Area Recovery under Depth Constraint by Cut Substitution for Technology Mapping for LUT-based FPGAs

Taiga Takata

Graduate School of Information Science
and Electrical Engineering
Kyushu University
e-mail: taiga@c.csce.kyushu-u.ac.jp

Yusuke Matsunaga

Faculty of Information Science
and Electrical Engineering
Kyushu University
e-mail matsunaga@c.csce.kyushu-u.ac.jp

Abstract— In this paper we present the post-processing algorithm, Cut Substitution, for technology mapping for LUT-based FPGAs to minimize the area under depth minimum constraint. The problem to generate a LUT's network whose area is minimum under depth minimum costraint seems to be as difficult as NP-Hard class problem. Cut Substitution is the process to generate a local optimum solution by eliminating redundant LUTs while the depth of network is maintained. The experiments shows that the proposed method derives the solutions whose area are 9% smaller than the solutions of a previous state-of-the-art, DAOmap on average.

I. Introduction

Recently, design for LUT-based Field Programmable Gate Array(FPGA) is becoming popular. LUT-based FPGA is a reconfigurable device whose function can be changed after fabrication. As LUT-based FPGAs can be used for universal design, using FPGAs is suitable for the low-cost design of small-lot production.

Technology mapping for LUT-based FPGA is a part of logic synthesis which has a significant impact on delay and power of the implemented circuit. Technology mapping for LUT-based FPGA is the process to convert a given boolean network into a functionally equivalent network comprised of K-input Look-Up Tables (LUT). A K-input LUT is a basic programmable logic element that can implement any Boolean function of up to K variables.

To minimize the delay of implemented circuits, it is preferable to generate a depth-minimum network during a technology mapping process. The depth of a network means the length of the longest path in the network. Among the depth-minimum networks, an area-minimum network is appropriate to expand the flexibility of placement and routing.Area denotes the number of LUTs in this paper.

The problem to generate area minimum network in technology mapping is NP-Hard[3], and the problem to generate area minimum network under depth minimum constraint seems to be very difficult. Some previous technology mapping algorithms (FlowMap[1], FlowMap-r[2], CutMap[7], Hermes[5], DAOMap[4], Imap[8]) guarantee to make a depth-minimum network, but none of them guarantee that a generated network is area-minimum under the depth-minimum constraint.

In this paper we propose Cut Substitution that is a heuristic algorithm for the post-processing of technology mapping to generate area-minimal network under depth constraint. Cut Substitution reduces the area of a networks while keeping the same depth. The concept of Cut Substitution is to eliminate some excessive cuts directly. A cut is a nomination of a LUT. Cut Substitution identifies the excessive cuts among the cuts selected at technology mapping phase. Cut Substitution substitutes the cuts connected with the excessive cuts for other cuts, and eliminate the excessive cuts with keeping same depth. The Optimality of the depth of the network is guaranteed.

Experimental results show that Cut Substitution, combined with a simple depth optimum technology mapping algorithm, has generated LUT networks with 9% smaller area than the LUT networks generated by DAOmap on average. The run time of the combination of the simple heuristic algorithm and Cut Substitution is 3% shorter than that of DAOmap.

The rest of this paper is organized as follows. Section 2 presents some basic definitions and formulation. Section 3 presents a review of a typical heuristic algorithm based on cost calculation. Section 4 explains Cut Substitution. Section 5 presents experimental results. Section 6 concludes this paper.

II. Definition and Problem Formulation

A boolean network given for technology mapping is called a subject graph. Subject graphs is a Directed Acyclic Graphs (DAG). A node of a subject graph represents a primitive boolean function. For a subject graph $N(V, E)$, if a node $i \in V$ is an input of node $j \in V$, there is a directed edge $(i, j) \in E$. The solution of technology mapping is called a LUT network. A LUT network is a DAG where each node represents a K-input LUT and an edge (v, w) exists if v is an input of w.

