

# Short Papers

## An Efficient Heterogeneous Tree Multiplexer Synthesis Technique

Hsu-Wei Huang, Cheng-Yeh Wang, and Jing-Yang Jou

**Abstract**—In this paper, a novel strategy for designing the heterogeneous tree multiplexer is proposed. The authors build the multiplexer delay model by curve fitting and then formulate the heterogeneous tree multiplexer design problem as a special type of optimization problem called mixed-integer nonlinear programming (MINLP). A new design parameter, the switch size in each stage, is introduced to improve the speed of the heterogeneous tree multiplexer. The proposed strategy can determine the multiplexer architecture and the switch size in each stage simultaneously. Three optimization methods are provided to synthesize the heterogeneous tree multiplexer according to the design specifications.

**Index Terms**—Capacitance, modeling, optimization, simulation, transistor sizing.

### I. INTRODUCTION

High fan-in multiplexers are used extensively in many applications, including the column decoders of memories [6], [8] and the resistor-chain digital-to-analog converters [7]. Traditionally, high fan-in multiplexers have been designed using the uniform approach [9], in which all switches are interconnected with the output node in common, or the binary tree structure [10], which is a tree-like multi-stage structure. However, a more general architecture called the heterogeneous tree multiplexer was recently proposed [1]. This new architecture has better speed performance than the traditional architectures. Design strategies based on different kinds of switch topology, such as transmission gate and tri-state buffer, are also proposed in [1]–[3].

Alioto and Palumbo [2] presented a strategy for designing a heterogeneous tree multiplexer, which uses minimally sized tri-state buffers as switches. It determines the optimum number of stages and the optimum number of switches per group in each stage such that the multiplexer delay is minimal. In [2], the optimum number of switches per group in each stage depends on the technology adopted. In the example presented in [2], the optimum number of switches per group in each stage is almost four, which is a power of two, as desired. However, changing the technology adopted may cause the number of switches per group in each stage not to be a power of two. Another assumption made by [2] is that the multiplexer delay is roughly independent of the gate sizes. However, the heterogeneous tree multiplexer delay can be improved by suitably selecting the multiplexer architecture, including the number of stages and the number of switches per group in each stage, at the same time as the switch size in each stage. This improvement will be experimentally demonstrated.

This paper introduces a new method that has several important advantages over previous methods. The delay formula of the hetero-

geneous tree multiplexer is built by curve fitting, and then the heterogeneous tree multiplexer design problem is formulated as a special type of optimization problem called mixed-integer nonlinear programming (MINLP). The important advantage of MINLP is that the integer constraints can be set so that the design parameters of the heterogeneous tree multiplexer can be determined properly. Reasonable design parameters can always be obtained directly and no rounding is required. Another feature of the proposed approach is that the proper multiplexer architecture and the switch size in each stage can be determined simultaneously, useful in reducing the multiplexer delay. In [2], only the multiplexer architecture that minimizes the multiplexer delay could be determined. This paper provides three optimization methods—delay minimization, delay minimization under area constraint, and area minimization under delay constraint—to help the designers determine the multiplexer architecture and the switch size in each stage according to the design specifications.

The rest of this paper is organized as follows. Section II defines the problem, Section III describes the design strategies, Section IV presents the experimental results, and Section V draws the conclusions.

### II. PROBLEM FORMULATION

The structure of the heterogeneous tree multiplexer is shown in Fig. 1. The multiplexer has  $N$  inputs and is partitioned into  $k$  stages. In the  $i$ th stage, the switches are combined into groups of  $S_i$  elements with a common output node, which represents an input to the subsequent stage. This paper considers the case in which the switches are implemented using a tri-state buffer (Fig. 2).

The first stage includes  $N/S_1$  groups of tri-state buffers and  $N/S_1$  outputs. The second stage includes  $N/S_1 S_2$  outputs. The  $k$ th stage includes  $N/S_1 S_2 \dots S_k$  outputs. The last stage has a single output that represents the multiplexer output, so the following constraint can be derived

$$\prod_{i=1}^k S_i = N. \quad (1)$$

In this paper, it will be assumed that the delay of the address decoder of the multiplexer can be neglected. The delay of the address decoder can be reduced by careful circuit design, for example, using superbuffer or bipolar complimentary metal–oxide–semiconductor (BiCMOS) drivers [9], [10].

The delay of the heterogeneous tree multiplexer depends strongly on the number of stages and the number of switches per group in each stage, so the problem as defined in previous approaches [1], [2] is to determine the optimal number of stages and the optimal number of switches per group in each stage such that the multiplexer delay is minimal. In [1]–[3], the switch size is kept constant and assumed to be minimal. Here, another parameter—the switch size in each stage—is introduced. This paper will consider the multiplexer architecture, which includes the number of stages and the number of switches per group in each stage, at the same time as the switch size in each stage. The problem will be defined as follows. The problem is to determine the multiplexer architecture and the switch size in each stage such that the multiplexer meets the design specifications. Three optimization methods—delay minimization, delay minimization under area

Manuscript received June 18, 2004; revised September 5, 2004 and November 10, 2004. This work was supported in part by the R.O.C. National Science Council under Grant NSC-91-2218-E-009-013. This paper was recommended by Associate Editor S. Sapatnekar.

The authors are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C. (e-mail: sherwin@eda.ee.nctu.edu.tw; cywang@eda.ee.nctu.edu.tw; jyjou@eda.ee.nctu.edu.tw).

