Technology Mapping using Fuzzy Logic *

Sasan Iman, Massoud Pedram
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089

Kamal Chaudhary
Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124

Abstract - This paper presents a placement-driven technology mapping procedure based on fuzzy delay curves. The fuzziness has been introduced to deal with the inherent vagueness in wiring loads (derived from a dynamically updated placement) and used by the mapper to calculate the signal arrival times. In the process we describe a number of fuzzy operations which are needed to generate the fuzzy delay curves and to select a minimum area mapping solution satisfying a set of timing constraints. This procedure has been implemented and the results are on average 1% and 26% (5% and 3%)better in terms of area and delay compared to a technology mapping procedure with zero (crisp) wire load values.

1 Introduction

In the past, researchers have separated logic synthesis from physical design and have relegated interconnect optimization to the physical design phase. However, with recent studies [10, 4] indicating that interconnections occupy more than half the total chip area and account for a significant part of the chip delay, it is appropriate that wiring be incorporated into the cost function for logic synthesis [1, 6]. This is because physical design which is expected to address these issues, comes much later in the design hierarchy. By then, many of the key architectural and structural decisions have been made, hence, limiting the capability of the physical design tools to generate the "best" solutions in terms of area and performance.

To elaborate on the importance of the wiring load, consider a two-input NAND gate driving an inverter gate through 0.2 cm of aluminum interconnect (2 μm wide, 0.5 μm thick, with a 1.0 μm thick field oxide beneath it). 0.2 cm is the expected length of a *local* interconnect line on a $2cm \times 2cm$ chip [2]. We calculate the rise time (to 50% of its final value) at the input of the inverter gate using two methods. One method ignores the capacitance of the interconnect line and uses

$$delay = \tau_g + R_g \times C_g = 0.4ns$$

where τ_g is the intrinsic gate delay, R_g is the on-resistance of the driver gate, and C_g is the input capacitance of the fanout gate. The second method [9] uses

$$delay = \tau_g + R_{out} \times (C_{input} + C_{unit} \times length) = 1.0ns$$

where C_{unit} is the interconnect capacitance per unit length and $length$ is the interconnect length. Gate and interconnect parameters are taken from data sheets for an industrial 1.0 micron ASIC library: $\tau_g = 0.3ns$, $R_g = 1K\Omega$, $C_g = 0.1pF$, $C_{unit} = 3.0pF/cm$. The delay calculations clearly show that the interconnect capacitance dominates the gate input capacitance.

In summary, with the existing technology, the capacitive term is dominated by the capacitance between the interconnection and substrate. For local aluminum lines, the resistive term is dominated by the on-resistance of the MOS transistor. As the chip dimension increases and the minimum feature size decreases, the interconnection capacitance bottoms at about 1 - 2 pF/cm while the input gate capacitance decreases. Therefore, the RC delay of interconnect lines will become even more dominant in the future.

Given a boolean network representing a combinational logic circuit optimized by technology independent synthesis procedures and a target library, the technology mapping process binds the nodes in the network to gates in the library such that the area of the final implementation is minimized and timing constraints are satisfied.

In [13] an approach is presented to solve the technology mapping problem for minimizing area under delay constraints. The authors first compute a range of "interesting" values for the required times at each node (by finding the minimum area and the minimum delay mapping solutions) and then divide this range into equal intervals. The best mapping solution for each of the required times are generated and stored at the node during a postorder traversal (from primary inputs to primary outputs) of the tree. The final mapping solution is generated during a preorder (from primary outputs toward primary inputs) traversal of the tree. In order to obtain high quality mapping solutions, this method requires a small time step resulting in large number of delay-area points.

[3] examines the problem of mapping a Boolean network to a circuit implementation using gates from a finite size cell library. The objective is to minimize the total gate area subject to constraints on signal arrival time at the primary outputs. This approach consists of two steps. In the first step, delay curves (that capture gate area – arrival time tradeoffs) at all nodes in the network are computed. In the second step, the mapping solution is generated based on the computed delay curves and the required times at the primary outputs. For a NAND-decomposed tree, subject to load calculation errors, this two step approach finds the **minimum** area mapping satisfying any delay constraint if such a solution exists. The algorithm has polynomial run time on a node-balanced tree and is easily extended to mapping a network modeled by a directed acyclic graph.

