more levels of hierarchy if needed.

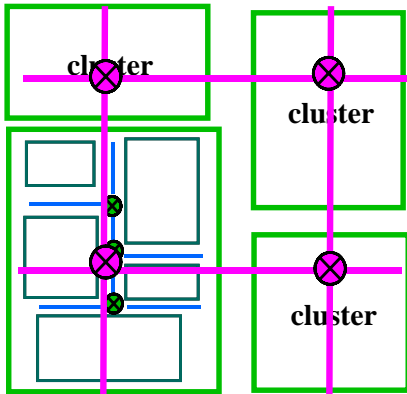


FIGURE 9. Hierarchical generalized mesh

Compared with the global interconnection networks presented earlier, this approach trades off flexibility for energy-efficiency. One major challenge in the design of the hierarchical network is that the interface points between the different hierarchy levels are prone to introduce connectivity bottlenecks. The clustering process is hence one of the most crucial steps in the design of such a network. Other implementation issues to be considered are the placement of the

heterogeneous modules to minimize the chip area, the location of the cluster interface ports, and the routing of the global mesh network.

4. EVALUATIONS OF RECONFIGURABLE INTERCONNECT ARCHITECTURES

The generalized mesh and hierarchical generalized mesh networks seem to show some promise for heterogeneous reconfigurable systems. To evaluate their potential, we will compare them with a global interconnect architecture, being the multiple-bus network. Global interconnect structures are attractive because of their simplicity and easy of programming. Domain-specific architectures exploiting locality have to yield dramatic improvements in performance with respect to these generic structures to be of interest. Incremental gains in either energy or speed are not worth the loss in flexibility and the more complex design process.

In order to compare these three architectures, a bottom-up evaluation methodology is developed based on a combination of accurate physical models with an architecture-independent router.

4.1 Energy and Delay Model

First, we derive analytical models for both the energy-consumption and network-delay for every architecture under evaluation.

An universal energy model for the interconnect network is easily derived (assuming that the wires are switching from rail-to-rail):

$$E = [C(wire) + C(switches)]V^2 = K_L \cdot L + K_{SW} \cdot W_{SW} \cdot N_{SW}$$

where K_L (PJ/mm) is the energy per unit wire length, and K_{SW} (PJ/um) is the energy per unit switch-width. W_{SW} represents the switch width, and N_{SW} is the number of switches.

We use the Elmore model to estimate the delays. For the multiple-bus network, we assume that the switches are evenly distributed along the wire:

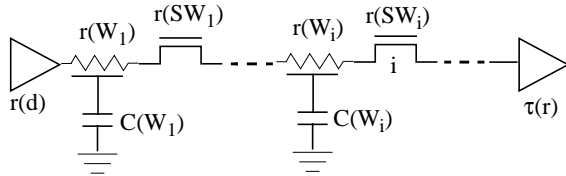
$$t_p(multibus) = r(in)C_{total} + 0.5 \times r(wire)C_{total}$$

where $r(in)$ is the resistance of the input switch, C_{total} is the total capacitance due to wire and switches, and $r(wire)$ is the wire resistance.

For the generalized mesh or hierarchical mesh structures, the Elmore delay model results in the following equation,

$$t_p(mesh) = r(d)C_{total} + \sum_{i=1} \{r(W_i) \cdot [0.5C(W_i) + C(i)] + r(SW_i)C(i)\} + \tau(r)$$

where the symbols are explained in Fig. 10.



- $r(d)$: resistance of the driver
- $r(W_i)$: resistance of the i th wire segment
- $C(W_i)$: capacitance of the i th wire segment
- $r(SW_i)$: resistance of the i th switch
- $C(i)$: subtotal capacitance after the i th switch
- $\tau(r)$: delay of the receiver

FIGURE 10. Wire model of interconnect with switching element

Using the analytical models presented above, parameterized energy and delay models as a function of switch size, wire length, and number of switches are derived for each of the architectures of interest.

