A Study On Optimized Technology Mapping for FPGA

Yi, Jae Young*, Szirmay Laszlo**, Yi, Cheon Hee***
* Technical Univ. of Budapest, Ph.D. Student E-mail : yi_young@hotmail.com
** Technical Univ. of Budapest, Professor E-mail : szirmay@seeger.bme.iit.hu
*** Chongju Univ., Dept of Electronic Engineering Professor E-mail : yicheon@chongju.ac.kr
Tel : +81-43-229-8448 Fax : +81-43-229-8461

Abstract: We studied on the performance optimized synthesis and mapping of design on to one or more FPGA device. Our multi-phased approach optimized the key parameters that affect performance by adequately modeling the impact on wire length, routability, and performance during technology mapping to produce designs that have high performance and high routability potential.

Key words: FPGA, Mapping, routability, LUT.

1. INTRODUCTION

In this paper we are studied on the performance optimized mapping of design on to one or more FPGA Device. Our goal in the technology mapping phase is to arrive at a design implementation which has the best performance and routing potential.[1][2][3] in order to achieve high performance implementations it is important to minimization with minimal increases in area and interconnections and thereby indirectly improves the quality of placement and routing to promote smaller wire delays in general. In additions, we propose to complement the depth mapping with minimum critical wire lengths using timing driver pre-placement to derive placement and routing constraints.

2. APPROACH TO PERFORMANCE OPTIMIZED TECHNOLOGY MAPPING

In this chapter we propose a two phased approach for technology mapping shown in Figure 1.

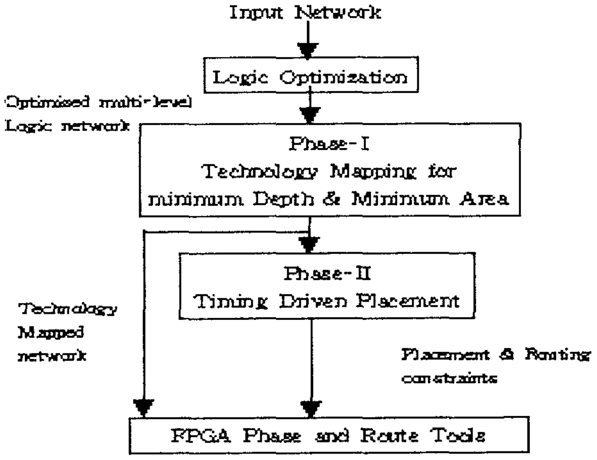
Input Network
Logic Optimization
Optimized multi-level Logic network
Phase-I
Technology Mapping for minimum Depth & Minimum Area
Phase-II
Timing Driven Placement
Technology Mapped network
Placement & Routing constraints
FPGA Phase and Route Tools

Figure 1. Performance of optimized technology mapping

The input network is first conditioned through two-level and multi-level optimization using logic optimization.[4] The input network consists of primitive gates(i.e. AND, OR, NOT etc). In the first phase we perform simultaneous depth and area minimized technology mapping. In the second phase we perform a timing driver placement to minimize critical wire lengths and to prevent alternate critical path. The outcome of the second phase is a set of placement and routing constraints which are than phased along with the mapped design to Xilinx's FPGA place and route tools.[5]

The chortle-d approach[1][6] succeeded in significantly reducing the depth of logic, but, at a significant cost in terms of number of LUTs and number of connections. Chortle-d demonstrated that their approach produced mapping with optimal depth when the input is fan-out free tree and when the number of inputs to the LUT[3][7] is less than or equal to 6. The mis-pga(delay)[8] uses a two phased approach to delay optimized mapping. In the first phase the network depth is minimized by controlling critical node into their fan-outs and re-synthesizing the collapsed node with fewer number of levels using a number of decomposition techniques such as Roth-Karp,[9] co-factoring AND/OR decompositions, and algebraic decompositions (i.e. Kernel cube factoring). The second phase used logic re-synthesis during a simulated annealing based timing driver placement to minimize critical paths delays The results from the first phase of mis-pga(delay) were significantly better in terms of area and number of connections, but yielded larger number of levels, the smaller area and connections in designs produced by mis_pga(delay) resulted in factor designs after place and route compared to chortle-d.[6] The results from phase-II of mis-pga(delay) however were not quite promising. In many instances it was observed that re-synthesis operations during placement significantly deteriorated the circuit performance.