Digital Object Identifier 10.1109/TCAD.2005.852032

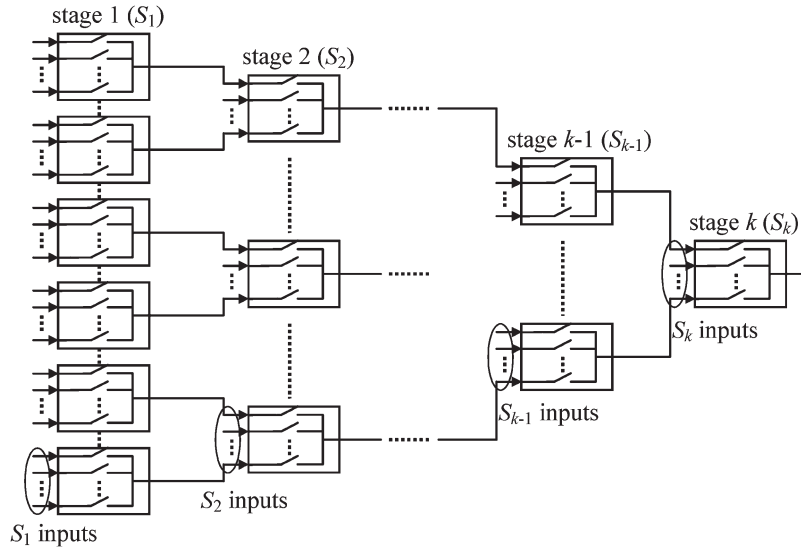


Fig. 1. Heterogeneous tree multiplexer architecture.

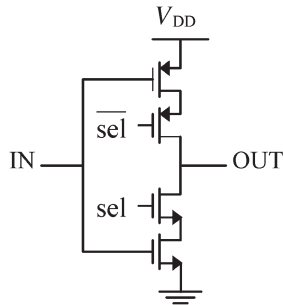


Fig. 2. Tri-state buffer.

constraint, and area minimization under delay constraint—are provided to synthesize automatically the heterogeneous tree multiplexer such that the multiplexer meets the design specifications.

### III. PROPOSED DESIGN STRATEGY

This section introduces the proposed design strategy. The delay model to be used is first considered and then the concept of MINLP is introduced. Next, the path delay model and the load capacitance model in each stage of the heterogeneous tree multiplexer are constructed, and then the delay model of the heterogeneous tree multiplexer is derived. Finally, the objective functions and the constraint functions of the MINLP for the design problem of the heterogeneous tree multiplexer are proposed.

It is easy to show by exhaustive search that there exists a stage number  $k$  such that the heterogeneous tree multiplexer meets the design specifications. Since  $k$  is a small integer, the proposed strategy will find the optimal value of  $k$  by exhaustive search. That is, every possible value of  $k$  will be tried in solving MINLP and then the optimal value of  $k$  will be selected such that the heterogeneous tree multiplexer meets the design specifications.

#### A. Delay Modeling Approach

Most delay modeling approaches, such as those used for standard cell characterization, estimate the delay of a gate for a given input

transition time and output load capacitance, with the sizes of the transistors inside the gate being kept constant. However, in order to improve the speed performance of the heterogeneous tree multiplexer, the authors would like to consider the effects of varying the transistor sizes and find out a strategy to determine appropriate gate size for each stage of the heterogeneous tree multiplexer. The delay model applied in this paper is the convex delay model proposed in [4] and [5]. This model allows us to capture the effects of varying the transistor sizes on the gate delay. The general form of the delay model is given by

$$\text{Delay} = \sum_{j=1}^m P_j \times \prod_{i=1}^n (x_i^{\Delta} + c_{ij})^{\beta_{ij}} + C. \quad (2)$$

Here, the  $x_i$ 's are characterization variables and the  $c_{ij}$ 's,  $\beta_{ij}$ 's,  $C$ , and  $P_j$ 's are referred to collectively as characterization constants.  $m$  is the parameter used to increase the characterization flexibility and  $n$  is the number of characterization variables. The parameter  $\Delta$  is set to either  $-1$  or  $1$ , depending on the variable. For example, the fall delay increases as the output load capacitance  $C_L$ , the positive-channel metal-oxide-semiconductor (PMOS) transistor width  $W_p$ , and the input time constant  $\tau$  are increased, and decreases as the negative-channel metal-oxide-semiconductor (NMOS) transistor width  $W_n$  is increased, implying that the value of  $\Delta$  for the first three variables must be  $1$  and that for  $W_n$  should be  $-1$ . The problem of characterization is to determine appropriate values for the characterization constants. Due to the curve-fitting nature of the characterization procedure, it is not possible to describe direct physical meanings of the characterization constants.

A two-step methodology is used to complete the characterization. In the first step, a number of circuit simulations are performed to collect the experimental data using the HSPICE circuit simulator. In the second step, a least-squares procedure is used to fit the data points to (2). The characterization constants are determined by solving the following nonlinear program that minimizes the sum of the squares of the errors over all data points.

$$\sum_{i=0}^N [D_{\text{estim}}(i) - D_{\text{actual}}(i)]^2 \quad (3)$$

where  $N$  is the number of data points and  $D_{\text{estim}}(i)$  and  $D_{\text{actual}}(i)$  represent the value given by (2) and the corresponding measured value at the  $i$ th data point, respectively.

### B. MINLP

An MINLP is an optimization problem of the form

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && c_{Li} \leq f_i(x) \leq c_{Ui}, \quad i = 1, 2, \dots, m \\ & && b_{Lj} \leq g_j(x) \leq b_{Uj}, \quad j = 1, 2, \dots, p \\ & && x_{Lk} \leq x_k \leq x_{Uk}, \quad k = 1, 2, \dots, n \\ & && x_e \in Z, x_e \subseteq x_k \end{aligned}$$

where  $f_0(x)$  is the objective function,  $f_i(x)$  are the nonlinear constraint functions,  $g_j(x)$  are the linear constraint functions,  $x_k$  are the variables,  $c_{Ui}$ ,  $b_{Uj}$ , and  $x_{Uk}$  are the upper bounds, and  $c_{Li}$ ,  $b_{Lj}$ , and  $x_{Lk}$  are the lower bounds.

