Gate-Size Selection for Standard Cell Libraries

Frederik Beeftink
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Prabhakar Kudva David Kung Leon Stok
IBM TJ Watson Research Center
Yorktown Heights, NY

Abstract

This paper presents an algorithm to select a good set of gate sizes for the primitive gates of a standard cell library. A measurement error on a gate is defined to quantify the discrepancy resulting from replacing the size required by a synthesis sizing algorithm with a size available in a discrete cell library. The criterion for gate size selection is a set of gate sizes that minimizes the cumulative error of a prescribed measurement. Optimal solutions to the gate size selection problem targetting size and delay measurements are presented for cases when the probability distribution and the delay equations are simple. A realistic probability distribution is obtained using a sample space of gates derived from a group of designs that is synthesized under the semi-custom synthesis methodology [1]. A "delay-match" (minimizing delay error) and a "size-match" (minimizing size error) set of gate sizes are obtained numerically, and are subsequently realized as discrete cell libraries. The previous group of designs are synthesized using the two selected cell libraries and two other cell libraries, one with "equal-spacing" of cell sizes and the other with "exponential-spacing" of cell sizes. The "size-match" library gives the best overall slack and area results.

1 Introduction

It is well known that the quality of designs produced by logic synthesis tools depends on the library that is being used [2]. It is therefore important to develop such a standard cell library carefully. Creating a good library involves a proper selection of the set of basic functions (primitive gates) and the set of sizes for each of the primitive gates.

Recently, some efforts have been made to develop a methodology for gate-size selection. Reference [3] primarily focuses on area optimization. In [2], the authors concludes that it is advantageous to provide a variety of sizes for each primitive gate. However, they do not propose a method for determining these sizes. A more systematic approach is described in [4], where it is assumed that an arbitrary gate size can be approximated by a parallel combination (dotting) of smaller sizes. An algorithm is presented to select a set of base sizes to cover the widest range. However, current delay fault testing methodologies cannot handle such dotting configurations, making the algorithm impractical.

This paper addresses gate-size selection for primitive gates by developing a theoretical framework and presenting a numerical approach to solve the problem in practice. In principle, if we make available a large number of sizes for each cell type in the library then an "equal-spacing" size selection is adequate since there will always be a cell size close to a required gate size. However a large number of gate sizes is impractical because it puts a heavy burden on managing the data volume of the library and on the gate sizing algorithms. We show that our selection algorithm leads to better synthesis results than "rule-of-thumb" size selections when the number of available sizes is small.

The paper is organized as follows. We first present a theoretical analysis by defining an error function with respect to a measurement. This error function quantifies the discrepancy in the measurement when a required gate-size is replaced by an available size in the library. We assume the existence of a sample space of gates and the existence of a probability distribution over gate size and other attributes. The measurement error is then the expectation of the error function over the sample space. The gate-size selection problem is to find the set of gate sizes which minimize the measurement error. Optimal solutions to the gate-size selection problem are given for a size measurement and a delay measurement where the probability distribution and delay equation are simple. To put the theory into practice, we extract a realistic probability distribution

from a sample of designs optimized under the semi-custom synthesis methodology [1] [5] which is briefly discussed in section 4. We obtain two sets of gate sizes numerically: a "delay-match" set by minimizing the delay measurement error and a "size-match" set by minimizing the size measurement error, both with respect to the extracted probability distribution. We compare the results of gate-sizing among the following libraries: the "delay-match" library, the "size-match" library, a library with "equal-spacing" of sizes and a library with "exponential-spacing" of sizes in section 5.

2 The Gate-Size Selection Problem

The gate-size selection problem is motivated as follows. In a technology library, T, each gate type can have a finite number of sizes. Let g be a gate of type t whose set of available sizes is $\{S_1, S_2, \ldots, S_n\}$. A measurement on g, M, is a function

$$M : G \rightarrow R$$

where G is a set of gates and R is the set of real numbers. The size, $S(g)$, of g, the load, $L(g)$, that g drives and the average (over all delay arcs) delay of g are examples of such measurements. A measurement can be a function of other measurements, e.g., the average delay of g is a function of $S(g)$ and $L(g)$. Since we are concerned with size selection, we only consider measurements of the form $M(S(g), \vec{A}(g))$ where $\vec{A}(g)$ represent zero or more attributes (other than size) of g.

