

# Fault Tolerance of Programmable Switch Blocks

J. Huang, M. B. Tahoori, and F. Lombardi

Department of Electrical and Computer Engineering  
Northeastern University  
Boston, Mass 02115, USA

{hjing,mtahoori,lombardi@ece.neu.edu}

## Abstract

*This paper presents a new approach for the evaluation of FPGA routing resources in the presence of faulty switches. This is considered under the worst case scenario of open faults. Signal routing in the presence of faulty switches is analyzed at switch block level; probabilistic routing (routability) is used as figure of merit for evaluating the interconnect resources of FPGAs. The presented approach utilizes a path-based technique to find the probability of establishing a path between pairs of input and output endpoints in a switch block. The results are reported for various commercial and academic FPGAs.*

## 1. Introduction

Field Programmable Gate Arrays (FPGAs) have been extensively used in many facets of digital design. An FPGA usually consists of a two-dimensional (2-D) array of logic resources connected by routing resources (I/O, interconnect). The interconnect resource includes horizontal and vertical routing channels; programmable switch blocks are present to connect the routing channels and logic resources. The programmable switches of the blocks can be individually programmed (on/off) to establish paths to route signals inside the FPGA. Almost 80% of the FPGA area is devoted to interconnect resources.

The programmability and modularity of an FPGA are readily adaptable to fault tolerance. When faults are present, in some cases it is possible to reconfigure the FPGA so that the desired circuit can still function correctly. The availability and flexibility of the interconnect resources often determine whether it is possible to successfully reconfigure the FPGA with minimum performance penalty. Very little work has been reported under which conditions a certain configuration is realizable in the FPGA when faults are present. This effectively differentiates the switch block structures currently available from different manufacturers and their ability to provide routing when switches are faulty.

In this paper a new method for establishing the fault tolerance of arbitrary structured FPGA switch blocks is proposed. Based on a polynomial-time algorithm, routability is established as a function of the number of possible paths in the switch block.

## 2. Path-Based Method

Here we present a path-based method to evaluate the fault tolerance of a switch block in terms of the number of paths

which can be routed. The switch block is modeled as an unweighted, undirected graph  $G = (V, E)$ ,  $|V| = n$ ,  $|E| = e$ ; each endpoint of the switch block is a vertex in  $G$ ; programmable switches are represented as edges in this graph. Let the minimal degree of the vertices in  $G$  be  $d_{min} = d$ .

Only switch stuck-open faults are considered as representative of current technology as well as the worst case scenario. In a fault free switch block, a number of paths exist between pairs of endpoints for routing signals. If at least one switch is faulty, then some paths may no longer be routed. Therefore the ability to route (routability) is lost due to faults.

The following criterion is proposed to evaluate the routability of a switch block.

**Criterion 1:** *The number of paths (up to a maximum length) from each source vertex  $s$  to all other vertices in a switch block.*

The proposed method can be described as follows. The number of paths of at most length  $d$  from a source endpoint (in the presence of a number of switch faults) is established; this is compared with the number of paths in a fault-free switch block (previously found). As most switch blocks used in FPGAs have a symmetric (homogeneous) internal structure, this process is independent of the source endpoint.

Initially all paths of length up to  $d$  (from a source vertex) are found for a fault free switch block. Let  $T_{all}^l$  be the number of paths of length  $l$  in a fault-free switch block starting from a source vertex. Assume  $k$  switches fail in the switch block, then the number of possible fault patterns are  $N_{fp} = \binom{e}{k}$ . Let the number of paths of length  $l$  for pattern  $i$  be  $T_i^l$  ( $0 \leq i \leq N_{fp}$ ,  $1 \leq l \leq d$ ). The average number of paths of length  $l$  that still exist, is:  $T_{av}^l = \frac{1}{N_{fp}} \sum_{i=1}^{N_{fp}} T_i^l$  ( $1 \leq l \leq d$ ). Routability is defined as the probability of finding a path of length  $l$  in the presence of  $k$  faulty switches ( $1 \leq k \leq e$ ). If the graph is regular then  $d_{min} = d$ . We define the Routability Metric as  $RM_l = \frac{T_{av}^l}{T_{all}^l}$  ( $1 \leq l \leq d$ ).  $RM$  represents the percentage of paths of length  $l$  which can be routed in the presence of one switch fault.

An algorithm is presented based on Depth First Search [2], which is used to find all possible paths of length  $l$  in an unweighted, undirected graph  $G = (V, E)$ , starting from a vertex  $s \in V$  to a target vertex  $t \in V$ . Let the minimal degree of the graph be  $d_{min}$  and the maximum degree be  $d_{max}$ .  $len$  is the length of the path to be found. *Path* contains the current

path, the length of the current path is given by  $k$ . *Starting* is 1 for the top call and 0 otherwise.

```

Function ListPath(s,t,len):
if this is top call then
    initialize Path to be empty, k=0;
    put vertex s to Path
s ← the kth element of Path
if a path with len starting from s ending in t is found then
    store Path;
    return;
if path already has length len but didn't reach vertex t then
    return;
current_vertex ← first vertex in the adjacency list of s;
while there's still vertices in the adjacency list loop
    if current_vertex already in Path and current vertex is not t then
        continue;
    put current_vertex in Path, k++
    call ListPath with s,t,len
    k--;
    remove current_vertex from Path
    current_vertex ← next vertex from the adjacency list.