### C. Design Strategy I

This section presents design strategy I, which only determines the multiplexer architecture and keeps the switch size minimal.

1) *Delay of Heterogeneous Tree Multiplexer*: The switch delay model associated with design strategy I is defined as follows. As the equation shows, the switch delay depends only on the load capacitance of the switch.

$$\text{Switch Delay} = \sum_{j=1}^m P_j (C + c_j)^{\beta_j} + q \quad (4)$$

where  $C$  is the load capacitance of the switch and  $P_j$ ,  $c_j$ ,  $\beta_j$ , and  $q$  are the characterization constants.

The path delay of the multiplexer is the sum of the switch delay in each stage of the heterogeneous tree multiplexer, so the path delay model of the heterogeneous tree multiplexer can be obtained as

$$\text{Path Delay} = \sum_{i=1}^k \left[ \sum_{j=1}^m P_j (C_i + c_j)^{\beta_j} + q \right] \quad (5)$$

where  $C_i$  is the load capacitance of stage  $i$  and  $P_j$ ,  $c_j$ ,  $\beta_j$ , and  $q$  are the characterization constants.

The switch associated with stage  $i$  drives the output capacitance  $C_{\text{out}}$  of the other  $(S_i - 1)$  OFF switches that belong to the same group, as well as the input capacitance  $C_{\text{in}}$  of the switch in the subsequent stage (see Fig. 3). Hence, the load capacitance of stage  $i$  is defined as

$$C_i = (S_i - 1)C_{\text{out}} + C_{\text{in}} \quad (6)$$

where  $C_{\text{out}}$  is the output capacitance of the switch and  $C_{\text{in}}$  is the input capacitance of the switch.

2) *Design Optimization*: This section presents the objective functions and constraint functions of the MINLP for three different optimization methods.

Substituting (6) into (5) yields the multiplexer delay equation

$$\sum_{i=1}^k \left[ \sum_{j=1}^m P_j \left( ((S_i - 1)C_{\text{out}} + C_{\text{in}}) + c_j \right)^{\beta_j} + q \right]. \quad (7)$$

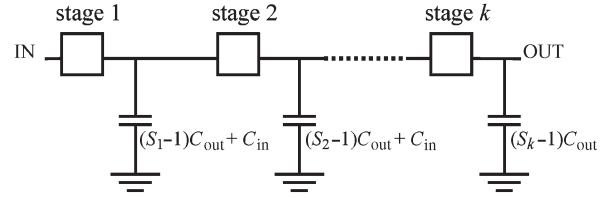


Fig. 3. Critical path model associated with design strategy I.

The area of the multiplexer is defined as the number of switches in the multiplexer

$$N + \left( \frac{N}{S_1} \right) + \left( \frac{N}{S_1 S_2} \right) + \dots + \left( \frac{N}{S_1 S_2 \dots S_{k-1}} \right). \quad (8)$$

The basic constraints are as

$$S_i \geq 2 \text{ and } S_i \in Z, \quad i = 1, 2, \dots, k. \quad (9)$$

- **Optimization Method I**: The multiplexer architecture whose delay is minimal can be obtained using this optimization method. The objective function of MINLP is defined by (7), and the constraint functions of MINLP are defined by (1) and (9).
- **Optimization Method II**: The multiplexer architecture whose delay is minimal under area constraint can be obtained using this optimization method. The objective function of MINLP is defined by (7), and the constraint functions of MINLP are defined by (1), (8), and (9).
- **Optimization Method III**: The multiplexer architecture whose area is minimal under delay constraint can be obtained using this optimization method. The objective function of MINLP is defined by (8), and the constraint functions of MINLP are defined by (1), (7), and (9).

### D. Design Strategy II

This section presents the design strategy II, by which the multiplexer architecture and the switch size in each stage of the multiplexer are determined simultaneously. The switch size can be different in each stage of the heterogeneous tree multiplexer.

1) *Delay of Heterogeneous Tree Multiplexer*: The switch delay model associated with design strategy II is defined as follows. The switch delay depends on both the load capacitance and the switch size as

$$\text{Switch Delay} = \sum_{j=1}^m P_j (W^{-1} + c_{1j})^{\beta_{1j}} (C + c_{2j})^{\beta_{2j}} + q \quad (10)$$

where  $W$  is the NMOS transistor width of the switch,  $C$  is the load capacitance of the switch, and  $P_j$ ,  $c_{1j}$ ,  $\beta_{1j}$ ,  $c_{2j}$ ,  $\beta_{2j}$ , and  $q$  are the characterization constants.

The path delay model of the heterogeneous tree multiplexer can be obtained as

$$\text{Path Delay} = \sum_{i=1}^k \left[ \sum_{j=1}^m P_j (W_i^{-1} + c_{1j})^{\beta_{1j}} (C_i + c_{2j})^{\beta_{2j}} + q \right] \quad (11)$$

where  $W_i$  is the NMOS transistor width of the switch in stage  $i$ ,  $C_i$  is the load capacitance of stage  $i$ , and  $P_j$ ,  $c_{1j}$ ,  $\beta_{1j}$ ,  $c_{2j}$ ,  $\beta_{2j}$ , and  $q$  are the characterization constants.

The load capacitance of stage  $i$  is defined as

$$C_i = (S_i - 1)C_{\text{out}_i} + C_{\text{in}_{i+1}} \quad (12)$$

where  $C_{\text{out}_i}$  is the output capacitance of the switch in stage  $i$  and  $C_{\text{in}_{i+1}}$  is the input capacitance of the switch in stage  $i + 1$ .