Let C be the set of all the designs that targets technology T and let G be the set of gates contained in C. We assume that the designs in C are synthesized "optimally" and that $S(g)$ and $\vec{A}(g)$ are the continuous size and other attribute measurements of the gates in these "optimized" designs. Since T can only offer finite number of sizes for each gate type, $S(g)$ is replaced by the best match from $\{S_1, S_2, \ldots, S_n\}$. This substitution will lead to a discrepancy in a measurement, M, and the error incurred in the measurement is

$$\varepsilon(M, g) = \min_{i \in \{1, \ldots, n\}} |M(S_i, \vec{A}(g)) - M(S(g), \vec{A}(g))| \tag{1}$$

The best match is therefore the size S_j at which the minimum of eqn. 1 is attained. When a size measurement is considered, the minimum will occur when the library cell with the closest size, S_j, to g is chosen, so that $\varepsilon(S, g) = |S(g) - S_j|$. Intuitively, the gate-size selection problem is to find, for each type of gates, the set $\{S_1, S_2, \ldots, S_n\}$ such that the cumulative error of some measurement is minimized.

We now define the gate-size selection problem. The inputs to the gate-size selection problem are

1. a technology library T and a gate type t
2. a minimum size $S_{\min}$ and a maximum size $S_{\max}$ for type t
3. the number of sizes, n, available for gate type t
4. a measurement M
5. a set of gates, G, with required size and attributes associated with each gate in G.

The objective of gate-size selection is to find a set of sizes, $\{S_1, S_2, \ldots, S_n\}$, where $S_1 = S_{\min}$ and $S_n = S_{\max}$ such that

$$\sum_{g \in G} \varepsilon(M, g) \tag{2}$$

is minimized.

The set G can be interpreted as a sample space and the size and other attribute measurements can be interpreted as random variables. The probability density function with respect to these random variables is defined to be

$$P(S, \vec{A}) = \frac{|H|}{|G|}$$

where H is the set of gates in G with size S and attributes $\vec{A}$. Then eqn. 2 is equivalent to the expectation of $\varepsilon(M)$ in this probability space.

3 Solvable Simple Cases

In this section we present optimal solutions to two cases in which the probability density function and the measurement function are simple. We restrict the relevant attributes of a gate to be the size of the gate, S, and the load, L, that it drives. It is assumed that the variables S and L are continuous and that they lie in the intervals $[S_{min}, S_{max}]$ and $[L_{min}, L_{max}]$. The probability density function is assumed to be separable, i.e.

$$P(S, L) = P_1(S) \cdot P_2(L)$$

and uniform, i.e.

$$P_1(S) = \frac{1}{S_{max} - S_{min}} \tag{3}$$

$$P_2(L) = \frac{1}{L_{max} - L_{min}}. \tag{4}$$

We first consider the size selection problem targetting the size measurement. Fig. 1 depicts the error

function for the size measurement where the expectation of $\varepsilon(S)$ is the area of the $n-1$ right-angle isosceles triangles

$$\frac{1}{4}\sum_{i=2}^{n}(S_i - S_{i-1})^2$$

The minimal solution occurs when

$$S_i - S_{i-1} = S_{i+1} - S_i$$

i.e. the optimal size selection is one in which the selected sizes are evenly distributed in the interval $[S_{min}, S_{max}]$.

A more interesting case is the size selection problem targetting the delay measurement. We assume the delay D of a gate g is given by the simple equation

$$D = p + k \cdot \frac{L}{S}$$

where p and k are constants that depends on cell types, L is the load that g drives and S is the size of g. The expectation of $\varepsilon(D)$ is given by

$$k \int_{L_{min}}^{L_{max}} \left(\sum_{i=1}^{n-1} \int_{S_i}^{S_{i,i+1}} (\frac{1}{S_i} - \frac{1}{S}) L \cdot P(S,L) dS + \sum_{i=1}^{n-1} \int_{S_{i,i+1}}^{S_{i+1}} (\frac{1}{S} - \frac{1}{S_{i+1}}) L \cdot P(S,L) dS \right) dL \tag{5}$$

where $S_{i,i+1} = 2S_iS_{i+1}/(S_i + S_{i+1})$. After performing the integration, the expression becomes

$$\frac{(L_{min} + L_{max})}{2(S_{max} - S_{min})} \left(\sum_{i=1}^{n-1} \ln \frac{(S_i + S_{i+1})^2}{4S_iS_{i+1}} - 2(n-1)\ln 2 \right) \tag{6}$$

