Technology Mapping of Lookup Table-Based FPGAs for Performance

Robert J. Francis, Jonathan Rose, Zvonko Vranesic

Department of Electrical Engineering, University of Toronto

Abstract

A new technology mapping algorithm that reduces the delay of combinational circuits implemented with lookup table-based Field-Programmable Gate Arrays (FPGAs) is presented. [1] *The algorithm reduces the contribution of logic block delays to the critical path delay by reducing the number of lookup tables on the critical path.*

The key feature of the algorithm is the use of bin packing to determine the gate-level decomposition of every node in the network. In addition, reconvergent paths and the replication of logic at fanout nodes are exploited to further reduce the depth of the lookup table circuit. For fanout-free trees the algorithm will construct the optimal depth K-input lookup table circuit when K is less than or equal to 6.

1 Introduction

Lookup table-based Field-Programmable Gate Arrays (FPGAs) require a technology mapping approach that deals specifically with lookup tables because the large libraries required to represent lookup tables make library-based technology mapping impractical [1]. Previous lookup table technology mapping programs include mis-pga [2], Asyl [3], Hydra [4], Chortle-crf [5], and Xmap [6]. This paper focuses on technology mapping for the delay optimization of FPGAs that use lookup tables to implement combinational logic.

Even though routing delays in FPGAs are layout dependent, an important task in minimizing the critical path delay is to reduce the number of lookup tables on the critical path. In this paper, the critical path delay is measured by the number of lookup tables on the path. This makes the assumption that all lookup tables have the same intrinsic delay and approximates the routing delay between lookup tables as a fixed constant.

2 Depth Optimization Algorithm

The algorithm takes a combinational network and produces a circuit of K-input lookup tables that implements the network. The principal technique used to map the network is dynamic programming [7]. The network is traversed beginning at the primary inputs and proceeding toward the primary outputs. At each node in the network a *Best Circuit* is constructed that implements the sub-network extending from the node to the primary inputs. The principal goal is to minimize the depth of the Best Circuit. A secondary goal is to maximize the number of unused inputs of the output lookup table, because it may be possible to use these inputs to reduce the depth of the Best Circuit implementing the subsequent node.

Figure 1b illustrates a circuit of 5-input lookup tables. The dotted boundaries indicate the functions implemented by each lookup table. The number in the lower right hand corner of each lookup table indicates the depth of the lookup table. Note that the lookup tables preceding the AND nodes are not shown in this figure, but they are assumed to contribute to the overall depth as indicated. All examples in the remainder of this paper will assume that K is equal to 5.

The gate-level decomposition of a network node can have a significant effect on the depth of the final circuit. The depth of the circuit of 5-input lookup tables shown in Figure 1a is 5 levels. The circuit in Figure 1b implements the same network with the OR node decomposed in such a way that the circuit has a depth of only 4 levels.

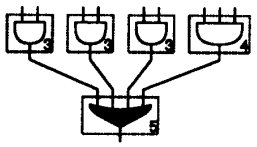
The construction of the Best Circuit for a node depends upon the Best Circuits for its immediate fanin nodes. The order of the network traversal ensures that these fanin Best Circuits have been previously constructed. The output lookup tables of the fanin Best Circuits are referred to as the fanin lookup tables. For each node in the network a tree of lookup tables that implements both the functions of the fanin lookup tables and a decomposition of the node is constructed. This decomposition tree consists of a series of strata where each *stratum* contains all lookup tables at the same depth. An outline of the procedure used to find the Best Circuit for a node is given in Figure 2.

2.1 Minimizing Stratum Width

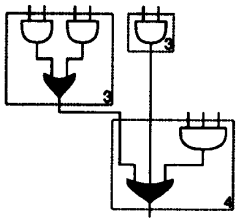
The first step in constructing the decomposition tree is to separate the fanin lookup tables into strata and to minimize the number of lookup tables in each stratum using a bin packing algorithm. In general, the goal of bin packing is to find the minimum number of bins into which a set of boxes can be packed. In this case, the

[1] This work was supported by NSERC Operating Grants #URF0043298 and #OGP0005280, a research grant from Bell-Northern Research, and a research grant from the ITRC of Ontario.