The gate capacitance and the source/drain capacitance of the metal-oxide-semiconductor field-effect transistor (MOSFET) are proportional to the channel width of the MOSFET, so the input/output capacitance of the tri-state buffer is defined as

$$C_{\text{out}_i} = aW_i + b \quad (13)$$

$$C_{\text{in}_{i+1}} = cW_{i+1} + d \quad (14)$$

where  $a$ ,  $b$ ,  $c$ , and  $d$  are the characterization constants and  $W_i$  and  $W_{i+1}$  are the NMOS transistor width of switches in stages  $i$  and  $i + 1$ , respectively.

2) *Design Optimization*: This section presents the objective functions and constraint functions of the MINLP associated with three different optimization methods.

Substituting (12) into (11) yields the multiplexer delay equation

$$\sum_{i=1}^k \left[ \sum_{j=1}^m P_j (W_i^{-1} + c_{1j})^{\beta_{1j}} ((S_i - 1)(aW_i + b) + (cW_{i+1} + d)) + c_{2j}^{\beta_{2j}} + q \right]. \quad (15)$$

The area of the multiplexer is defined as the sum of the NMOS transistor widths of the switches in the multiplexer given as

$$NW_1 + \left(\frac{N}{S_1}\right)W_2 + \left(\frac{N}{S_1S_2}\right)W_3 + \cdots + \left(\frac{N}{S_1S_2 \cdots S_{k-1}}\right)W_k. \quad (16)$$

In the experiment, the range of NMOS transistor widths is  $0.3\text{--}3 \mu\text{m}$ . In order to obtain reasonable design parameters,  $10W_i$  must be a positive integer. So the constraints shown in (17) at the bottom of the page were obtained.

- Optimization Method I: This optimization method can determine the multiplexer architecture and the switch size in each stage such that the multiplexer delay is minimal. The objective function of MINLP is defined by (15), and the constraint functions of MINLP are defined by (1) and (17).
- Optimization Method II: This optimization method can determine the multiplexer architecture and the switch size in each stage such that the multiplexer delay is minimal under area constraint. The objective function of MINLP is defined by (15), and the constraint functions of MINLP are defined by (1), (16), and (17).
- Optimization Method III: This optimization method can determine the multiplexer architecture and the switch size in each stage such that the multiplexer area is minimal under delay constraint. The objective function of MINLP is defined by (16), and the constraint functions of MINLP are defined by (1), (15), and (17).

TABLE I  
DELAY MODEL PARAMETERS ASSOCIATED WITH DESIGN STRATEGY I

$P_1$	7.041685
$c_1$	0.009220
$\beta_1$	0.994263
$q$	0

TABLE II  
SWITCH INPUT/OUTPUT CAPACITANCE ASSOCIATED WITH DESIGN STRATEGY I

$C_{\text{out}}$	0.0021 pF
$C_{\text{in}}$	0.0024 pF

#### IV. EXPERIMENTAL RESULTS

This section presents the experimental results obtained using the proposed approach. A 256-input multiplexer is designed and simulated using the  $0.18\text{-}\mu\text{m}$  CMOS process. In this experiment, the parameter  $m$  is set to 1. The output loading of the 256-input multiplexer is  $0.003 \text{ pF}$ . The input signal for switch characterization is a rising signal, which rises from 0 to 1.8 V during 0.1 ns. The input signals for the simulation of the multiplexer path delay are a rising signal, which rises from 0 to 1.8 V during 0.1 ns, and a falling signal, which falls from 1.8 to 0 V during 0.1 ns. The PMOS/NMOS ratio is 3.57:1, which is set to equalize the rising and falling times. The experiment has two parts. In part A, the multiplexer is designed according to strategy I. In part B, the multiplexer is designed according to strategy II. In each part, three different optimization methods are applied. The first is delay minimization, which is the same as the previous approach [2]. The second is delay minimization under area constraint, and the third is area minimization under delay constraint.

An Intel/Pentium-IV 2.4-GHz personal computer (PC) with 512 MB of random access memory (RAM) was used to perform the experiment. The curve-fitting problem was solved using MINOS [11]–[13]. The MINLP problem was solved using BARON [14], [16], [17], which is a global solver and can guarantee that a global optimal solution to the problem is found.

##### A. Experimental Results Obtained Using Design Strategy I

Table I shows the switch delay model parameters. Table II shows the input and output capacitances of the switch. Table III shows the experimental results of the three optimization methods. Fig. 4 shows the delay/area tradeoff curve. Table IV shows the parameters associated with Fig. 4.

A two-step methodology is used to complete the switch characterization. In the first step, a number of circuit simulations are performed using HSPICE to collect 37 data points. The NMOS width is fixed at  $0.3 \mu\text{m}$ . The load capacitance ranges from  $0.004$  to  $0.04 \text{ pF}$  in steps of  $0.001 \text{ pF}$ . In the second step, a least-squares procedure is used to fit the data points to (4).