4.2 Evaluation Methodology

To evaluate the interconnect architectures, we use the Maia chip, currently under development at UC Berkeley, as a case study. Maia, one instance of the Pleiades architectures, is targeted towards the baseband voice coding domain, and combines the following modules: 2 MACs, 8 memories and address generators, 2 ALUs, 4 I/O Ports, an embedded FPGA, and an ARM microprocessor. The chip is implemented using the SGS-Thomson 0.25 μm 6-metal CMOS process. Over-the-cell routing is not considered to simplify the wire-length estimation (and is of limited use anyhow for most of the modules).

The implementation parameters (such as the number of buses) were optimized against the flexibility requirement using a set of benchmark examples representing one specific application domain. In this case study, the benchmarks represented various computational kernels occurring in speech-coding applications, such as the dot_product, IIR, vector sum with scalar multiply, etc. [10].

4.2.1 Multiple bus network

To estimate the performance and energy of the architecture,

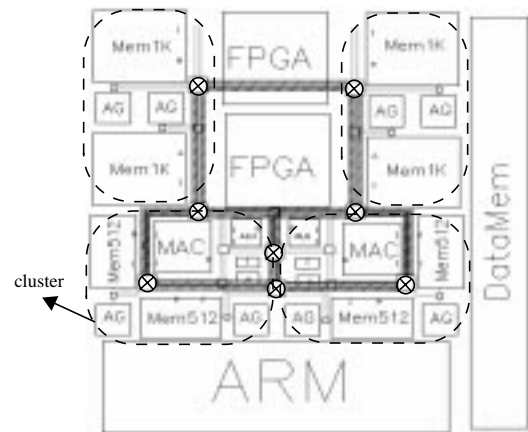
we need to know the loads-per-connection (from an input to an output) and the wire length of each global connection. The loads per connection can be easily analyzed and are shown in row three of Table 1. For the Maia chip, $N=24$, $M=30$ and $B=14$. The main difficulty to evaluate the interconnect energy consumption is to estimate the wire length, which is dependent on many factors such as module sizes, bus routing, bus width and number of buses. But by intuition, one can surmise that the global-bus length is proportional to the chip diameter, or at least to $O(\sqrt{N})$. To produce a more accurate estimate of the bus length, a chip floorplan assuming a multiple bus-network was drawn. Next, the switches were carefully sized to meet the timing constraints.

4.2.2 Local interconnect networks

The mesh and hierarchical mesh architectures were implemented based on the concepts introduced in Section 3.2 and Section 3.3.3.



(a) Flat generalized mesh architecture



□ lower-level switchbox ⊗ hierarchical switchbox
(b) Hierarchical 2-level generalized mesh architecture

FIGURE 11. Maia chip floorplan with interconnect network

Fig. 11(a) shows an implementation of the generalized mesh interconnect network for the Maia chip. The number of buses per channel is set to meet the routing requirements of all the computation kernels to be supported (4 in this case). Fig. 12(b) shows an implementation of the hierarchical 2-level generalized mesh interconnect network. Four clusters of tightly connected modules are formed according to the kernel benchmark set to minimize the global connections. Each cluster has a local mesh (with 2 buses per channel) for intra-cluster connections, and some interface ports for inter-cluster connections. The interface ports are implemented as hierarchical switchboxes which can provide connectivity within each mesh and across two different layers of meshes. On top of local meshes, a second larger-granularity mesh (appears in bold in the graph) which also has 2 buses per channel connects the hierarchical switchboxes together.