CH3026-2/91/0000/0568/$01.00 © 1991 IEEE



a) without gate decomposition



b) with gate decomposition
Figure 1.

bins are the lookup tables of the stratum and the boxes are the fanin lookup tables at the same depth as the stratum. The capacity of each bin is K and the size of each box (fanin lookup table) is its number of used inputs. In Figure 3a the size of each of the five boxes is 2. The stratum lookup tables after bin packing are shown in Figure 3b. The actual bin packing algorithm used is the First Fit Decreasing heuristic as described in [5].

2.2 Completing the Decomposition Tree

The decomposition tree is completed by proceeding from the uppermost stratum to the deepest stratum, connecting the outputs of lookup tables in stratum D to unused inputs of lookup tables in stratum $D+1$. Note that this may require the addition of extra lookup tables in stratum $D+1$. When a stratum containing only one lookup table is reached, with no other deeper strata remaining, the decomposition tree is complete. Figure 3c shows the completed decomposition tree constructed from the stratum lookup tables in Figure 3b. At this point single input lookup tables, such as the one in stratum 4 of Figure 3c, can be eliminated.

2.3 Depth Optimality

The Best Circuit for a node constructed using the completed decomposition tree can be shown to have the minimum possible depth if the sub-network extending from the node to the primary inputs is a fanout-free tree and $K \leq 6$ [8]. For $K > 6$ there exists a set of boxes that the First Fit Decreasing bin packing algorithm does not pack into the minimum number of bins. This can lead to the construction of a sub-optimal circuit.

2.4 Exploiting Reconvergent Paths

The packing of the stratum lookup tables may be improved by exploiting local reconvergent paths. When two boxes that share an input are packed into one bin the number of inputs required is the total number of *distinct* inputs, which is less than the sum of the boxes' individual sizes. This may lead to a superior bin packing within a stratum, which in turn may lead to a superior Best Circuit. It is possible, however, that the best bin packing requires that the two boxes be packed into different bins. To find the Best Circuit, both the packing with and without the forced packing of the two boxes into the same bin are considered.

A further complication is that more than one pair of reconvergent paths may terminate at the node. The algorithm first finds all pairs of local reconvergent paths and then searches all possible combinations, including none, of these pairs. For each combination a circuit is constructed by first packing the respective boxes of the chosen pairs and then proceeding with the construction of the decomposition tree. The circuit where the output lookup table has the minimum depth, and the maximum number of unused inputs, is retained as the Best Circuit.

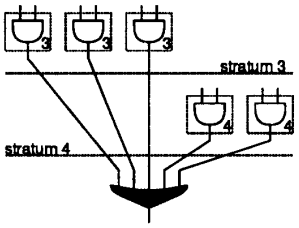
The number of combinations to be searched grows exponentially with the number of reconvergent pairs. However, in our experiments with the MCNC benchmark networks the number of pairs found was small enough to make the search practical.

```
FindBestCircuit (node)
   {
   group fanin lookup tables of the
   same depth into separate strata

   for all strata
      {
      minimize number of lookup tables
      in the stratum using bin packing
      }

   D = depth of shallowest stratum

   while the number of lookup tables
   in stratum D is greater than 1 or
   there are strata deeper than D
      {
      for all lookup tables in stratum D
         {
         if there are no unused inputs among
         the lookup tables in stratum D+1
            {
            create a new lookup table and
            add it to stratum D+1
            }
         connect the output of the stratum D
         lookup table to an unused input of
         a stratum D+1 lookup table
         }
      D = D+1
      }
   }
```

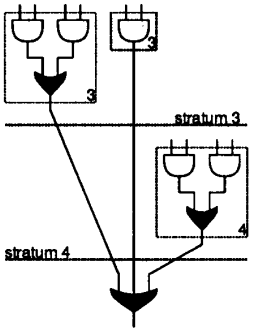
Figure 2: Pseudo code to find Best Circuit