For a DAG, a fanin of $v \in V$ is the immediate predecessor node of v, and the set of fanins of v is denoted by $fanin(v) = \{u \in V \mid \exists(u, v) \in E\}$. A fanout of $v \in V$ is the immediate succesor of v. and the set of fanouts of v is denoted by $fanout(v) = \{w \in V \mid \exists(v, w)\}$. A node v whose $fanin(v) = \phi$ is called a Primary Input(PI). A node v whose $fanout(v) = \phi$ is called a Primary Output(PO). A depth of a DAG is the length of the longest path from any PI to PO. A transitive fanin (TFI) of a node v is the set of all nodes that exists on all path from PI to v. More exactly, a transitive fanin, denoted by $TFI(v)$, is defined by $fanin(v) \cup \bigcup_{u \in fanin(v)} TFI(u)$. A transitive fanin subgraph of $v \in V$ is the subgraph induced by TFI(v), denoted by $TFIG(v)$.

A cut at node v is a partitioning of TFI(v), denoted by $c_v = (X, \bar{X})$, where $X \subseteq TFI(v), \bar{X} \subseteq TFI(v), X \cap \bar{X} = \phi$. For a cut c_v, the node v is called a root of c_v. A cut $c_v = (X, \bar{X})$ satisfies the condition; $\forall pi \in TFI(v), pi \in X, v \in \bar{X}$, where pi represents a PI. For a cut $c_v = (X, \bar{X})$, the cut-set

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

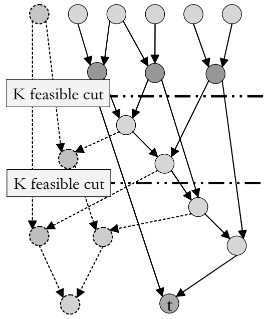

Fig. 1. k-feasible cut (example if k=3)

$input(c_v)$, is defined as $\{i \mid i \in X, \exists j \in \bar{X}, \exists(i,j) \in E\}$. A cut is called k-feasible if a cut meets the following conditions; the subgraph induced by $\bar{X}$ is a connected graph; any node in $\bar{X}$ do not have a fanout edge to any node in X, i.e. $\forall i \in \bar{X}, \forall j \in X, (i,j) \notin E$; the number of the elements of cut set is less than k, i.e. $\mid input(c) \mid \leq k$, where k is the maximum number of inputs of a LUT. Figure 1 shows two examples of k-feasible cuts whose roots are t if $k = 3$. The circles and arrows represent the nodes and edges of a DAG. The two horizontal dash lines are borders of each k-feasible cuts. In figure 1, other k-feasible cuts with root t are omitted. In the rest of this paper, only k-feasible cuts are considered. Therefore a k-feasible cut is simply called a cut in the rest of this paper. The subgraph induced by $\bar{X}$ of a cut $c_v = (X, \bar{X})$ is called a k-feasible cone(KFC), denoted by $kfc(c_v)$. For a KFC $kfc(c_v)$, v is called a root of $kfc(c_v)$, and a node in $input(c_v)$ is called a leaf of $kfc(c_v)$. Each KFC in a subject graph can be implemented in a functionally equivalent k-input LUT that have v as output, and $input(c_v)$ as inputs. A $kfc(c_v)$ is unique for a cut c_v. So a cut c_v and a node of a LUT network are one-to-one correspondence. A LUT corresponding to a cut c_v is represented as LUT_{c_v}.

The technology mapping problem for LUT-based FPGAs is to cover a given subject graph with KFCs. More specifically, technology mapping problem is to decide a set of cuts such that the KFCs corresponding to the cuts in the set cover all of the nodes of a subject graph. A set of cuts is feasible if all of the inputs of any cut in the set are PI or the root of other cuts in the selected set. A result of technology mapping must be feasible. The technology mapping problem for area minimization under depth minimum constraint is to decide a set of cuts in such a way that the total number of cuts in a set is minimized while the minimum depth of the LUT network is guaranteed.

III. Review of Traditional Covering based Cost Propagation

The process of typical heuristic algorithms for depth-optimum area-optimal technology mapping for LUT-based FPGAs consists of three phases. First phase is cut enumeration[4] that tick off all of the k-feasible cuts in a subject graph. The second phase is cost calculation, and the final phase is covering (also called cut selection). Because cut enumeration has no effect on the quality of a LUT network, we omit the description of cut enumeration.