Some important implementation steps (such as placing modules, clustering, and locating cluster interface ports) were done manually, and a detailed methodology of optimization will be outlined in future work. All the implementation parameters are optimized iteratively with the help of a graph-based router which can route each net on the underlying network architecture to predict the energy and delay data. In the router, each interconnect network is described as a graph, G , with the vertices representing the ports and switches and edges presenting feasible connections. Finding an optimal route between two ports is formulated as finding the minimum Steiner tree on the graph and a heuristic algorithm proposed by [3] is employed to solve this NP-complete problem. The model obtained in section 4.1 is used to compute the weights for each connection. The final cost of a net is computed as the total cost of the segments on the route.

4.3 Results

The results of the architecture evaluation are presented in two ways. The first method credits the architectures that optimize both local and global connections by looking at the statistics of energy consumption of all the module-to-module connections. The second method shows how each architecture performs for the set of benchmarks defined above (Section 4.2.2). To ensure a fair energy comparison, a timing constraint of 10 nsec is set on the worst-case module-to-module delay for all the interconnect networks. The switches are sized according to this timing constraint and then used in the energy model.

4.3.1 Module-to-module metric

A good interconnect network should provide low-cost connections for modules close to each other while keeping the cost of global connections to a minimum. To quantify this

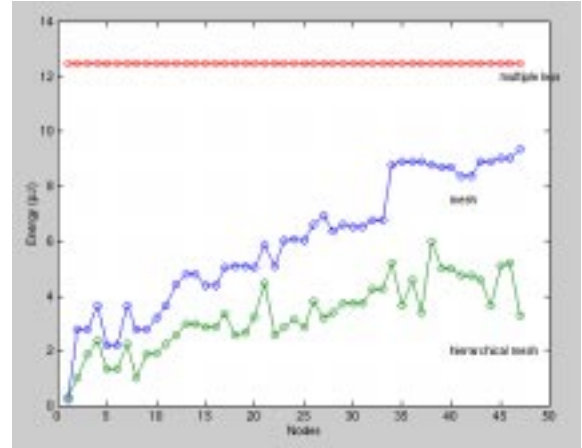


FIGURE 12. Module-to-module energy of different architectures.

feature, a reference module is first picked and the rest of the modules are numbered according to their manhattan distances from the reference module. In this context, a node with a lower number represents a “local” connection and vice versa. For each module, the energy of the shortest route from the reference module (assuming independence between every route) is calculated and plotted. This module to module cost delineates the lower bound on the cost of each connection and the results are plotted in Fig. 12 for all modules. As expected, the hierarchical mesh network performs the best for both global and local interconnections. At the best case, the energy of a net in the hierarchical mesh is about one tenth of the net cost in the multiple bus network; at the worst case, the cost only goes up to about one half of the multiple bus cost.

4.3.2 Benchmark results

The module-to-module metric presents an evaluation of the optimal performance of each net. However, this optimal module-to-module cost cannot be guaranteed because there are many other factors such as the different physical mappings of each kernel and the routability due to channel congestion, etc.. Therefore, we have also evaluated the architectures over the given benchmark set. For each benchmark, many physical bindings were considered. For the multiple-bus network, the same energy data is obtained for all the bindings because it does not exploit the locality of connections. For local interconnect networks, the ‘best’ binding (as used in Table 2) should possess more local connections and the ‘worst’ bindings contains many global connections. Fig. 13 shows an example of the best and worst bindings for the dot_product kernel.

The results for some of the benchmarks are shown in Table 2 and they show the same tendency as in Fig. 12 with the module-to-module metric — the hierarchical intercon-

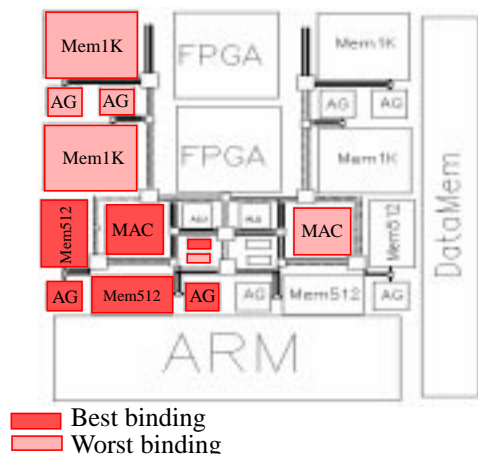


FIGURE 13. Best and worst bindings of the dot_product kernel

nect has the lowest energy consumption. For example, for the “dot-product” benchmark, the best mapping on a hierarchical mesh consumes less than one tenth of the energy of multiple bus network; even the worst mapping consumes less than one fourth.