2.5 Replication of Logic

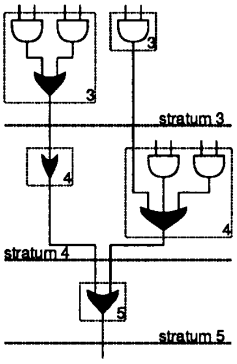
The replication of logic at a fanout node may also reduce the depth of lookup tables in the circuit. For example, in Figure 4a the fanout node is implemented



a) separated into strata



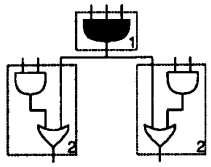
b) minimized strata



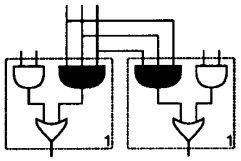
c) complete decomposition tree
Figure 3.

as a lookup table output and the maximum depth of a lookup table in the circuit is 2. In Figure 4b the AND gate is implemented implicitly by replicating it within two lookup tables, which reduces the maximum depth of a lookup table in the circuit to 1.

When the dynamic programming traversal of the network encounters a fanout node, the Best Circuit implementing the fanout node is constructed. A replica of the function of the output lookup table of this Best Circuit is made for each of the fanout edges. The source of the fanout edge is replaced with this replica only if this reduces the depth, or reduces the number of lookup tables without increasing the depth, at the next fanout node, or output, encountered on the path starting with the fanout edge.



a) no replicated logic



b) with replicated logic
Figure 4.

2.6 Reducing Area

To reduce the total number of lookup tables without increasing the maximum depth of the circuit, the depth optimizing gate-level decomposition is applied only to the critical paths of the circuit. First the target maximum depth is found by reducing the depth of all lookup tables in the circuit. Next the the total number of lookup tables in the circuit is reduced by mapping the entire network using the Chortle-crf [5] area optimizing gate-level decomposition. Any portions of the circuit that exceed the target depth are then re-mapped using the depth optimizing gate-level decomposition in an iterative approach that proceeds until the entire circuit meets the target depth.

Finally a traversal of the circuit is made to find lookup tables, without fanout, that can be eliminated by merging their function into the following lookup table without exceeding its K input capacity.

3 Results

The algorithm has been implemented as an extension to the Chortle-crf [5] technology mapping program and for clarity will be referred to as Chortle-d. A series of experiments were performed comparing Chortle-d to Chortle-crf and mis-pga [2]. In these experiments 20 networks from the MCNC logic synthesis benchmark suite were mapped into circuits of 5-input lookup tables, Note that the goal of both Chortle-crf and mis-pga is the reduction of the total number of lookup tables in the circuit. Before each network was mapped, technology independent logic optimization was performed using the misII [9] standard script followed by the mis2.2 speedup command [10].

The maximum depth and the number of lookup tables in the mapped circuits are recorded in Table 1. Chortle-d uses an average 35 % fewer levels of lookup tables than Chortle-crf, and an average of 29 % fewer levels than mis-pga to implement the benchmark networks. Note, however, that the Chortle-d circuits have on average 59 % more lookup tables than the Chortle-crf circuits.

Network	Chortle-crf		mis-pga		Chortle-d			XNFOPT	
	depth	lookups	depth	lookups	depth	lookups	CLBs	depth	CLBs
z4ml	4	13	3	10	3	25	18	4	8
misex1	4	18	3	16	2	19	13	3	12
vg2	5	39	5	37	4	55	43	4	34
5xp1	4	27	4	26	3	26	21	5	28
count	5	58	5	59	4	91	78	5	72
9symml	7	62	7	65	5	59	51	7	61
9sym	8	65	8	65	5	63	53	6	75
apex7	6	78	5	72	4	108	80	5	78
rd84	7	41	6	40	4	61	54	6	46
e64	7	139	6	139	3	209	140	4	135
C880	13	172	11	172	8	329	249	11	256
apex2	8	124	7	133	6	201	155	7	179
alu2	13	128	15	127	9	227	177	13	205
duke2	7	152	7	161	4	241	180	5	178
C499	8	141	7	123	6	382	270	8	248
rot	11	214	11	203	6	326	257	9	235
apex6	6	235	6	221	4	308	268	6	237
alu4	17	231	16	234	10	500	382	15	444
apex4	11	624	9	664	6	1112	932	7	869
des	10	981	*	*	6	2086	1580	7	1251

*mis-pga terminated abnormally

Table 1.

