Methodology of Logic Synthesis for Implementation Using Heterogeneous LUT FPGAs

I.Lemberski
Riga Aviation University

Abstract

Logic synthesis method for heterogeneous LUT FPGAs implementation is proposed. As an example, XILINX4000 architecture is considered. The method takes XILINX4000 architectural features (heterogeneous LUTs of 3 and 4 inputs) into account and includes two step decomposition. In the first step, two-level logic representation is transformed into a graph of at most 4 fanin nodes (after this step, each node can be mapped onto 4 input LUT). In the second step, selected 4 fanin nodes are re-decomposed into 3 fanin nodes to ensure mapping onto 3 input LUTs. Re-decomposition task is formulated as substituting node two fanins for exactly one fanin.

1: Introduction

LUT FPGA-based logic synthesis methods developed don't consider architectural features of LUT FPGAs[1] (as exception, the paper [2] can be mentioned where decomposition methodology is oriented on XILINX3000 and 4000 structure). Instead of it, two-level logic is transformed into the set of sub-functions, where each one depends on the restricted number of inputs. Usually, this number doesn't exceed the number of LUT inputs. Therefore, each sub-function can be implemented onto one LUT. To optimize , methods of minimization of the number of sub-functions are considered (LUT-driven synthesis).

However, FPGAs such approach may be not optimal for FPGAs having several heterogeneous LUTs with specific connections between them.

In this paper, architecture - driven (instead of LUT-driven) approach to logic synthesis targeting XILINX4000 structure is offered. It takes heterogeneous LUT architecture (two type LUTs of 3 and 4 inputs) into account amd includes two step decomposition to ensure minimization of the number of logic blocks (rather than the number of LUTs) .

2 :XILINX4000 Architecture and Mapping

XILINX4000 logic structure bases on cooperating Configurable Logic Blocks (CLBs) (minimum 64 CLBs per module). A CLB consists of 3 look-up-tables (LUTs) (function generators), namely, LUT of type F, G and H (further,F-LUT,G-LUT,H-LUT)[3]. F-LUT and G-LUT can implement any function of up to 4 variables . H-LUT implements up to 3 input function, where two inputs are F-LUT and G-LUT outputs (internal inputs), but the third input is external. CLB has two external outputs . Any two of three LUT outputs can be connected to CLB outputs either directly or through flip-flops .

Let decomposed logic functions PO = f(PI) be represented by a directed acyclic graph D [4] : D = (PI, PO, G), where PI, PO - set of primary input and output nodes, G - set of nodes obtained as a result of decomposition. We call the node $g_{i-1} \in RI \cup G$ as a fanin of the node $g_i \in G \cup PO$, if there is an arc from g_{i-1} to g_i (fig.1). Let sup (g_i), $g_i \in G \cup PO$, be the set of g_i fanins. Suppose, that $|sup(g_i)| = 4$ for each $g_i \in G \cup PO$. Therefore, any two nodes can be mapped onto the same CLB (F- and G- LUTs) . We call such mapping as trivial. To reduce the number of CLBs, H-LUTs have to be used for implementation. Suppose:$|sup(g_i)| = 3$ (fig.1). In this case, three nodes (g_i , g_{i-1} , g_{i-2}) instead of two ones can be mapped onto the same CLB. As a result, the total number of CLBs is reduced.

Let τ be a CLB propagation delay. In case of trivial mapping, the critical path propagation delay d :$d=n\times\tau$, where n - number of CLBs (or what is the same, nodes) within the critical path. Suppose: $| sup(g_i) | = 3$ (fig.1) and g_{i-1} , g_i are in the critical path. The nodes g_{i-1} and g_i can be mapped onto the same CLB . As the CLB delay remains the same, the propagation delay within the critical path is reduced : $d= (n-1)*\tau$.

Our goal is to develop decomposition technique, to map onto H-LUTs as many as possible nodes (area

0-7695-0104-4/99 $10.00 © 1999 IEEE

optimization) or / and as many as possible nodes within the critical path (speed optimization).