As the experimental results in Table III show, the predicted delay agrees well with the simulated result. The maximum error between the estimated result and the HSPICE simulation result is less than 5%. Several similar optimal results using optimization method I were obtained. They are  $k = 3$ ,  $(S_1, S_2, S_3) = (4, 8, 8)$ ,  $(8, 8, 4)$ ,  $(8, 4, 8)$

$$3 \geq W_i \geq 0.3 \quad \text{and} \quad 10W_i \in \mathbb{Z}, \quad S_i \geq 2 \quad \text{and} \quad S_i \in \mathbb{Z}, \quad i = 1, 2, \dots, k \quad (17)$$

TABLE III  
EXPERIMENTAL RESULTS OF DESIGN STRATEGY I

	Multiplexer Architecture	Constraint	Area (switch number)	Delay (estimation) (nanosecond)	Rising Delay (HSPICE simulation) (nanosecond)	Error (percent)	Falling Delay (HSPICE simulation) (nanosecond)	Error (percent)
Optimization Method I	$S_i = (4, 4, 4, 4)$ $W_i = (0.3, 0.3, 0.3, 0.3)$	None	340	0.5208	0.5278	1.3	0.5338	2.4
Optimization Method II	$S_i = (16, 4, 4)$ $W_i = (0.3, 0.3, 0.3)$	290 switches	276	0.5717	0.6005	4.8	0.5797	1.4
Optimization Method III	$S_i = (16, 16)$ $W_i = (0.3, 0.3)$	0.65 ns	272	0.6226	0.6290	1	0.6326	1.6

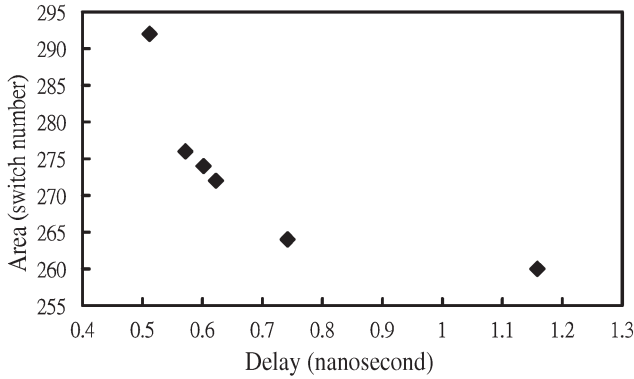


Fig. 4. Area versus delay curve obtained using design strategy I.

TABLE IV  
PARAMETERS ASSOCIATED WITH FIG. 4

Delay (nanosecond)	Area (switch number)	$k$	$S_i$
0.5119	292	3	(8, 8, 4)
0.5717	276	3	(16, 4, 4)
0.6017	274	3	(16, 8, 2)
0.6226	272	2	(16, 16)
0.7418	264	2	(32, 8)
1.1582	260	2	(64, 4)

and  $k = 4$ ,  $(S_1, S_2, S_3, S_4) = (4, 4, 4, 4)$ . The estimated delay for the first three results is 0.5119 ns, and the estimated delay for the last result is 0.5208 ns. The HSPICE simulation results of the four optimal solutions are also similar. So  $k = 4$  and  $S_1 = S_2 = S_3 = S_4 = 4$  were selected, which are the same as the result of the previous approach [2], [3], as the solution of optimization method I. Using optimization method II, the multiplexer architecture that yields minimal delay under the area constraint of 290 switches is  $k = 3$ ,  $S_1 = 16$  and  $S_2 = S_3 = 4$ . According to optimization method III, the multiplexer architecture that yields minimal area under the delay constraint of 0.65 ns is  $k = 2$ ,  $S_1 = S_2 = 16$ . The experimental results reveal that the proposed approach can effectively determine the multiplexer architecture according to the design specifications.

### B. Experimental Results Obtained Using Design Strategy II

Tables V and VI show the switch delay model parameters and the switch input/output capacitance model parameters, obtained by curve

TABLE V  
DELAY MODEL PARAMETERS ASSOCIATED WITH DESIGN STRATEGY II

$P_1$	2.322326
$c_{11}$	-0.021905
$\beta_{11}$	0.908354
$c_{21}$	0.000001
$\beta_{21}$	0.989680
$q$	0.067169

TABLE VI  
SWITCH INPUT/OUTPUT CAPACITANCE MODEL PARAMETERS ASSOCIATED WITH DESIGN STRATEGY II

$C_{out}$ (picofarad)	$a$	0.005612
	$b$	0.000320
$C_{in}$ (picofarad)	$c$	0.007279
	$d$	0.000120

fitting, respectively. Table VII shows the experimental results of the three optimization methods. Fig. 5 shows the delay/area tradeoff curve. Table VIII shows the parameters associated with Fig. 5.

A two-step methodology is used to complete the switch characterization. In the first step, a number of circuit simulations are performed using HSPICE to collect 1260 data points. The size of NMOS ranges from 0.3 to 10.3  $\mu\text{m}$  in steps of 0.5  $\mu\text{m}$ . The load capacitance ranges from 0.004 to 0.594 pF in steps of 0.01 pF. In the second step, a least-squares procedure is used to fit the data points to (10).

The same two-step methodology is used to complete the characterization of switch input/output capacitances. In the first step, ten data points for each switch input and output capacitances are collected separately using HSPICE. The size of NMOS ranges from 0.3 to 3  $\mu\text{m}$  in steps of 0.3  $\mu\text{m}$ . In the second step, a least-squares procedure is used to fit the data points to (13) and (14) and then the parameters for switch output and input capacitances can be obtained.

Alioto and Palumbo [2] and Alioto *et al.* [3] use only minimally sized switches and do not consider the transistor sizing problem. In this experiment, the authors try to show that the delay of the heterogeneous tree multiplexer can be improved by considering the multiplexer architecture and the switch size in each stage simultaneously. As the experimental result shown in the first row of Table III, the optimal multiplexer delays for rising input and falling input are 0.5278 and 0.5338 ns, which are obtained by considering the multiplexer architecture only and keeping the switch size in each stage minimal. As the experimental result shown in the first row of Table VII, the optimal multiplexer delays for rising input and falling input are 0.4595 and 0.4529 ns, which are obtained by considering the multiplexer