3. APPROACH TO PERFORMANCE DIRECTED TECHNOLOGY MAPPING

We will define some basic terms which are used in describing the technology mapping approach. A combinational input network for technology mapping consisting of a set of Boolean functions may be looked upon as a directed acyclic graph(DAG) G=(V.E). The articles(or nodes) are the primitive Boolean operators(i.e. AND, OR, NOT etc) and the directed edges(from the output of a node to an input of another node) are the connections between operators. Edges also carry phase information indicating whether an operators output must be complemented. A primary output node has no outgoing edges and a primary input node has no incoming edges. The mapping process adding one or more look-up-table(LUT)[6][10] to each node visited to realize the node's function. It showed be noted that our assumption of a combinational logic network as an input to technology mapping is not a limitations. When we are giver a general Boolean network consisting of sequential elements, the sequential elements are ignored during the technology mapping process, after the mapping of the combinational logic is completed, the sequential elements are either assigned to existing LUTs or to new LUTs as necessary. For the dúration of technology mapping the inputs to the sequential elements are treated as primary outputs and the outputs of sequential elements are treated as primary inputs. The performance of a design mapped on to an FPGA device is governed:

.Logic Delays(Ld) encountered to the number of levels of logic on a circuit path. .Wire Delays(Wd) encountered to program able switches and capacitances of the wire segments presents in the circuit path. We propose a new approach to performance optimized mapping to coherently address the factors that govern performance. The importance attributes of our approach are; .Simultaneous depths and area minimized technology mapping. Generate placement and routing constraints to minimize critical wire lengths and control the wire delays.

In order to achieve the above, we propose a two phased approach. In the first phase we present an approach to simultaneous depth and area minimization. In the second phase we reinforce the depth minimization by controlling the critical wire lengths and wire delays via timing driven placement.

1. Clique partitioning based technology mapping

The mapping process involves a post-order traversal of the input network(or Directed Acyclic Graph G=(V.E). At each node v visited in post-order our goal is to minimize the number of LUTs(Look Up Tables) required to realize the function of node v. In this process we identify an efficient decomposition of v and merge as many of v's fan-in LUTs as possible to realize the function of v. The pseudo-code for technology mapping out lining our approach for area minimized technology mapping is shown in Figure2.

```
Technology mapping (G=(V.E)) for each node v encountered in post-order{
1) Construct a merge-graph G'=(V',E') for node V.
2) Perform clique partitioning on G' to produce v1', v2', vj', where j is minimized and each vi' is a feasible clique(i.e. fits into a k-LUT)
If (numinps (v)>1) {
3) Combine the set of LUTs in the clique and PIs in the fan-in of v using IT3 operations.
4) If(mapping not complete)
   Add minimum additional LUTs to complete the mapping.
  }
}
```

Figure 2. Pseudo-code for area optimized technology mapping

In the first step of TechMap-A we construct a merge-graph G'=(V',E'). The vertex set V consists of one vertex for each LUT in the fan-in of node v. The edge set E' consists of an edge(u,w) for each mergeable pair of LUTs u and w in the fan-in of v. These edges correspond to the IT1 and IT2 types interactions discussed earlier.

Simultaneous depth and area minimization in order to minimize path delays, it is essential to minimize the depth of the logic and also the wire lengths of the connections in the path. However, since wire lengths are not available prior to technology mapping and placement, we indirectly address the factors that affect wire lengths, placement, and routing(i.e. area and number of interconnections) during depth minimization our approach to depth minimization is outlined in Figure 3. Algorithms DM(G=(V.E) Regd-Depth)

```
1) Perform area optimized technology mapping of the starting network G using technology mapping
2) Compute level slacks for the area mapped network and identify critical nodes(i.e. slacks(node) <=0).
3) For each node v of G visited in post-order
 {
 If(v is critical)
   minimize-depth(v)
 else
   minimize-area(v)
 } Pseudo-code for area efficient depth minimization.
```

Figure 3. Pseudo-code for area efficient depth minimization

The inputs to Algorithm DM consist of the starting network to be mapped and also the required depth of logic in mapped network the goals of Algorithms DM are; Achieve technology mapping with specified depth whenever possible or minimized the depth when the specified depth requirements can not be met.

Minimize the number of LUTs and the number of interconnections which in influence wire lengths and the wire delays. This Algorithm DM can produces level efficient designs with fewer number of LUTs and fewer connections which in turn improve the potential the potential to minimize wire lengths during placement and routing.