Table 2: Benchmark results

		dot_product	vector sum w/ scalar mult.	IIR
Multi-bus		50	50	138
Mesh	<i>Best</i>	8.7	5.2	24.6
	<i>Worst</i>	17.7	14.7	43.4
H. Mesh	<i>Best</i>	4.7	3.8	18.8
	<i>Worst</i>	11.1	10.2	31.3

5. CONCLUSION AND FUTURE WORK

From the analysis of various architectures and experimental results of our case study, we found that the hierarchical interconnect network delivers the best energy efficiency while maintaining flexibility for heterogeneous reconfigurable systems. The results show that a hierarchical interconnect network reduces the energy consumption by 25% to 50% over a flat mesh interconnect network. Overall, the best results from local interconnect networks present an energy-efficiency which is 5 to 10 times higher compared to global interconnect networks.

This work presents a possible implementation of a hierarchical interconnect network for heterogeneous elements, by overlaying various levels of generalized mesh structures.

Future work is to investigate other hierarchical interconnect architectures which make better utilization of opportunities offered by the available metal layers. Further efforts with respect to the analytical modeling of hierarchical and heterogeneous networks are also necessary.

6. REFERENCES

- [1] A. Abnous and J. Rabaey, “Ultra-Low-Power Domain-Specific Multimedia Processors”, *Proceedings of the IEEE VLSI Signal Processing Workshop*, San Francisco, California, USA, October 1996.
- [2] A. Abnous et al., “Evaluation of a Low-Power Reconfigurable DSP Architecture”, *Proceedings of the Reconfigurable Architectures Workshop*, Orlando, Florida, USA, March 1998.
- [3] M. J. Alexander, et al. “Performance-Oriented Placement and Routing for Field Programmable Gate Arrays”, *Proceedings of EURO-DAC*, European Design Automation Conference, Brighton, UK, 18-22 Sept. 1995.
- [4] BRASS research homepage: <http://HTTP.CS.Berkeley.EDU/Research/Projects/brass/>
- [5] G. R. Goslin, “A Guide to Using Field Programmable Gate Arrays for Application-Specific Digital Signal Processing Performance”, *Proceedings of SPIE*, vol. 2914, p321-331.
- [6] A. Kalavade and P. A. Subrahmayan, “Hardware/Software Partitioning of Multi-function Systems”, *Proceedings of IEEE International Conference on Computer Aided Design*, San Jose, CA, USA, 9-13 Nov. 1997.
- [7] E. Kusse, “Analysis and Circuit Design for Low Power Programmable Logic Modules”, Master’s Thesis, UC Berkeley, 1997.
- [8] Charles E. Leiserson et al., “The Network Architecture of the Connection Machine CM-5”, *Proceedings of the Fourth Annual ACM Symposium on Parallel Algorithms and Architectures*, 1992.
- [9] Anujan Varma and C. S. Raghavendra, “Interconnection Networks for Multiprocessors and Multicomputers: Theory and Practice”, *IEEE Computer Society Press*, 1994.
- [10] M. Wan, D. Lidsky, Y. Ichikawa, J. Rabaey, “An Energy Conscious Methodology for Early Design Exploration of Heterogeneous DSPS”, *Proceedings of the Custom Integrated Circuit Conference*, Santa Clara, CA, USA, May 1998.
- [11] Xilinx Corporation, “XC4000 Field Programmable Gate Arrays: Programmable Logic Databook”, 1996.