In [6] an attempt is made to increase the interaction between logic synthesis and technology mapping. The idea is to generate a "companion" placement solution for the circuit before it is mapped. This placement is then used to evaluate the cost of a matching gate during the mapping process. The placement is dynamically updated in order to maintain the correspondence between the logic and layout representations. In the end, a mapped network along with a placement solution are

*This research was supported in part by the NSF's Research Initiation Award under contract No. MIP92/11668.

generated. The placement solution is then globally relaxed in order to produce a feasible placement according to the target layout style (e.g., standard-cell or sea-of-gates). Using these techniques, circuits with smaller area and higher performance have been synthesized.

A difficulty with the placement-driven approach is that after mapping, the network is very different from the one we started with. This makes the wiring estimation process during mapping inherently imprecise. The technology mapping results are therefore sensitive to the methods used for estimating the wiring and updating the placement. Fuzzy theory provides an effective method to remedy this difficulty by reducing the reliance on crisp wire values calculated based on the placement solution. Instead, the placement merely provides information about the wiring cost either in the form of a range of wire length values or in the form of a qualitative classification of wire length values (long, medium, or short).

In this paper we describe a placement-driven technology mapping procedure based on fuzzy logic. We start by calculating fuzzy locations for the nodes in the network. Using these fuzzy locations, we obtain a fuzzy wiring load for each gate which will in turn be used to calculate a fuzzy arrival time at the output of each gate. Based on this information we calculate fuzzy area-delay curves at all nodes using a postorder traversal of the network and then find a minimum area solution satisfying timing constraints during a preorder traversal of the network.

The same fuzzy logic framework can be applied to many other problems in logic synthesis, timing and power analysis and physical design where some parameter of interest or the objective function are imprecisely defined. The technology mapping procedure presented in this paper only serves as an example (see section 6).

The flow of the paper is as follows. In section 2 we review the process of technology mapping using area-delay curves. This process will be extended to include the fuzzy wiring load. In section 3, we describe the principles of fuzzy theory. Extensions of arithmetic operations in the crisp domain to the fuzzy domain are also presented. In section 4 details of fuzzy area-delay curve computation and fuzzy gate selection are presented. Experimental results and concluding remarks are given in sections 5 and 6.

2 Technology Mapping using Area Delay Curves

The technology mapping procedure used in this paper is based on the AD_MAP approach given in [3]. In the remainder of this section, we give an overview of that approach.

With each node in the network, we store a delay curve. A point on the delay curve represents the arrival time at the output of the node and the total gate area which is required to map its transitive fanin cone up to (and including) the node. In addition to the area and delay value, the matching gate and input bindings for the match are also stored with each point on the curve. Points on the curve represent various mapping solutions with different tradeoffs between area and speed. We are interested in a mapping with minimum area satisfying delay requirements. Consequently, we can drop point P_1 on the curve if there exists another point P_2 on the curve with lower area but equal or lower delay. By dropping inferior points, the delay curve can always be made monotonically non-increasing without loss of optimality.

The technology mapping procedure consists of two graph traversal steps. Initially a postorder traversal of the NAND-decomposed network is performed, where for each node n and for each gate g matching at n (a candidate match), a new delay curve is produced by appropriately merging the delay curves at the $inputs(n, g)$. The delay curves for successive gates g matching at n are then merged by applying a *lower-bound merge* operation on the corresponding delay curves. At a given node n, the resulting delay curve will describe the arrival time-area tradeoffs in propagating a signal from the network inputs to the output of n. The delay curve computation and merging are performed recursively until a circuit output is reached. The set of (t, a) pairs corresponding to the composite delay curve at the circuit output will define a set of arrival time-area tradeoffs for the user to choose from.

Given the required time t at the circuit output, a suitable (t, a) point on the delay curve for the circuit output is chosen. The gate g matching at the circuit output which corresponds to this point and its inputs are thus identified. The required times t_i at the inputs are computed from t, g, and the fact that these inputs must now drive gate g. The preorder traversal resumes at inputs of g where t_i is the constraining factor and a matching gate g_i with minimum a_i satisfying t_i is sought.