The expectation of $\varepsilon(D)$ is minimized when

$$S_i = \sqrt{S_{i-1}S_{i+1}} \qquad 2 \le i \le n-1.$$

It immediately follows that the set of optimal sizes satisfies a geometric progression

$$S_i = S_1 (\frac{S_n}{S_1})^{\frac{i-1}{n-1}} \qquad 1 \le i \le n. \tag{7}$$

In a realistic sample of gates, the probability density function over size and loads is seldom separable and uniform. To obtain a realistic probability distribution, we need a sample of optimized designs in which gates take on continuous sizes. In reality, cell libraries have a finite number of discrete sizes since each cell in the library is characterized for timing and other measurements before synthesis is performed. Fortunately, a semi-custom library allows a continuous parameterization of gates without an underlying continuously sized cell library. We will outline the salient features of the semi-custom methodology in the next section.

4 The Semi-Custom Synthesis Methodology

The goal of the semi-custom synthesis methodology is to enable synthesis to take advantage of some of the flexibilities of custom design. To effectively exploit these added degrees of freedom, changes in the library design and the synthesis paradigm are required. One flexibility which enhances synthesis results is increasing the number of sizes available in the library. However, most traditional sizing algorithms are sensitive to the number of sizes and become inefficient as the number of available sizes is large. A judicious library design which can circumvent this combinatorial problem is to manipulate cell topologies such that each primitive cell in the library obeys a delay equation which satisfies the following condition

1. Given the gain, the input slew and the cell type of a gate, the delay and the output slew of the gate are determined independently of the load that it drives.

The *gain* of a gate is defined to be L/S where L is the load driven by the gate and S is the size of the gate which is chosen to be the average input pin capacitance of the gate. Such a library is referred to as a semi-custom library [1]. One immediate consequence is that gates of a specific cell type are described by one delay equation which is independent of the size of the gates. This unified delay description via the concept of gain allows gate sizing to be performed without combinatorial search over the available sizes and therefore enables gate sizing to handle libraries with large number of sizes.

The cell library can be discrete, i.e., each cell type can contain a finite number of sizes as long as condition 1 is satisfied. However, the gates in a circuit are parameterized by *gain* instead of the discrete sizes. The size of a gate is load dependent and is given by $L/gain$. Therefore size computation of a gate requires a propagation of the primary output loads through the gate's fanout cone. Continuous sizing is performed by assigning gain values to each gate in the circuit such that the timing assertions and the capacitance assertions are met. The size of a gate is intrinsically continuous so if the library is discrete, a discretization process [6] will be used to match a continuously sized gate with a cell available in the library. Our experiments target such a discrete semi-custom library.

5 Experimental Results

We apply our gate-size selection algorithm to a discrete semi-custom library. The delay equation of each

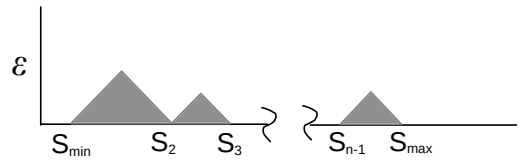

Figure 1: The error function of size measurement.