Cost calculation is to calculate the depth cost and area cost of each cut at each node[6] [4]. Depth cost of a cut c_v is the minimum depth of the mapping solution for transitive fanin subgraph $TFIG(v)$ under the constraint that c_v is selected as a part of the solution. Area cost of a cut c_v means the minimum number of cuts of the solution of $TFIG(v)$ similarly. Because the way to estimate exact minimum number of cuts in practical time is not known, existing algorithms calculate approximate area cost by some heuristic methods. On the other hand, depth cost can be calculated exactly in practical time.

```
Cut Substitution( $S(V,E), C, C', N(V_{C'}, E_{C'})$ ){
  while (1) {
      $C_{exc}$ := Excessive_Cut_Enum($C'$, $C$);
      if $C_{exc}$ is empty {
          break;
      }
      $c^{best}$ = Choice_Best_Cut( $C_{exc}$ );
      $N(V_{C'}, E_{C'})$
          := Cut_Elim( $c^{best}$, $N(V_{C'}, E_{C'})$ );
  }
  return $N(V_{C'}, E_{C'})$;
}
```

Fig. 2. Pseudo code for Cut Substitution

Covering is to traverse the subject graph from POs to PIs, and select the Min-cost cuts, and convert the selected cuts into LUTs. A min-cost cut at a node v is the cut whose area cost is minimum as far as the depth cost is not larger than the required depth at v. After setting the minimum depth to all POs as required depth, covering is executed for each POs as follows. From a PO o, a min-cost c is selected. Next, the min-cost cuts of $input(c)$ are selected. Above process executed iteratively until the process reaches all PI in $TFI(o)$.

In some cases, excessive cuts are selected during covering process. This is because the coverings are essentially executed for each transitive fanin graph of each PO separately. Figure 3(a) shows an example of an excessive cut. The cut c_i is represented as an inverted trapezoid that illustrates $kfc(c_i)$, where $i \in \{v, w, x, y, z\}$. The cut c_z and c_y is selected to cover $TFIG(z)$. If the cut c_x and c_w is the min-cost cut at x and w respectively, c_x and c_w is selected, and c_v is selected. If the cuts whose inputs $\{p, y\}$ at x and $\{q, y\}$ at w are selected, c_v is not selected. In this case, c_v is an excessive cut.

Some of existing algorithms try to reduce some area after covering. For example, DAOmap[4] reduces area by re-mapping iteratively. At re-mapping phase, DAOmap adjusts the area cost of each cut to reduce several excessive cuts. At re-mapping phase, the adjusted cost of area is heuristics, and the covering at re-mapping phase are executed for each transitive fanin graph of each PO separately.

IV. Cut Substitution for Cut Elimination

Cut Substitution directly eliminates several excessive cuts from the set of cuts selected at covering phase. Let $S(V, E)$ represent a given subject graph. C is a set of all the cuts found in $S(V, E)$. $C' \subseteq C$ is a set of the cuts selected as a mapping solution. $N(V_{C'}, E_{C'})$ is a LUT network that is derived from C'. $S(V, E)$, C, C', and $N(V_{C'}, E_{C'})$ are given for Cut Substitution. For $LUT \in V_{C'}$ we define level $lev(LUT)$, and required level $req(LUT)$. Level $lev(LUT)$ is the length of the longest path from any PI to LUT, and can be defined as $\max_{LUT_i \in fanin(LUT)} lev(LUT_i) + 1$. The level of PI is 0. Required level $req(LUT)$ is the level required to maintain the depth of a LUT network, and defined as $\min_{LUT_i \in fanout(LUT)} req(LUT_i) - 1$. The required level of PO is a depth constraint that is given.

Figure 2 provides a pseudo code of Cut Substitution. In fugure 2, C_{exc} represents a set of cuts, and c^{best} represents a cut. Each iteration consists of three phase; Excessive Cut

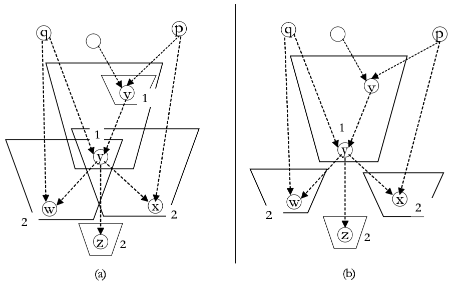

Fig. 3. An example of an elimination of excessive cut c_v (K=3)

Enumeration, Choice of the Best Cut, and Cut Elimination. $Excessive\_Cut\_Enum$, $Choice\_Best\_Cut$, and Cut_Elim in figure2 represents Excessive Cut Enumeration, Choice of the Best Cut, and Cut Elimination, respectively. First, Cut Substitution identifies all the excessive cuts among the cuts selected by covering. An excessive cut is such a cut that all of the Fo-Cuts are able to be substituted by other cuts, maintaining the depth of LUT network. A Fo-Cut of a cut c is such a cut c' that $LUT_c \in fanin(LUT_{c'})$. Second, Cut Substitution calculates the best-cut among the excessive cuts with the heuristic metric. Third, Cut Substitution substitutes the Fo-Cuts for other cuts, and eliminates the best-cut. That iteration is executed until no excessive cut is found.

