Mapping method of reconfigurable cell matrices based on nanoscale devices using inter-stage fixed interconnection scheme

Pierre-Emmanuel Gaillardon, Fabien Clermidy
CEA – LETI – MINATEC
Grenoble, France
pierre-emmanuel.gaillardon@cea.fr

Ian O’Connor, Junchen Liu, Renaud Daviot
Université Lyon
Lyon, France

Abstract—Emerging devices open the way to build nanoscale logic cells, dedicated to high-density reconfigurable computation. Nevertheless, in an architectural context, fine-grain logic cells integration is limited by traditional interconnection scheme and associated overload. This paper describes an interconnection scheme, based on static and incomplete interconnection topologies. We also propose a method to map functions onto such architectures. Then, to evaluate 4 proposed topologies, we test mapping efficiency and fault tolerance. The analyses show that this approach could improve scalability of traditional FPGAs by a factor of 8.

I. INTRODUCTION

For the last four decades, the semiconductor industry has witnessed an exponential growth in accordance with Moore’s Law. As we advance into the era of nanotechnology, the promise is enormous and semiconductor devices are expected to be scaled down to their physical and economic limits. The International Technology Roadmap for Semiconductors (ITRS) has pointed out significant future intrinsic device hurdles, such as leakage, power or quantum effects. These limitations require the semiconductor industry to explore the use of novel materials and devices able to complement or even replace the CMOS transistor in systems on chip. New solutions have to be proved within the next decade and before silicon based technology reaches its limits around 2020, when the physical channel length of MOSFETs is expected to be below 10 nm. In this context, nanotechnologies could bring advantages, since the nanometric scale increases not only integration density, but also functionality. In nanoscience, research is focused principally on the understanding of new physical phenomena and on device fabrication techniques. However, it is also necessary to work on the design of new circuits. Efficient use of the projected tens of billions of elementary, unreliable, nanometric devices will lead, among others, to the emergence of reconfigurable platforms as the principal computing fabric before the end of the next decade. The reconfigurable approach allows volume manufacturing and reduces the impact of mask costs increase, while demonstrating broad defect tolerance capabilities.

Even if several structures and architectures have been proposed for Beyond CMOS computing [1, 2], an approach using (in particular) 1D-structures for FET channels is preferable for the medium term, due to its compatibility with CMOS computation. While design considerations are quite close to CMOS, these emerging devices bring some novelties. We can consider for example, the logic cell designed by O’Connor et al. [4]. This cell uses recent technological breakthroughs, given by double gate CNTFETs. A double-gate CNTFET, built from a single nanotube with front and back gates controlling the transistor channel transport characteristics, exhibits ambivalence [3]. Ambivalence means that n- and p-type behavior can be observed in the same device depending on the back gate voltage polarity. This feature opens up new opportunities for using CNTFETs in logic circuits. The proposed dynamically reconfigurable logic cell, called DRLC_7T [4], uses only 7 transistors and can be configured to any one of fourteen basic binary operation modes, through three ternary control signals (-1V, 0V, +1V). Integration density and unreliability drive us to consider the architectural question specifically. In this paper, we propose an organization of reconfigurable cells in future system.

This paper begins by describing an architecture based on cell matrices. Then, a method to map function graphs to matrices is presented. Finally, we apply the method to various intra-matrix interconnection topologies in order to evaluate the most suitable one in terms of different metrics.

II. CELL MATRICES AND INTERCONNECTION TOPOLOGIES

Fine-grain reconfigurable gates open the way towards high density programmable structures. In a conventional approach, each calculation cell would be directly connected to switchboxes, as shown in figure 1. A conventional approach allows complete connectivity, which is useful for complex interconnect and heavy-duty logic blocks. In FPGA architectures, such a solution is used to interconnect each CLB (Configurable Logic Block) to others. However, in the case of fine-grain logic cells, this approach would lead to a loss of efficiency due to a large overhead in terms of device complexity. In the case of DRLC_7T, 7 transistors are used in the cell, while a similar number of transistors (at least 6) are used for a 1-bit switchbox.

In order to avoid this overhead problem, we propose a cluster-based approach as shown in figure 2, which consists of assembling cells in a matrix pattern, with the use of fixed intra-matrix interconnect between layers of cells. Switchboxes are used for intra-matrix interconnect.
III. FUNCTION MAPPING METHOD

While the application is quite close to logic synthesis and network routing, the fact that we have introduced computing inside the matrix means that we cannot use routing algorithms or synthesis algorithms directly. We present in this section a mapping method designed to map a logic function graph onto the architecture described above. As an example, we consider a matrix which is 4 cells deep and 4 cells wide (4d4w) using a Banyan interconnect topology.