Depth minimization

Our approach to simultaneous depth and are a minimized technology mapping for each critical node is In Figure 4.

```
Minimize-depth(v)
1) Cost-limit=Estimate-cost(v.0.0)
2) Construct a merge-graph G'=V',E')
   Consisting of a vertex for each LUT in the fan-in of node v, Add and edge(x,y) for each pair of LUTs X and Y in v' that can we merged into a single LUT;
3) While(E' is not empty){
4)    Initialize min-cost and best-edge;
5)    For each edge I in E'
6)        Cost(e) = Estimate-cost(v,1,e)
7)        Update min-cost and best-edge;
      }
8) If (min-cost>cost-limit) break;
9) If(best-edge found) {
10)       Merge vertices p,q connected by
      best-edge into a new vertex r.
11)       Update Graph G' by deleting vertices p,z and associated edges. Add new vertex r and edges from r to other mergeable vertices in G';
      }
      Cost-limit=min-cost;
      }
13) Construct k-ary tree to complete the mapping;
```

Figure 4. Pseudo-code for minimize-depth.

Our goals during the mapping of critical node v are to;

Minimize depth of the LUT realizing node v .Minimize the number of LUTs required to realize v.

.Maximize the number of unused inputs(i.e extension potential) of the lead LUT realizing v.

This strategy results in selecting an area efficient and minimum depth mapping of the critical node additionally, our approach to maximizing the extension potential of the load LUT provides further depth and area optimization opportunities to the fan-out node of the nodes currently being mapped. The first step in the minimize depth routine uses an estimate-cost routine to determine the cost of mapping node v with minimum depth assuming that none of the LUTs in the fan-in of node v can be merged. The second and third parameters used in Estimate-cost specify whether a pair of LUTs mush be merged and the corresponding edge respectively. The Estimate-cost routine constructs a K-ary tree utilizing the unused inputs can meet some of there demands and are considered to be suppliers for connections to other LUTs and PIs. The PIs are considered to be at depth o.

```
Estimate-cost
  depth=0
  while(tree not complete) {
1) Meet demands at depth d using suppliers at depth d+1;
2) Meet any additional demands at depth d by adding new K-input nodes at dept h d+1 when |demands|>k.Carry and fractional demands to depth d+1;
3) depth=depth+1;
```

Figure 5. Pseudo code of a simplified version of estimate-cost.

4. CONCLUSION

In this study we address the problem of performance optimized synthesis of designs on to one or more FPGAs. Our multi-phased approach optimizes the key parameters that affect performance by adequately modeling the impact on wire length. routability, and performance during technology mapping to produce designs that have high performance and high routability potential. From this approach we have developed novel techniques for technology mapping which produce designs with high performance and routability potential. Our approach was to perform a simultaneous depth and area minimized technology mapping in the first phase so that logic depth of the network in minimized with minimal area penalty. This strategy of controlling the area costs during depth minimization promotes good placement and routing configurations and does not adversely impact the wire lengths and wire delays that can be achieved. The second phase of our approach involved a timing driven placement to control the critical wire lengths and generate placement and routing constraints that could be used with the actual FPGA place and route tools.

References

[1] R.J.Francis, J. Rose, and Z.Vranesie, "Technology Mapping of Look-Up Table based FPGAs for Performance", ICCAD, 1991.

[2] J.Cory, Y.Ding, "On Area/Depth Trade off in LUT based FPGA Technology Mapping", IEEE Trans, on. VLSI systems, Jan.1994.

[3] J.Cory, Y.Ding, "On Area/Depth Trade off in LUT based FPGA Technology Mapping", IEEE Trans, on. VLSI systems, Jan.1994.

[4] R.K. Bryton, G.D. Hachtel, A.L.S. Vincentelli, "Multilevel logic ynthesis", Proc. of IEEE, Feb. 1990

[5] Xilinx's user manual, 1998.

[6] R.J. Francis, J. Rose, K.Chung, Chortle: "A Technology Mapping Program for Look Up Table-based Field Programmable Gate Arrays", DAC-1990.

[7] H.Yang, D.F.Wong, Edge-Map: "Optimal Performance Driven Technology Mapping for Iterative LUT based FPGA Designs", ICCAD, 1994.

[8] R.Murgai, N. Shenoy, R.K. Brayton and A.S. Vincentelli, "Performance Directed Synthesis Table Look Up Programmable Gate Arrays", ICCAD, 1991.

[9] J.P. Roth, R.M. Karp, "Minimization over Boolean Graphs" IBM Journal of research and development April, 1992..

[10] N.B. Bhat, "Library Based Mapping for LUT Based FPGAs Revisited", Int workshop on logic synthesis. 1993.