In the above technology mapping procedure, all parameters are crisp numbers. In our approach, we model the unknown wire loads as fuzzy numbers. We thus generate fuzzy delay curves during a postorder traversal of the subject network using fuzzy arithmetic operations. In the process, it becomes necessary to remove inferior points from these curves. This will require sorting of the area-delay points. Fuzzy decision making is used at this stage to rank the points based on their area and delay values. Gates are assigned to nodes in the circuit during the preorder traversal of the network using fuzzy decision making.

In order to be able to compute the fuzzy delay curves, it is necessary to generalize the operations used by the AD_MAP to the fuzzy domain. This will require a valid and efficient representation of the fuzzy numbers in addition to methods which will assist in operating on and making decisions based on the fuzzy parameters as explained in the next section.

3 Fuzzy Logic

In general, it is difficult to model the real world by a precise model. One difficulty is that in real life problems, parameters are not exactly known. We might plan to minimize power consumption in a circuit where the switching rate of the circuit input is "approximately 0.25". Another difficulty is that in many cases goals are not clearly expressed. For example, the goal of an optimization process might be to achieve "a delay of essentially 10 nanoseconds or less". Fuzzy theory helps us deal with this imprecision in parameters and goals. These are represented by *fuzzy numbers* and *fuzzy goals*.

In the following section, extensions of the crisp arithmetic operations to fuzzy domain are described.

3.1 Fuzzy Numbers

A fuzzy number $\tilde{n}$ is defined by a membership function $\mu_{\tilde{n}}(x)$ where x is a real number [1]. The value of the membership function represents the possibility of the event represented by this fuzzy number to assume a value of x. The membership function is normalized such that $0 \leq \mu_{\tilde{n}}(x) \leq 1$. The term "possibility" is used to emphasize the fact that the value of the membership function is not a probability value. In general if an event a has probability p_a and event b has probability p_b, then probability of either event is $p_a + p_b$ and probability of both events is $p_a.p_b$. On the other hand, if p_a and p_b represent the possibility of these events, then the possibility of either event is $Max(p_a, p_b)$ and possibility of both events is $Min(p_a, p_b)$.

For practical purposes it is generally more appropriate to resort to a specific kind of membership function. Indeed, triangular fuzzy numbers [16] are often used.

[1] We will represent *fuzzy* numbers with ~ to differentiate them from *crisp* numbers.

Figure 2: Fuzzy Absolute Value Opera

Given fuzzy numbers $\tilde{n}(x) = (m, l, r)$, we define bers $\tilde{n}_{pos}(x)$ and $\tilde{n}_{neg}(x)$ as follows:

$$\mu_{\tilde{n}_{pos}}(x) = \begin{cases} \mu_{\tilde{n}}(x) & x \geq 0 \\ 0 & x < 0. \end{cases}$$

$$\mu_{\tilde{n}_{neg}}(x) = \begin{cases} \mu_{\tilde{n}}(-x) & x \geq 0 \\ 0 & x < 0. \end{cases}$$

Then $abs\_value(\tilde{n}(x)) = max(\tilde{n}_{pos}(x), \tilde{n}_{neg}(x))$ ple of this calculation is shown in Figure 2 wh represents the triangular approximation of the fu

Figure 4: Fuzzy Max Merge Op

Given fuzzy numbers $\tilde{n_1} = (m_1, l_1, r_1)$ and

$$\tilde{n}_1 \geq \tilde{n}_2 \Leftrightarrow D_{\tilde{n}_1} \geq D_{\tilde{n}_2}$$

where $D_{\tilde{n}} = m + (r - l)/2$. Using this co
a total ordering can be imposed on a set o
numbers.