The function to map is represented by a graph (an example is shown in figure 4), generated by a random graph generator for test purposes. In the adjacency matrix (shown in figure 4), (i, j) refers to the intersection of the row i and column j. A 1 at the position (i, j) means that the point i is connected to the point j. This matrix is essential to subsequent processing steps.

In a first approach, we have to adapt the logic function to our architecture. Due to the layered structure, the system is pipelined, which makes adding synchronization elements necessary. Synchronizations are required to constraint graphs to the fixed structure, for data propagation over logic layers. Thus, in the function graph, we identify logic layers and we divide the adjacency matrix into 4x4 small matrices according to the different layers as shown in figure 4. Among the 16 matrices, $C_{01}$, $C_{02}$, $C_{03}$, $C_{12}$, $C_{13}$, $C_{23}$ are 6 matrices containing 1’s to be considered. $C_{\text{int}}$ is the adjacency matrix between the points in the logic layer $n$ and those in the logic layer $m$. We therefore pay particular attention to $C_{02}$, $C_{03}$, $C_{13}$ (where $m\geq n+1$) because the presence of any non-zero element in these three matrices identifies a connection which “jumps” at least one logic layer. Such a direct connection cannot be realized in the topology since the interconnect topology is physically fixed. In our example, the connections between points $(2, 6), (5, 8)$ and $(4, 9)$ are the three connections to be adjusted. The solution is to add synchronization elements, then repeat the process until no more connections jump logic layers. In our example, 3 points are added into the graph to introduce synchronizations and 3 others to extend inputs and outputs.

Then, in a second phase, each point is assigned to the cells, with respect to the interconnect topology. The process starts from the first layer. The function graph is firstly explored recursively to find the sequence of the point assignments. Then, each layer’s connections are then compared to the relevant inter-layer connectivity matrix - this allows the assignment of functions to cells. In the example, the first point $p_1$ is assigned to the cell $l^0$. According to the path defined in the previous step, $p_4$ is the next point to assign to a cell in the matrix. Since $l^0$ is physically connected to $l^3$ and $l^4$, the cell
with lower γ-index (here \( t^\text{th} \)) is arbitrarily chosen for p4 assignment, and the other possibility is memorized. Branching (i.e. the exploration of the immediately preceding alternative) is used when the arbitrary choice leads to a dead-end, and the process is repeated until all functions are assigned to cells. In our example, the final programmed matrix is shown in figure 5.

![Matrix after function mapping](image)

Figure 5: Matrix after function mapping

The algorithm has been implemented in Matlab, and executes in under 0.1s for a complete 4d4w mapping operation using a standard 2GHz PC. While this approach enables the mapping of simple functions to the fixed-interconnected matrix based on the reconfigurable cells, function partitioning and merging methods will be required to map more complex functions over several matrices.

IV. EVALUATION OF INTERCONNECT TOPOLOGIES

The aim of this part of our work is to evaluate and compare performance metrics for the 4 interconnect topologies. Our study is made on a 4d4w matrix using intra-matrix interconnection topologies, mentioned previously. 4d4w matrices have been chosen because of a good balance between complexity and simulation time. To compare, we use, as metrics, the success rate of mapping function graphs and the fault tolerance.

We carried out detailed analyses to compare the efficiency of the different intra-matrix interconnect topologies. We use a random graph generator to generate static sets of function graphs containing 6-16 points, in order to have fixed comparison criteria between topologies. Each set, corresponding to a given number of points in the function graph, contains 1000 samples. Using the previously described mapping method, each function is programmed onto the 4d4w matrix using the various intra-matrix interconnect topologies, ideal or faulty, and metrics are calculated. Figure 6 summarizes the evaluation methodology and the associated parameters.

A. Mapping success rate

Applying static sets to ideal interconnect topologies, we can test the ability of the matrix-topology ensemble to have complex functions mapped onto it. Considering the percentage of function graphs successfully mapped onto matrix with respect to the number of samples in set, we obtained the success rate. Figure 7 shows the comparison of success rates for 4d4w Banyan, Omega, Flip and Baseline topologies. For Banyan, Flip and Baseline interconnect topologies, the success rate is about 80% when the function graphs have 6 points. At 12 points, the success rate is about 25%. The difference between these two topologies is thus relatively small. However for the Omega interconnect topology, the success rate is about 90% for 6-point function graphs and about 40% for 12-point graphs. This clearly shows that the Omega interconnect topology is more suitable for this type of matrix.