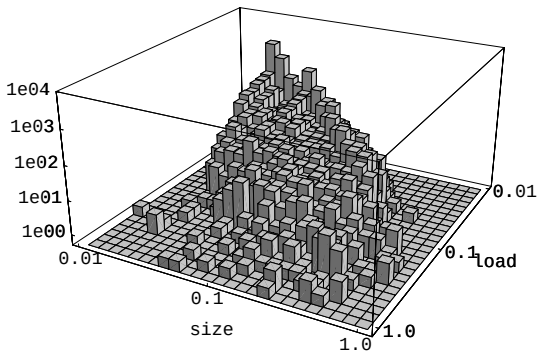

Figure 2: The size-load frequency density function of an inverter.

cell satisfies condition 1 but there are a finite number of cells for each cell type. Given a cell type, a target number of sizes, n, and a range of size $[S_{min}, S_{max}]$ for that cell type, our goal is to determine the corresponding "delay-match" and "size-match" set of sizes $\{S_1, S_2, \ldots, S_n\}$. The sample space of gates is derived from the gates of a group of designs in a design project that utilizes the semi-custom library. These designs are then synthesized by our logic synthesis system to provide a "near-optimal" circuit topology for the starting point of our experiment.

The extraction of the probability density functions proceeds as follows. Since the measurements in question are delay and size, the basic random variables required are the size and load variables. The initial synthesized circuits contain gates with discrete sizes since the semi-custom library is discrete. An initial gain of each gate is computed based on the current discrete sizes of the gate and its successors. Continuous gate sizing is then performed on each circuit by tuning the gain of each gate to improve timing and minimize area while satisfying size and slew constraints. These tuned circuits are referred to as the continuously-sized circuits. The resulting size (continuous) and output load of each gate are the required size and load measurements which are used to generate the probaility density functions, $P(S, L)$.

The frequency density functions ($N \cdot P(S, L)$ where

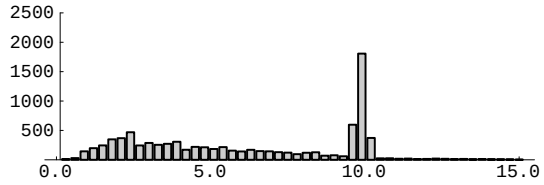

Figure 3: The gain frequency density function of an inverter.

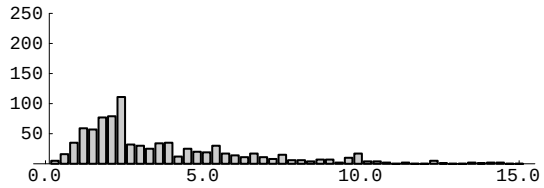

Figure 4: The gain frequency density function of an inverter sampled from a critical region.

N is the number of samples) for different attributes (e.g. size, load) for the inverter are displayed. The plots for other gate types, e.g. 2 input NAND, exhibit similar behavior. For the load and size plots, the frequency axes are plotted logarithmically and the load and size axes are normalized. For the gain plot (figs. 3 and 4), a linear scale is used for the frequency axes and the gain axes reflect their actual values. The frequency density function for the inverter is shown in fig. 2 where a correlation between size and load is evident and the distribution strongly peaks at low values. Therefore the geometric progression solution (which is a popular size selection among library designers) is unlikely to be the best size selection. Fig. 3 shows the frequency density plot for gain of the inverter cell type. The peak at gain = 10 is due to a gain cutoff to avoid slew violation. Fig. 4 shows the frequency density plot for the gain of the inverter cell type when gates are sampled from critical regions. In this case, the most frequent gain occurs in the gain interval $[2.0, 3.0]$ which is in the neighborhood of e (Euler's constant), the optimal gain [7] for a chain of inverters. Designers have been using this number as a rule of thumb to speed up critical paths.

The load distribution of the inverter is shown in fig. 5 which is strongly peaked at low values and then falls off rapidly. The distribution can be well approximated by a Weibull distribution which is not surprising since the load distribution might be strongly correlated to the wire length distribution. The size distribution (fig. 7) exhibits similar behavior to the load distribution since size equals load/gain and gain varies relatively slowly.

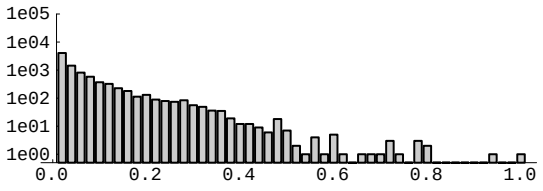

Figure 5: The load frequency density function of an inverter.