4 Fuzzy Delay Curves

For submicron technologies, the effect o
circuit delay is of more importance than its e
area. Therefore, we only consider the forme
latter effect can be easily captured in a simi

The delay curve at each node now consis
inferior points $\tilde{P} = (\tilde{t}, a)$ where $\tilde{t}$ is a fuzzy
ing delay, and a is the crisp area. Let the l
of a node n be represented as a fuzzy numbe
the output consists of two components: the g
fanout nodes and the wiring load. The gate
at the output is not known because the outp

Figure 5: Fuzzy Node Location

Once the fuzzy coordinates of a node and its fanouts are known, the wire length can be efficiently calculated by using a number of models. These models include the star connection model, the single trunk Steiner tree model and the enclosing rectangle approximation. In the following we use the latter model as it is accurate, yet easy to compute. We use the fz_max and fz_min operations to find a fuzzy bounding box for the node and its fanouts. This bounding box is speci-

goals. This is du
time at the outp
fanins are obtain
from the required
the gate is fuzzy
the gate will bec
for signals in the
about the gate d

The network is
cone refers to a
fanin. Since logic
a node to be in t
already been pro
to arrive at a noc
the arrival time
time required at
However if the m
arrival time for c
until a node is re
the primary inpu

After each con
each of the remai
that the current
network. Conseq
information will

5 Results

The procedure
program called FZ
of the SIS_MAP [
but generates de
optimized *blif* fi
cuits were first c
were then decom
SIS_MAP, PL_MAP
using GORDIANI
[7]. All results ar
presents the tota
technology mapp
with respect to th
show on average
area and delay c
load values. Tab
Station II with 6
PL_MAP is on ave
FZ_MAP is 2 time

Table 3 presen
ter palcement an
ment and routing
ter in terms of ar
These results sho
to improve on bo
values.

activity calculation under a real delay model (which accounts for hazards) cannot be performed exactly as it requires a very time consuming and memory intensive symbolic simulation [5]. It can, however, be approximately performed, say, using the tagged probabilistic simulation [14]. Fuzzy switching activities can be calculated and used to derive fuzzy power-delay curves. Other applications of the fuzzy framework proposed here, can be found in timing analysis, common subexpression extraction, etc.

References

[1] P. Abouzeid, K. Sakouti, G. Saucier, and F. Poirot. Multilevel synthesis minimizing the routing factor. In *Proceedings of the 27th Design Automation Conference*, pages 365–368, June 1990.

[2] H. B. Bakoglu. *Circuits, interconnections, and packaging for VLSI*. Addison-Wesley, 1990.

[3] K. Chaudhary and M. Pedram. A near-optimal algorithm for technology mapping minimizing area under delay constraints. Proceedings of the 29th Design Automation Conference, June 1992.

[4] Y. A. El-Mansy and W. M. Siu. MOS technology advances. In G. Rabbat, editor, *Handbook of Advanced Semiconductor Technology and Computer Systems*, pages 229–259. Van Nostrand-Reinhold, Princeton, New Jersey, 1988.

[5] A. A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of average switching activity in combinational and sequential circuits. In *Proceedings of the 29th Design Automation Conference*, pages 253–259, June 1992.

[6] M. Pedram and N. Bhat. Layout driven technology mapping. In *Proceedings of the 28th Design Automation Conference*, pages 99–105, June 1991.

[7] J. Reed, A. Sangiovanni-Vincentelli, and M. Santamauro. A new symbolic channel router: YACR2. *IEEE Trans. on Computer-Aided Design*, 4(3):208–219, March 1985.

[8] M. Roubens. Inequality constraints between fuzzy numbers and their use in mathematical programming. In R. Slowinski and J. Tegham, editors, *Stochastic versus Fuzzy approaches to multiobjective Mathematical Programming under Uncertainty*. Kluwer, Dordrecht, The Netherlands, 1990.

[9] T. Sakurai. Approximation of wiring delays in MOSFET LSI. IEEE *Journal of Solid State Circuits*, SC-18(4):418–426, August 1983.

[10] K. C. Saraswat and F. Mohammadi. Effect of scaling of interconnections on the time delay of VLSI circuits. IEEE *Transactions on Electron Devices*, ED-29:645–650, 1982.

[11] H. Savoj and H. Y. Wang. Improved scripts in MIS-II for logic minimization of combinational circuits. In *Proceedings of the International Workshop on Logic Synthesis*, May 1991.

[12] G. Sigl, K. Doll, and F. M. Johannes. Analytical placement: A linear or a quadratic objective function? In *Proceedings of the 28th Design Automation Conference*, pages 427–432, June 1991.