TABLE VII  
EXPERIMENTAL RESULTS OF DESIGN STRATEGY II

	Multiplexer Architecture	Constraint	Area (micrometer)	Delay (estimation) (nanosecond)	Rising Delay (HSPICE simulation) (nanosecond)	Error (percent)	Falling Delay (HSPICE simulation) (nanosecond)	Error (percent)
Optimization Method I	$S_i = (4, 8, 8)$ $W_i = (3, 1.3, 0.7)$	None	856.8	0.4669	0.4595	1.6	0.4529	3
Optimization Method II	$S_i = (8, 8, 4)$ $W_i = (1.4, 0.8, 0.6)$	400 $\mu\text{m}$	386.4	0.4737	0.4834	2	0.4617	2.6
Optimization Method III	$S_i = (16, 16)$ $W_i = (0.3, 0.3)$	0.65 ns	81.6	0.6013	0.6290	4.4	0.6326	4.9
400-input Multiplexer	$S_i = (5, 8, 10)$ $W_i = (3, 1.3, 0.7)$	None	1311	0.5085	0.5024	1.2	0.4924	3.2

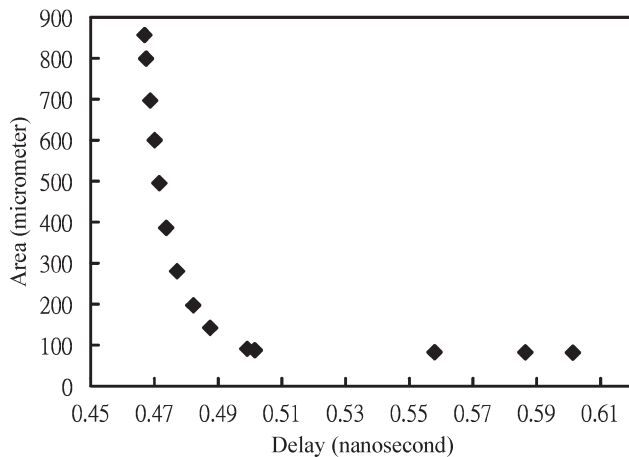


Fig. 5. Area versus delay curve obtained using design strategy II.

architecture and the switch size in each stage simultaneously. The minimal multiplexer delays for rising input and falling input can be reduced by almost 15% and 18%, respectively, if the multiplexer architecture and the switch size in each stage are suitably selected using the optimization method I of strategy II.

As the experimental results in Table VII show, the predicted delay agrees well with the simulated result. The maximum error between the estimated result and the HSPICE simulation result is less than 5%. In optimization method I, as desired, the minimal multiplexer delay can be obtained by determining the multiplexer architecture and the switch size in each stage simultaneously. Using optimization methods II and III, the multiplexer architecture and the switch size in each stage, meeting the design constraints, can be obtained.

Table IX shows the detail run time of the experiment. “(number)” is used to represent the run time for finding an infeasible solution, meaning that no solution that satisfies the constraints exists. The time required for solving the MINLP problem depends on the MINLP solver. Several MINLP solvers were tried, including minlpBB, glc-Solve, OQNL, SBB [15], and BARON [16], [17]. After comparing their performance and run time, BARON [16], [17] is used as the solver of the proposed MINLP problem. SBB [15] is also a good choice for solving the MINLP problem although it does not guarantee that a global optimal solution is found. In the experiment, SBB [15] always yields the same result as BARON [16], [17]. They both provide very good optimization results in a reasonable time. The optimization result using design strategy I can be obtained in a short time. The run

TABLE VIII  
PARAMETERS ASSOCIATED WITH FIG. 5

Delay (nanosecond)	Area (micrometer)	$k$	$S_i$	$W_i$
0.4669	856.8	3	(4, 8, 8)	(3, 1.3, 0.7)
0.4674	799.2	3	(4, 8, 8)	(2.8, 1.2, 0.7)
0.4687	696.8	3	(4, 8, 8)	(2.4, 1.2, 0.7)
0.4701	600	3	(4, 8, 8)	(2.1, 0.9, 0.6)
0.4715	495.2	3	(8, 8, 4)	(1.8, 1, 0.6)
0.4737	386.4	3	(8, 8, 4)	(1.4, 0.8, 0.6)
0.4771	280.4	3	(8, 8, 4)	(1, 0.7, 0.5)
0.4822	197.2	3	(8, 8, 4)	(0.7, 0.5, 0.5)
0.4875	142.4	3	(8, 8, 4)	(0.5, 0.5, 0.4)
0.4991	91.2	3	(8, 8, 4)	(0.3, 0.4, 0.4)
0.5015	87.6	3	(8, 8, 4)	(0.3, 0.3, 0.3)
0.5579	82.8	3	(16, 4, 4)	(0.3, 0.3, 0.3)
0.5864	82.2	3	(16, 8, 2)	(0.3, 0.3, 0.3)
0.6013	81.6	2	(16, 16)	(0.3, 0.3)

time of design strategy II is larger than design strategy I, but it is also reasonable. As the experimental results for design strategies I and II reveal, the optimal  $k$  value is between 2 and 4. The run time for these solutions in Table IX is less than 1 min.

### C. Discussion

In this experiment, the PMOS/NMOS ratio is set to balance the rising delay and the falling delay, and a single delay model is applied to predict the rising and falling delays of the multiplexer obtained by HSPICE simulation. In order to extend the proposed approach to handle various PMOS/NMOS ratios, the rising and falling delays need to be modeled separately. The authors propose a method to model the rising and falling delays separately as follows. The switch is separately characterized for rising and falling inputs, and the delay model parameters for rising and falling inputs are separately obtained. To obtain the rising delay model of the multiplexer, the switch rising delay model is used in the odd stages of the multiplexer and the switch falling delay model is used in the even stages of the multiplexer.