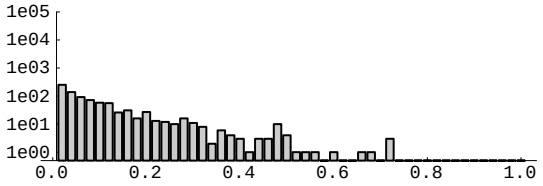

Figure 6: The load frequency density function of an inverter sampled from a critical region.

The corresponding distributions when the inverters are sampled from critical regions are shown in figs. 6 and 8. As expected the number of gates with low values decreases substantially in both cases.

The numerical optimization proceeds as follows. For the "delay-match" set, we minimize for each cell type the delay error

$$\int_{L_{min}}^{L_{max}} \int_{S_{min}}^{S_{max}} \min_{i \in \{1,\ldots,n\}} |d(S_i, L) - d(S, L)| \cdot P_c(S, L) dS dL \tag{8}$$

where $P_c(S, L)$ is the probability density function sampled with respect to critical regions. For the "size-match" set, we minimize for each cell type the size error

$$\int_{L_{min}}^{L_{max}} \int_{S_{min}}^{S_{max}} \min_{i \in \{1,\ldots,n\}} |S_i - S| \cdot P(S, L) dS dL \tag{9}$$

where $P(S, L)$ is the probability density function sampled with respect to entire circuits. $\{S_2, \ldots, S_{n-1}\}$ is

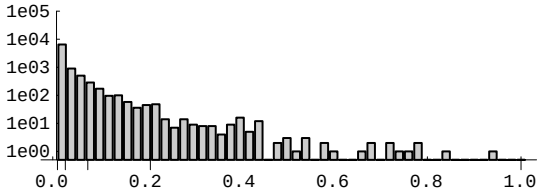

Figure 7: The size frequency density function of an inverter.

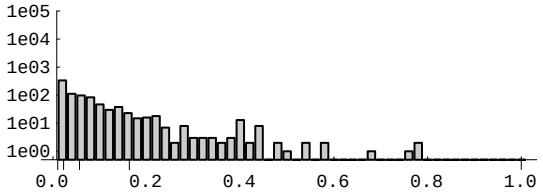

Figure 8: The size frequency density function of an inverter sampled from a critical region.

the set of variables over which the minimzation of the errors is performed.

A "delay-match" library and an "size-match" library are generated using these optimal sets of sizes. A third library, referred to as the "equal-spacing" library, is generated with "equal-spacing" of sizes and a fourth library, referred to as the "exponential-spacing" library, is generated according to eqn. 7. We choose $n = 5$ so there are 5 sizes available for each cell type in each of the four libraries. The size selections for the "size-match" library and the "delay-match" library are indicated by markings on the size axis (x-axis) in figs. 7 and 8 respectively. In both cases small sizes are favored since the the frequency density function is peaked at low sizes. The continuously-sized circuits are then mapped into the "delay-match" library, the "size-match" library, the "equal-spacing" library and the "exponential-spacing" library. A continuously-sized design should obtain the closest match in size by using the "size-match" library. Since a close match in size also results in a close match (may not be the best) in delay, the "size-match" library should give good delay values. The "delay-match" library is an attempt at fine-tuning the sizes for delay. However, the slack of a circuit is not an average measurement, it depends strongly on the circuit topology. Therefore minimizing the delay error which is an average quantity does not necessarily lead to the best sizes for slack. Despite the fact that only critical regions are used in the sample space, the result need not be optimal in practice.

The comparisons on 9 designs are presented in table 1. The "delay-match" library and the "size-match" library in general gives better synthesis results than the "equal-spacing" library and the "exponential-spacing" library. However, the "size-match" library seems to be the library of choice since it is clearly better in area than the other three and slightly better in timing than the "delay-match" library, which is not surprising in light of the above discussion.

We also studied the effect of varying n, the number of sizes available for each cell type. The minimized delay measurement error is plotted against n in fig 9.

Table 1: Synthesis results for four libraries.