A. Excessive Cut Enumeration

At Excessive Cut Enumeration, all of the excessive cuts in C' are enumerated. An excessive cut is a cut $c_v \in C'$ that all of the cuts $c'_w \in C'$, such $v \in input(c'_w)$, can be substituted by other cuts $c''_w \in C$ whose root is the same as c'_w and $v \notin input(c''_w)$. The above cut c'_w is called Fo-Cut of c_v, and c''_w is called a Sub-Cut of c'_w.

A cut $c'_w \in C'$ is able to be substituted by the cut $c''_w \in C$ only if two conditions are satisfied. C'' denotes a set that is obtained by removing the cut c'_w from C', and adding $c''_w \in C$ to C'. The first condition is feasibility. The set C'' obtained by the substituting c''_w for c'_w is feasible if the following condition is met. The PIs represents the set of all PIs.

$$\forall i \in input(c''_w),\ \ i \in PIs \ or \ \exists c_i \in C''$$

The second condition is depth stability. The depth of $N(V_{C''}, E_{C''})$ is not larger than the depth of $N(V_{C'}, E_{C'})$ if the following condition is met.

$$\forall i \in input(c''_w),\ \exists c_i \in C'',\ lev(LUT_{c_i}) + 1 \leq req(LUT_{c'_w})$$

Each cut in C' is checked if it is an excessive cut. A cut $c \in C'$ is checked if all of the Fo-Cut c' of c can be substituted by any set of the cut $c'' \in C$ according to above conditional equations. The excessive cuts and the corresponding sets of Fo-Cuts and the sets of Sub-Cuts are hold. That is, the above check can be executed with C', C'' and $N(V_{C'}, E_{C'})$, and it is not necessary to generate $N(V_{C''}, E_{C''})$.

Figure 3 (a) shows an example of an excessive cut. In figure 3 (a), an inverted trapezoid means a cut in C', denoted by c_i where i is the root of cut c_i. For viewability, an edge of the subject graph is represented by a dash line. A number by a node i represents the level of the LUT_{c_i} corresponding to the cut c_i. In figure 3 (a), the node y and v are duplicated. The cut c_v is an excessive cut, because all of the Fo-Cuts $\{c_x, c_w\}$ are able to be substituted by other cuts while the feasibility and depth stability are met. There is the cut c'_x whose inputs are $\{y, p\}$, and we denotes C'' as the set of cuts obtained by removing c_x from C', and adding c'_x to C'. For node y of $input(c'_x)$, there is the cut $c_y \in C'$, and p of $input(c'_x)$ is PI. So feasibility is met about C''. Because $lev(LUT_{c_y}) = 1$, the depth stability is also met. For this reason, c_x can be substituted by c'_x. The cut c_w can able be substituted by c'_w such that $input(c'_w) = \{y, q\}$. So c_v is a excessive cut. $\{c_x, c_w\}$ is the set of Fo-Cuts of c_x, and $\{c'_x, c'_w\}$ is the set of Sub-Cuts of $\{c_x, c_w\}$.

B. Choice of Best Cut

The best cut is chosen at this phase. The best-cut is such a cut c that the most reduction of LUT is expected by elimination of c.

We define a gain for the cut $c \in C'$. A gain of cut $c \in C'$, denoted by $g(c)$, means the number of cuts $c' \in C'$ that would be eliminated if c is eliminated. A gain for c is calculated as following.

$$g(c) \ = \ \sum_{i \in input(c)} g'(c_i) + 1 \qquad (1)$$

$$g'(c_i) \ = \ \begin{cases} g(c_i) & (c_i \ is \ fanout \ free \ cut) \\ 0 & (c_i \ is \ not \ fanout \ free \ cut) \end{cases} \qquad (2)$$

A fanout free cut c^{ff} is the cut such $\mid fanout(LUT_{c^{ff}}) \mid = 1$. If a cut c is eliminated, and if there is any fanout free cut c_i^{ff} for $i \in input(c)$, c_i^{ff} is able to be eliminated because $fanout(LUT_{c_i^{ff}}) = \{LUT_c\}$.