TABLE IX  
RUN TIME OF DESIGN STRATEGIES I AND II

Run Time (second)	Design Strategy I			Design Strategy II		
	Method I	Method II	Method III	Method I	Method II	Method III
$k = 1$	0.03	0.02	(0.02)	0.06	0.06	(0.02)
$k = 2$	0.03	0.03	0.05	0.38	0.28	0.12
$k = 3$	0.19	0.12	0.11	22.2	10.89	0.25
$k = 4$	0.19	0.12	0.11	33.5	51.66	0.19
$k = 5$	0.38	0.02	0.11	501.89	1146.55	0.45
$k = 6$	0.33	(0.02)	(0.27)	1378.23	3278.05	249.38
$k = 7$	0.19	(0.02)	(0.02)	2461.44	2093.19	(18.56)
$k = 8$	0.02	(0.02)	(0.02)	1025.98	160.89	(3.23)
Total Run Time	1.36	0.37	0.71	5423.7	6741.6	272.2

To obtain the falling delay model of the multiplexer, the switch falling delay model is used in the odd stages of the multiplexer and the switch rising delay model is used in the even stages of the multiplexer. A parameter  $\max\_delay$  is defined to handle separately the rising and falling delays. Two additional constraint functions are defined as

$$\max\_delay > \text{the rising delay model of the multiplexer}$$

$$\max\_delay > \text{the falling delay model of the multiplexer.}$$

For optimization methods I and II, the parameter  $\max\_delay$  is used as the objective function instead of the original single delay model. For optimization method III, the parameter  $\max\_delay$  is unnecessary and the delay constraints are set separately for the rising and falling delay models of the multiplexer.

In this experiment, input drivers connected to the first stage of the multiplexer are not considered. When input drivers are connected to the first stage of the multiplexer, the delay due to the input drivers and the input capacitance of the multiplexer can be determined by the following method. First, the delay model of the input driver must be determined by curve fitting or from the cell library databook. Then, the input capacitance of the first stage of the multiplexer, which is the load capacitance of the input driver, is determined by the method proposed in this paper. The delay of the input driver is determined using the delay model of the input driver and the input capacitance of the first stage of the multiplexer, which is the load capacitance of the input driver. The total path delay is the sum of the delay of the input driver and the delay of the multiplexer. The authors use an example to explain it. When minimized-size tri-state buffers are used as input drivers and are connected to the first stage of the multiplexer shown in the first row of Table VII, the total path delay can be determined as follows. The delay model parameters of the minimized-size tri-state buffer are shown in Table I. The load capacitance of the input driver is 0.02196 pF, which is determined using (14). The estimated delay of the input driver is 0.224 ns, which is obtained using (4). The estimated delay of the multiplexer is 0.4669 ns, which is shown in the first row of Table VII. The total estimated delay is 0.6909 ns. The HSPICE simulated result is 0.7119 ns. The error is 3%. The method can also be applied to other types of input driver, for example, inverters, by applying the associated delay model of the input drivers.

In Section IV-B, the simultaneous approach, which simultaneously optimizes multiplexer architecture and switch size, is applied to optimize the multiplexer. Another alternative is the sequential approach.

Initially, design strategy I is used to determine the multiplexer architecture,  $k$  and  $S_i$ . Then,  $k$  and  $S_i$  are substituted into the MINLP model of design strategy II and the MINLP problem is solved to determine the transistor size in each stage of a multiplexer. A simple experiment is performed to show the run time and performance of the sequential approach. The authors apply the transistor sizing to the multiplexer shown in the first row of Table III. The run time for transistor sizing is 2.47 s. The optimal size  $W_i = (3, 1.5, 0.9, 0.6)$ . The rising delay and falling delay obtained by HSPICE simulation are 0.4694 and 0.4534 ns, respectively. The experimental results reveal that the result obtained using the sequential approach is slightly worse than the result obtained using the simultaneous approach, which is shown in the first row of Table VII, but the run time of the sequential approach is less than the run time of the simultaneous approach.

The proposed method has some advantages over the earlier method [2], [3]. The first is that it offers more flexibility in that it can be applied to design various types of multiplexer. The earlier method [2], [3] can be applied to some special multiplexers, whose inputs are powers of 4 or 2. The fourth row in Table VII presents the results of optimizing a 400-input multiplexer using design strategy II. The second advantage of the method is that it can be applied to various processes because the switch delay model is characterized using HSPICE simulation data. The third advantage of the method is that it can be used to size any specified architecture of a multiplexer. For example, if a 400-input multiplexer with an architecture of  $k = 4$  and  $S_i = (4, 4, 5, 5)$  needs to be sized,  $k$  and  $S_i$  are substituted into the MINLP model of design strategy II and then the MINLP problem is solved. The optimal switch size in each stage can be easily obtained. The optimal size  $W_i = (3, 1.5, 0.9, 0.6)$  and the estimated delay of the multiplexer is 0.5088 ns.

## V. CONCLUSION

In this paper, a novel design strategy for designing a high fan-in heterogeneous tree multiplexer is proposed. Curve fitting is used to build the multiplexer delay model, and then the heterogeneous tree multiplexer design problem is formulated as a special type of optimization problem called mixed-integer nonlinear programming (MINLP). The proposed design strategy can help designers determine the multiplexer architecture and the switch size in each stage according to the design requirement, such as delay minimization, delay minimization under area constraint, or area minimization under delay constraint. MINLP enables the optimal choice of design parameters to

be obtained directly. The proposed approach is applicable to different process technologies. Another important advantage is that the multiplexer delay can be improved by suitably selecting the multiplexer architecture and the switch size in each stage simultaneously.

#### ACKNOWLEDGMENT

The authors would like to thank Prof. T. Westerlund of Abo Akademi University for his constructive suggestions about solving the MINLP problems. The anonymous referees are also appreciated for their valuable comments.