3.1 Xilinx CLBs

The Configurable Logic Blocks (CLBs) in the Xilinx 3000 series FPGAs use lookup tables to implement combinational logic. A circuit of CLBs can be derived from a circuit of 5-input lookup tables because each CLB can implement one 5-input lookup table or two 4-input lookup tables with at most 5 distinct inputs. Reducing the number of CLBs in the derived circuit can be restated as a Maximum Cardinality Matching problem [2]. The depth of the derived CLB circuit is the same as the depth of the lookup table circuit.

The derived CLB circuits were compared to circuits produced using the Xilinx design tools XNFOPT and XNFMAP to perform logic optimization and technology mapping [11]. In these experiments the XNFOPT delay optimization was used and the program was limited to 4 iterations, in an attempt to keep the execution time comparable to Chortle-d. The maximum depth and the number of CLBs in the mapped circuits are recorded in Table 1. On average, Chortle-d requires 24 % fewer levels of CLBs, and 7 % more CLBs, than XNFOPT to implement the benchmark networks.

4 Conclusions

This paper has presented a technology mapping algorithm for K-input lookup table circuits that minimizes the depth of the final circuit. The key feature of the algorithm is the use of bin packing to choose a gate-level decomposition of each node in the network that reduces the depth of the circuit. The algorithm also exploits reconvergent paths and the replication of logic at fanout nodes to reduce depth.

In an experimental comparison Chortle-d required on average 35 % fewer levels on 5-input lookup tables than Chortle-crf, 29 % fewer levels of lookup tables than mis-pga and 24 % fewer levels of Xilinx 3000 CLBs than XNFOPT to implement a set of MCNC benchmark networks.

The algorithm is optimal, with respect to depth, provided that the combinational network is a fanout-free tree and $K \leq 6$.

References

[1] R. J. Francis, J. Rose, K. Chung, "Chortle: A Technology Mapping Program for Lookup Table-Based Field Programmable Gate Arrays," Proc. 27th DAC, June 1990, pp. 613-619.

[2] R. Murgai, Y, Nishizaki, N. Shenay, R. K. Brayton, A. Sangiovanni-Vincentelli, "Logic Synthesis for Programmable Gate Arrays," Proc. 27th DAC, June 1990, pp. 620-625.

[3] P. Abouzeid, L. Bouchet, K. Sakouti, G. Saucier, P. Sicard, "Lexicographical Expression of Boolean Function for Multilevel Synthesis of high Speed Circuits," Proc. SASHIMI 90, Oct. 1990, pp. 31-39.

[4] D. Filo, J. C. Yang, F. Mailhot, G. De Micheli, "Technology Mapping for a Two-Output RAM-based field Programmable Gate Array," Proc. EDAC 91, Feb, 1991, pp. 534-538.

[5] R. J. Francis, J. Rose, Z. Vranesic, "Chortle-crf: Fast Technology Mapping for Lookup Table-Based FPGAs," Proc. 28th DAC, June 1991 pp. 227-233.

[6] K. Karplus, "Xmap: a Technology Mapper for Table-lookup Field-Programmable Gate Arrays," Proc, 28th DAC, June 1991, pp. 240-243.

[7] K. Keutzer, "DAGON: Technology Binding and Local Optimization by DAG Matching," Proc. 24th DAC, June 1987, pp. 341-347.

[8] R. J. Francis, "Technology Mapping for Lookup Table-Based FPGAs," Ph.D. Thesis in preparation, University of Toronto, Department of Electrical Engineering.

[9] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, A. Wang, "MIS: a Multiple-Level Logic Optimization System," IEEE Tr. CAD, Vol CAD-6, No. 6, Nov. 1987, pp. 1062-1081.

[10] K. J. Singh, A. R. Wang, R. K. Brayton, A. Sangiovanni-Vincentelli, "Timing Optimization of Combinational Logic" Proc. ICCAD 88, Nov 1988, pp.282-285.

[11] XACT LCA Development System, Vol. II, Xilinx Inc., 1989.