According to the formula(1) and (2), the process traverse the LUT network $N(V_{C'}, E_{C'})$ from PIs to POs calculating gain of each cut in C'. The process holds the cut whose gain is maximum among the excessive cuts in C' as the best-cut.

C. Cut Elimination

The Fo-Cuts of the best-cut are substituted with Sub-Cuts, and the best-cut and fanout free cut that is predecessor of the best-cut are eliminated.

Cut Elimination is executed as the following. Each Fo-Cut c'_w and the best cut is eliminated from C'. The fanout free cut whose root is one of $input(c)$ of the eliminated cut c is recursively eliminated. Each Sub-Cut c''_w is added to C'. $N(V_{C'}, E_{C'})$ is generated according to the new C', and the level and required level of each node in $V_{C'}$ are calculated.

Figure 3 (b) shows the LUT network after the elimination of excessive cut c_v in figure 3 (a). The best-cutc_v and Fo-Cuts $\{c_x, c_w\}$ of c_v is removed from C', and the Sub-Cuts $\{c'_x, c'_w\}$ is added to C'.

V. Experimental Results

Cut Substitution algorithm has been implemented as a program using the C++ programming language within the Magus Logic Synthesis system of Kyushu University. We experimented to compare the Cut Substitution with one of $state\ of\ the\ arts$, DAOmap on the machine whose CPU is Intel Xeon $3.00GHz$ and memory size is $15GB$.

Cut Substitution has been implemented as the post-processing of DdMap. DdMap is very simple heuristics for technology mapping to generate a network whose depth is minimum, but whose area is not guaranteed to be minimum. A lot of the traditional algorithms use the heuristics that divide the area cost of $TFIG(v)$ by $\mid fanout(v) \mid$ to avoid that the area cost being redundantly summed along the reconvergent path.

TABLE I
COMPARISON WITH CUT SUBSTITUTION AND DAOMAP ON MCNC BENCHMARKS (K = 5)

bench	DaoMap		Dd	DdMap + CS			
marks	LUT	time	LUT	LUT	time	vs. Dd	vs. Da
C1908	150	0.16	159	143	0.2	90%	95%
C3540	368	0.28	366	325	0.35	89%	88%
C5315	438	0.51	448	425	0.56	95%	97%
C6288	560	2.70	894	631	3.70	71%	113%
C880	116	0.05	122	111	0.06	91%	96%
apex6	198	0.07	202	195	0.07	97%	98%
apex7	84	0.03	88	84	0.05	95%	100%
f51m	52	0.03	52	52	0.04	100%	100%
rot	385	0.15	379	359	0.16	95%	93%
vda	455	0.23	455	422	0.22	93%	93%
att10	935	0.57	932	882	0.43	95%	94%
att6	455	0.23	455	422	0.22	93%	93%
att8	619	0.22	598	598	0.17	100%	97%
C2670	205	0.27	208	190	0.25	91%	93%
des	2079	1.90	2140	1971	1.50	92%	95%
C7552	634	1.11	662	608	1.23	92%	96%
Total	7733	8.50	8160	7418	9.26	91%	96%

But they ignores the fanout from a node $w \in TFI(v)$ to a node $x \notin TFI(v)$, where $w \neq v$. For consideration of a fanout from $w \in TFI(v)$ to $x \notin TFI(v)$, DdMap defines weight on all of the edge in a subject graph, and propagates the product of area cost and the weight of edge along the path.

We compared our method and DAOmap on MCNC benchmarks and on ITC'99 benchmarks. Table I and Table II show the results on MCNC benchmarks and ITC'99 benchmarks, respectively. We omit the results for the small benchmarks for the constraint of page space. "Dd" and "DdMap + CS" mean Ddmap and our method, respectively. The columns labeled as "LUT" and "time" mean the number of nodes of LUT networks and run time in seconds, respectively. "vs. Dd", "vs. Da" are the ratios of the number of LUTs generated by our method to the number of LUTs generated by Ddmap, DAOmap, respectively. We omit the depth of LUT network, because the depth of LUT network generated by each program is equal if the benchmark is same.