```

The above algorithm is recursive and its complexity is  $O(n^l)$ . This is an enumeration problem because all paths of length  $l$  (starting from a source vertex in an arbitrary graph  $G$ ) must be found. However, switch blocks are sparsely connected (i.e.  $d_{max} \ll n$ ), so its complexity is  $O(d_{max}^l)$ , which in practice is considerably smaller than  $O(n^l)$ .

### 3. Results

In this section, the routability metric is computed for some commercial FPGA architectures, namely Xilinx XC4000 and Virtex FPGAs, as well as some academic FPGAs [1].

#### 3.1. Xilinx XC4000 and Virtex FPGAs

The Xilinx XC4000 switch block has 8 endpoints in each of the four ports and 48 switches ([4][3]). Table 1 shows the number of paths ( $T_{all}^l$ ) for  $l=1, 2, 3$  starting from a vertex to other vertices in the fault-free switch block. Only non-zero entries are shown. Table 1 shows the results in the presence of one, two, and three switch faults.

The Xilinx Virtex GRM has 24 endpoints in each of the four ports, a total of 96 endpoints and 144 switches [4]. The results for the Virtex is shown in Table 1.

#### 3.2. Academic FPGAs

The universal structure proposed in [1] is considered. An  $8 \times 8$  universal switch block has the same size as the Xilinx XC4000 switch block. The results are shown in Table 1. In order to compare this structure with Virtex switch block structure, a  $24 \times 24$  universal switch block with 96 endpoints and 144 switches, is also considered. The corresponding results are shown in Table 1.

To further investigate the effect of connectivity of the switch block on fault tolerance, a *double-connected* universal switch block (each endpoint has a degree of 6) is considered. The results for an  $8 \times 8$  double-connected switch block with 96 switches are shown in Table 1. For fair comparison, a  $16 \times 16$  XC4000-like switch block is considered; this has the same number of switches as an  $8 \times 8$  double-connected universal switch block (Table 1).

Based on these experiments, it can be concluded that routability based on the presented path-based metric is a function of the number of switches in the switch block, not the

Number of Faults	length 1 RM	length 2 RM	length 3 RM
Xilinx XC4000 Switch Block			
1	98%	96%	93.5%
2	96%	92%	88%
3	94%	88%	82%
Xilinx Virtex Switch Block			
1	99.3%	98.6%	97.9%
2	98.6%	97.2%	95.9%
3	97.9%	95.9%	93.8%
$8 \times 8$ Universal Switch Block			
1	98%	96%	93.8%
2	96%	92%	88%
3	94%	88%	82%
$24 \times 24$ Universal Switch Block			
1	99.3%	98.5%	98%
2	98.6%	97%	96%
3	97.9%	96%	94%
$8 \times 8$ Double-Connected Switch Block			
1	99%	98%	96.9%
2	98%	95.8%	93.8%
3	96.9%	93.8%	90.8%
$16 \times 16$ XC4000-like Switch Block			
1	99%	97.9%	96.9%
2	98%	95.85%	93.8%
3	96.9%	93.8%	90.8%

**Table 1. Switch Block with One, Two, and Three Faulty Switches**

structure or the number of endpoints of the switch block, provided the switch block has a regular structure, i.e. the degree of all endpoints are equal. Note that different switch block structures lead to totally different routing paths in the entire FPGA, but for fault tolerance, only the number of switches is relevant. More details of the path-based method and the results can be found in [3].

### 4. Conclusion

This paper has presented a new criteria for evaluating the fault tolerance of a switch block which is applicable to any arbitrary switch block structure. The impact of switch faults on the probability to route (routability) in a switch block has been assessed. This method has been used for comparing different switch block structures in commercial FPGAs (such as the Xilinx XC4000 and Virtex) as well as academic FPGAs. It has been shown that the routability in the presence of faults increases by choosing a switch block structure with more switches. Also, routability based on a path-based metric is a function of the number of switches, not the structure or connectivity of the switch block.

### References

- [1] Y.W. Chang, D.F. Wong, C.K. Wong, "Universal Switch-Module Design for Symmetric-Array-Based FPGAs", *Proc. of ACM International Symposium on Field Programmable Gate Arrays*, pp. 80-86, 1996.
- [2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein "Introduction to Algorithms", *McGraw-Hill*, pp. 643-693, 2001.
- [3] J. Huang, M.B. Tahoori, F. Lombardi, "Fault Tolerance of Programmable Switch Blocks", *Internal Report, ECE dept. Northeastern University*, 2003.
- [4] Xilinx Corp. "Xilinx Datasheets", *www.xilinx.com*.