To ensure optimal implementation we propose two step decomposition. In the first step, a graph D: $|sup(g_i)| \leq 4$, $g_i \in G \cup PO$ is obtained and optimized for area and speed using SIS tool [5]. In the second step, selected 4 fanin nodes are re-decomposed into 3 fanin nodes to map them onto H-LUTs.

3: Re-decomposition via Substituting

Given a 4 fanin node (fig1). To transform it into a 3 fanin one, we substitute its 2 fanins for an exactly one fanin while preserving functionality and acyclic.

Let $l(g_i)$ be a node g_i level (maximal number of CLBs between PI and g_i). To ensure reducing the critical path delay after substituting, we have to care for the level of the node g_j two nodes are substituted for. Following relation has to be valid: $l(g_j) \leq l(g_i) - 1$.

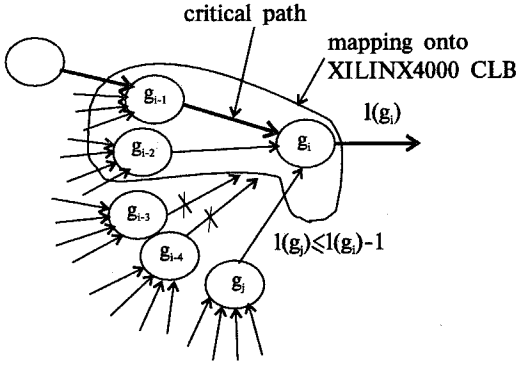

Fig.1. Substituting for a node

4: Experimental Results

Architecture driven re-decomposition procedure for speed and area optimization is implemented and tested on some FSM examples (fig.2). Results are compared with the trival implementation. For state encoding, the procedure [6] is used. In the fig.2 , one can see that the re-decomposition for speed reduces significantly the CLBs number within the critical path .

References

1. Cong, J,Ding ,Y. :Combinational Logic Synthesis for LUT Based Field Programmable Gate arrays . ACM Trans. on Design Automation of Electronic Systems, Vol.1, No. 2, April (1996) 145-204
2. Abouzeid, P., Babba, B., Crastes de Paulet, M., Saucier, G.:Input-Driven Partitioning Methods and Application to Synthesis on Table-Look-up-Based FPGA's. IEEE Trans. CAD, Vol. 12, No. 7, July (1993) 913-925
3. The Programmable Logic Data Book, XILINX (1994)
4. Murgai, R., Brayton, R., Sangiovanni-Vincentelli, A. :Logic Synthesis for Field - Programmable Gate Arrays. Kluwer Academic Publishers (1995)
5. Sentovich, E..et al: SIS: a System for Sequential Circuit Synthesis. Electronic Research Laboratory Memorandum No. UCB/ERL M92/41(1992)
6. Grass, W., Lemberski,I.: Support-Based State Encoding targeting FSM Optimal LUT FPGA Implementation.- Int.Workshop on Logic and Architecture Synthesis, Grenoble, France (1997) 97-104

XILINX XC4000						
Example	R-d for speed		R-d for area		Trivial	
	#CLB (c.p.)	#CLB (total)	#CLB (c.p.)	#CLB (total)	#CLB (c.p.)	#CLB (total)
ex2	2	10	3	10	4	12
ex3	2	5	4	4	4	5
ex4	2	13	3	13	3	14
ex5	2	4	2	4	3	5
ex6	3	26	4	25	5	32
ex7	2	4	3	4	3	4
bbara	2	12	3	12	5	13
beecount	2	7	4	7	4	8
dk14	2	15	3	15	4	17
coprodes	2	8	2	9	3	10
asic	2	24	4	23	5	26
Total:	23	128	35	124	43	146
Percentage :	53,5%	87,7%	81,4%	84,9%	100%	100%

R-d stands for re-decomposition
#CLB (c.p.) - CLBs number within the critical path
#CLB (total) - total number of CLBs needed for implementation

Fig.2. Experimental results