For all of the benchmarks but C6288 on MCNC benchmarks, the combination of DdMap and Cut Substitution has generated LUT networks with less LUTs than those generated by DAOmap. The total number of LUTs of all LUT networks generated by our method is 9% less than that of all LUT networks generated by DAOmap. On MCNC benchmarks, DdMap generates LUT networks with almost all the same number of LUTs as that of DAOmap. Cut Substitution has reduced 9% LUTs of LUT networks generated by DdMap on average, and our method generated LUT networks with 4% less LUTs than those generated by DAOmap as a result for MCNC benchmarks. On ITC'99 benchmarks, DdMap generated LUT networks with more LUTs than those generated by DAOmap. Cut Substitution reduced 11% LUTs of LUT network generated by DdMap on averages, and our method generated LUT networks with 9% less LUTs than those generated by DAOmap as a result for ITC'99 benchmarks. The run time of our method is 3% shorter than that of DAOmap on averages.

TABLE II
COMPARISON WITH OUR METHOD AND DAOMAP ON ITC'99 BENCHMARKS (K = 5)

bench	DaoMap		Dd	DdMap + CS			
marks	LUT	time	LUT	LUT	time	vs. Dd	vs. Da
b14	2265	3.67	2390	2097	3.74	88%	93%
b14_1	1749	2.43	1823	1632	2.53	90%	93%
b15	2944	3.08	3175	2695	3.59	85%	92%
b15_1	3571	5.52	3438	3145	3.92	91%	88%
b17	9457	12.4	10208	8724	11.5	85%	92%
b17_1	10747	16.4	10578	9678	12.2	91%	90%
b20	4932	8.18	4824	4275	8.33	89%	87%
b20_1	3437	5.46	3484	3085	5.86	89%	90%
b21	5069	8.32	5066	4498	8.47	89%	89%
b21_1	3553	5.55	3603	3217	6.03	89%	91%
b22	6860	11.7	7149	6306	12.1	88%	92%
b22_1	5084	8.12	5363	4728	9.07	88%	93%
Total	59668	90.8	61101	54080	87.3	89%	91%

VI. CONCLUSION

In this paper we presented the post-processing, Cut Substitution, for technology mapping for LUT-based FPGAs to minimize area under depth constraint. The concept of Cut Substitution is to eliminate the excessive cuts directly by substituting specific cuts releted to the excessive cut with other cuts. Experimental results show that Cut Substitution, combined with the simple depth optimum heuristic algorithm, has generated networks with 9% smaller area than the networks generated by DAOmap on average. The run time of our method is 97% compared to DAOmap.

Future work includes examining the effect of Cut Substitution combined to other technology mapping algorithms.

ACKNOWLEDGEMENTS

This reseach is supported by CREST of JST.

REFERENCES

[1] Jason Cong, Y.Ding, "FlowMap: An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 1994.

[2] J.Cong, Y.Ding, "On Area/Depth Trade-off in LUT-Based FPGA Technology Mapping," IEEE Transactions on Very Large Scale Integration Systems, 1994.

[3] Richard L.Rudell, "Logic synthesis for VLSI design," Ph.D.thesis, University of California, Berkeley, 1989.

[4] Deming Chen, Jason Cong, "DAOmap: A Depth-optimal Area Optimization Mapping Algorithm for FPGA Designs," IEEE International Conference on Computer Aided Design, 2004.

[5] Maxim Teslenko, Elena Dubrova, "Hermes: LUT FPGA Technology Mapping Algorithm for Area Minimization with Optimal Depth," IEEE International Conference on Computer Aided Design, 2004.

[6] Touati, Herve Jacques, "Performance-oriented Technology Mapping," Ph.D.thesis, University of California, Berkeley, 1990

[7] J.Cong and Y.HWang, "Simultaneous Depth and Area Minimization in LUT-Based FPGA Mapping," International Symposium on Field Programmable Gate Arrays, 1995.

[8] Valavan Manohararajah, Stephen D. Brown, Zvonko G. Vranesic, "Heuristics for Area Minimization in LUT-based FPGA Technology Mapping, " IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.25, No.11, 2006