	Eq Spaced		Expo		SizeMatch		DelayMatch	
Cir	*area*	*slack*	*area*	*slack*	*area*	*slack*	*area*	*slack*
d1	3573	-0.085	3647	-0.085	3529	-0.085	3639	-0.085
d2	4125	-0.188	4170	-0.063	4147	0.027	4252	-0.042
d3	10966	-0.147	11154	-0.137	10504	-0.126	10917	-0.111
d4	17327	0.006	17310	0.027	17244	0.031	17325	0.010
d5	13977	-0.149	14111	-0.092	13890	0.000	13928	-0.068
d6	16615	-0.094	16683	-0.103	16257	-0.074	16656	-0.113
d7	24350	-0.152	24531	-0.265	24007	-0.175	24374	-0.171
d8	38921	-0.090	39333	-0.089	38396	-0.099	39562	-0.098
d9	35517	-0.195	35810	-0.087	35646	-0.061	35952	-0.053
total	165371	-1.094	166749	-0.894	163620	-0.562	166605	-0.731

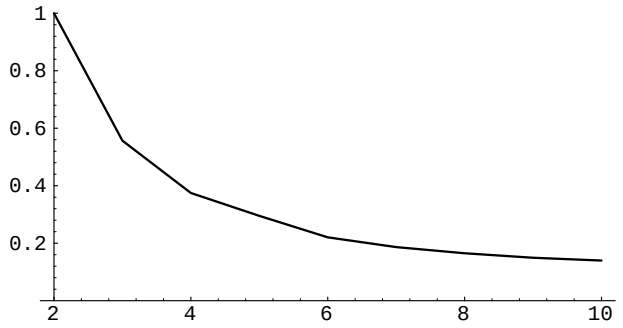

Figure 9: The delay measurement error of the inverter as a function of n.

The error decreases exponentially as n increases so the point of diminishing return is reached beyond $n = 8$. Hence if our algorithm is applied to select a set of good gate sizes, the proliferation of sizes can be potentially avoided.

6 Library Sizing

Given a new chip project targetting a new semi-custom library, the gate selection algorithm provides a guide for choosing the sizes of the primitive cells of the new library. First, the delay equation as a function of gain must be characterized. This is equivalent to generating a continuous cell library parameterized by gain and can be done without choosing the available sizes in the library. Then the macros of the chip project are designed and synthesized using the continuous semi-custom methodology. Since the gates are parameterized by gain during semi-custom synthesis, the absence of a discretely sized library is not a problem. The synthesized circuits are the continuously-sized circuits from which the probability density functions are extracted. The number of available sizes is chosen for each cell type and the "size-match" library can be implemented following the discussion in section 4.

7 Conclusions

We have presented an algorithm to select a good set of gate sizes for primitive cell types with respect to a general measurement function. The algorithm is applied to a realistic semi-custom library. The selection based on closest matching of gate sizes seems to provide the best overall synthesis results. Using our selection algorithm, the number of sizes needed for each cell type can be reduced while maintaining the quality of the circuit implementation. With all the other benefits of a semi-custom library, we hope that this work adds yet another incentive for library designers to choose the semi-custom style for ASICs as well as for high performance libraries.

References

[1] K. Shepard and et al. "Design Methodology for the S/390 Paralell Enterprise Server G4 Microprocessor". *IBM Journal of Research and Development*, 41:515–547, 1997.

[2] K. Keutzer and K. Scott. "Improving Cell Libraries for Synthesis". In *Proc. of the International Workshop on Logic Synthesis*, 1993.

[3] K. Keutzer, K. Kolwicz, and M. Lega. "Impact of Library Size on the Quality of Automated Synthesis". In *Digest of the International Conference on Computer-Aided Design*, pages 120–123, 1987.

[4] R. Haddad, L. van Ginneken, and N. Shenoy. "Drive Selection for Library Design". In *Proc. of the International Workshop on Logic Synthesis*, 1997.

[5] I. Sutherland and R. Sproull. "The Theory of Logical Effort: Desiging for Speed on the Back of an Envelope". In *Advanced Research in VLSI*, University of California at Santa Cruz, 1991.

[6] P. Kudva. "Continuous Optimizations in Synthesis: The Discretization Problem". In *Proc. of the International Workshop on Logic Synthesis*, 1998.

[7] H. Lin and L Linholm. "An Optimized Output Stage for MOS Integrated Circuits". *IEEE Journal of Solid State Circuits*, pages 106–109, 1975.