This is because this topology is less symmetric than the other topologies and spreads calculations over cells occupying less width, which seems to correspond better to typical function graphs. In fact in the matrix, there are pairs of cells which have the same inputs. For two cells which have the same inputs, the sum of the number of function they can achieve is 14. For two cells which do not have the same inputs, the sum of the number of functions they can implement is 14+14 = 28. In the Banyan topology for example, there are 6 pairs of cells which have the same inputs, while in the Omega topology, there are only 2. This is the main reason why the Omega topology has the potential to realize more functions than other topologies.

B. Fault tolerance

Due to the chemical processes used for building devices in a bottom-up approach, the latter will have significantly lower yields than those obtained via current fabrication practices, resulting in aggregates with high defect rates. Considering transistors based on CNT, defects introduced by the CNT synthesis process could impact the behavior of the CNTFET [6]. It is also necessary to consider that traditional processes have to shrink to build such devices. Due to variability, overall
system performance is likely to be dominated by unreliability as concerns individual device characteristics. For example, it is reasonable to consider that links in interconnection layers could be destroyed or stuck in a Boolean state. Typically, unreliable systems use defect-avoidance techniques [7], which are based on defect detection and adaptive mapping, to increase their reliability. Detection and adaptive mapping are out of the scope of this paper. To study fault tolerances in matrices, we introduce defects in interconnection layers, which will force the mapping method to work around them, simulating the behavior we could achieve with a defect avoidance technique. With these considerations, physical interconnection defects were simulated by randomly deleting links in layers. Cell non-functionality is simulated by deleting all related input/output links. Using the Banyan topology as a reference, figure 8 shows the comparison of success rates for 4d4w Omega and Banyan topologies in the cases of one or two faulty links on the first layer and one faulty cell, which are the most representative cases. For all topologies, when the function graphs have 6 points or 16 points, faults have no influence on the success rate, because the interconnect topologies are not a decisive factor at these limits. At 12 points, the success rate for the Omega topology falls faster than for the Banyan topology. The Banyan topology is more robust than the Omega topology, but in all situations, the success rate for the Omega topology is higher than that of the Banyan topology.

C. Discussion

Results, obtained above, are summarized in Table I. Values are normalized considering the use of Banyan topology for a 12-node graph as a reference. The Omega topology gives the best results in terms of success rate and fault tolerance. The three others topologies analyzed are quite similar in results. As mentioned above, these results might be explained by the symmetry of the interconnection topologies. In fact, highly symmetric topologies will give good results in terms of fault tolerance, while their mapping success rate is less convincing. Another fundamental point to consider is the gain compared to a classical CMOS-FPGA. In FPGA systems, it is recognized that it is systematically possible to place and route designs up to a 70%-80% filling rate, with automatic CAD tools. Comparing this to our matrix with the Omega topology, Figure 7 gives, for 75% filling (i.e. 12 operators for 4d4w), a success rate of about 40%. However, given that the area of DRLC. 7T cells has been estimated to be 20x less than FPGA cells [8], the overall scalability of this approach can be expected to better conventional solutions by a factor of 8 (20x40%).

<table>
<thead>
<tr>
<th>Topology</th>
<th>Mapping efficiency</th>
<th>Fault tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banyan</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.84</td>
<td>0.86</td>
</tr>
<tr>
<td>Flip</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Omega</td>
<td>1.65</td>
<td>1.42</td>
</tr>
</tbody>
</table>

V. Conclusion

This paper has firstly introduced a cluster-based architecture useful for fine-grain reconfigurable logic cells based on emerging devices. This cluster-based architecture uses fixed interconnection topologies in order to reduce the overhead induced by conventional approaches. We have proposed a method to map specific functions to the matrices of reconfigurable cells. This method has been used to analyze intra-matrix topologies with regard to various metrics and shown that the mapping success rate is about 90% for 6-point function graphs and about 40% for 12-point graphs when using the Omega interconnect topology in a 4x4 matrix. Moreover, we have shown that Omega interconnect topology is still more robust than other topologies in the presence of defects. A matrix with the Omega topology could expect to improve by 8 the scalability of FPGAs. Future work will focus on algorithm improvement and on topology modifications, such as triangular arrangements of cells, in order to increase and homogenize performances, whereas fabrication considerations have to be explored deeply.

REFERENCES