[13] H. J. Touati, C. W. Moon, R. K. Brayton, and A. Wang. Performance-oriented technology mapping. In *Proceedings of the Sixth M.I.T. Conference on Advanced Research in VLSI*, pages 79–97, April 1990.

[14] C. Y. Tsui, M. Pedram, and A. Despain. Efficient estimation of dynamic power dissipation under a real delay model. In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 224–228, Nov 1993.

[15] C. Y. Tsui, M. Pedram, and A. Despain. Technology decomposition and mapping targeting low power dissipation. In *Proceedings of the 30th Design Automation Conference*, pages 68–73, June 1993.

[16] H.-J. Zimmerman. *Fuzzy Set Theory*. Kluwer Academic Publishers, Norwell, Mass., 1991.

Ex.	Post Mapping (Timing Mode)					
	SIS_MAP		PL_MAP		FZ_MAP	
	area mm^2	delay ns	area mm^2	delay ns	area mm^2	delay ns
9symml	0.21	21.41	0.87	0.74	0.78	0.82
C1908	0.55	35.58	1.03	0.89	0.96	0.87
C2670	0.84	29.56	0.73	0.78	0.74	0.72
C880	0.42	45.60	0.93	0.75	1.03	0.66
apex6	0.85	24.50	0.75	0.71	0.77	0.72
b9	0.14	9.14	1.04	0.55	1.01	0.56
rot	0.72	27.33	0.92	0.77	0.91	0.73
apex7	0.26	16.92	0.94	0.80	0.99	0.72
count	0.14	27.87	0.99	0.45	0.92	0.48
frg1	0.15	9.72	0.81	0.88	0.82	0.84
misex3	0.64	24.89	0.91	0.81	0.91	0.86
ritex	0.06	8.64	1.25	0.79	1.18	0.81
ttt2	0.22	19.26	0.95	0.58	0.89	0.63
x1	0.34	11.56	0.89	0.72	0.89	0.71
x4	0.71	14.89	0.84	1.15	0.79	1.15
Avg.	1.00	1.00	0.93	0.77	0.91	0.76

Table 1: Normalized Post Mapping results in timing mode

Ex.	SIS_MAP	PL_MAP	FZ_MAP
	s	s	s
9symml	9.4	44.40	104.80
C1908	21.8	213.40	553.06
C2670	32.6	222.00	552.35
C880	18.1	105.50	253.52
apex6	32.5	116.20	231.78
b9	6.3	9.80	10.20
rot	29.0	122.50	255.83
apex7	10.4	20.91	53.81
count	6.9	12.81	36.37
frg1	7.3	10.60	24.67
misex3	25.1	174.51	453.56
ritex	3.9	9.00	21.53
ttt2	10.5	23.03	57.43
x1	14.8	62.50	163.94
x4	26.0	62.50	163.94

Table 2: CPU times for technology mapping

Ex.	Post Layout (Timing Mode)					
	SIS_MAP		PL_MAP		FZ_MAP	
	area mm^2	delay ns	area mm^2	delay ns	area mm^2	delay ns
9symml	0.70	28.49	0.95	0.81	0.83	0.82
C1908	2.03	50.05	1.13	0.90	1.01	0.88
C2670	5.15	45.35	0.73	0.78	0.76	0.69
C880	1.55	59.45	0.93	0.75	1.04	0.69
apex6	3.48	37.87	0.84	0.67	0.84	0.64
b9	0.42	11.58	1.21	0.67	1.17	0.59
rot	4.17	42.20	1.03	0.73	0.86	0.70
apex7	0.89	23.25	1.69	0.72	1.71	0.68
count	0.39	33.95	1.07	0.46	0.95	0.47
frg1	0.54	13.24	0.91	0.86	0.84	0.80
misex3	2.88	42.26	0.98	0.76	0.92	0.76
ritex	0.15	9.50	1.23	0.84	1.22	0.82
ttt2	0.70	25.18	1.00	0.56	0.91	0.59
x1	1.87	17.84	1.00	0.72	1.00	0.70
x4	2.48	23.14	0.91	1.06	0.86	1.08
Avg.	1.00	1.00	1.05	0.77	0.99	0.74

Table 3: Normalized Post Layout results