#### REFERENCES

- [1] M.-B. Lin, "On the design of fast large fan-in CMOS multiplexers," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 8, pp. 963–967, Aug. 2000.
- [2] M. Alioto and G. Palumbo, "Optimized design of high fan-in multiplexers using switches with driving capability," in *Proc. 8th IEEE Int. Conf. Electronics, Circuits and Systems (ICECS)*, St. Julians, Malta, 2001, pp. 737–740.
- [3] M. Alioto, G. Di Cataldo, and G. Palumbo, "Optimized design of high fan-in multiplexers using tri-state buffers," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 10, pp. 1500–1505, Oct. 2002.
- [4] M. Ketkar, K. Kasamsetty, and S. Sapatnekar, "Convex delay models for transistor sizing," in *Proc. 37th Design Automation Conf. (DAC)*, Los Angeles, CA, 2000, pp. 655–660.
- [5] K. Kasamsetty, M. Ketkar, and S. Sapatnekar, "A new class of convex functions for delay modeling and its application to the transistor sizing problem," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, pp. 779–788, Jul. 2000.
- [6] R. T. Howe and C. G. Sodini, *Microelectronics: An Integrated Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1997.
- [7] D. A. Johns and K. Martin, *Analog Integrated Circuit Design*. New York: Wiley, 1997.
- [8] J. Rabaey, *Digital Integrated Circuits (A Design Perspective)*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [9] S. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*. New York: McGraw-Hill, 1996.
- [10] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A System Perspective*, 2nd ed. Reading, MA: Addison-Wesley, 1993.
- [11] K. Holmstrom and A. Goran, *User's Guide for Tomlab v4.0.5*. San Diego, CA: Tomlab Optimization Inc., Feb. 28, 2003.
- [12] K. Holmstrom, A. Goran, and M. Edvall, *TOMLAB/MINOS User's Guide*. San Diego, CA: Tomlab Optimization Inc., May 4, 2004.
- [13] B. A. Murtagh and M. A. Saunders, *MINOS 5.5 User's Guide*, Dept. Operations Res., Stanford Univ., Stanford, CA, Rep. SOL 83-20R (Revised Jul. 1998).
- [14] A. Brooke, D. Kendrick, A. Meeraus, R. Raman, and R. E. Rosenthal, *GAMS: A User's Guide*. Washington, DC: GAMS Develop. Corp., 1998.
- [15] *GAMS/SBB Solver Manual*, GAMS Develop. Corp., Washington, DC, Apr. 30, 2002.
- [16] *GAMS/BARON Solver Manual*, GAMS Develop. Corp., Washington, DC, Nov. 25, 2003.
- [17] N. V. Sahinidis, *BARON: Branch and Reduce Optimization Navigator, User's Manual*. Urbana-Champaign, IL: Dept. Chem. Eng., Univ. Illinois Urbana-Champaign, 2000.

## Evaluating the Reliability of NAND Multiplexing With PRISM

Gethin Norman, David Parker, Marta Kwiatkowska,  
and Sandeep Shukla

**Abstract**—Probabilistic-model checking is a formal verification technique for analyzing the reliability and performance of systems exhibiting stochastic behavior. In this paper, we demonstrate the applicability of this approach and, in particular, the probabilistic-model-checking tool PRISM to the evaluation of reliability and redundancy of defect-tolerant systems in the field of computer-aided design. We illustrate the technique with an example due to von Neumann, namely NAND multiplexing. We show how, having constructed a model of a defect-tolerant system incorporating probabilistic assumptions about its defects, it is straightforward to compute a range of reliability measures and investigate how they are affected by slight variations in the behavior of the system. This allows a designer to evaluate, for example, the tradeoff between redundancy and reliability in the design. We also highlight errors in analytically computed reliability bounds, recently published for the same case study.

**Index Terms**—Defect-tolerant architectures, multiplexing, probabilistic-model checking, reliability.

### I. INTRODUCTION

Probabilistic-model checking is a formal verification technique, which has already been successfully used to analyze the performance and reliability of a wide range of real-life systems, including dynamic power-management schemes [1], embedded systems [2], computer networks, queueing systems, and manufacturing processes. It has also been used to study "quality of service" properties of real-time probabilistic communication protocols, such as IEEE 1394 FireWire [3], IEEE 802.3 CSMA/CD [4], Zeroconf [5], IEEE 802.11 wireless local area networks [6], and Bluetooth [7], and to verify both probabilistic security protocols (e.g., [8]) and randomized distributed algorithms (e.g., [9]).

In this paper we present results that demonstrate the advantages of using probabilistic-model checking and, in particular, the probabilistic-model-checking tool PRISM [10], to model and analyze defect-tolerant systems. We have chosen to investigate the reliability of NAND multiplexing [11], but this approach can be applied to other defect-tolerant systems such as  $R$ -fold Modular Redundancy [11] and Reconfiguration [12], [13]. This work differs from the standard approaches in the literature to analyzing multiplexing in that we evaluate the reliability of specific cases as opposed to considering the general framework, and hence are not necessarily restricted by the analytical bounds of reliability of, for example, von Neumann [11] and Pippenger [14].

Our results demonstrate that, by applying probabilistic-model checking, it is straightforward to investigate the effect on reliability of slight variations in the behavior of the system's components, for example the change in reliability as the probability of gate failure

Manuscript received February 2, 2004; revised July 15, 2004. This work was supported by NSF grants CCR-0098335 and CCR-0340740, EPSRC grants GR/N22960 and GR/S11107, FORWARD, SRC, and DARPA/ITO supported PADS project under the PAC/C program. This paper was recommended by Associate Editor J. H. Kukula.

G. Norman, D. Parker, and M. Kwiatkowska are with the School of Computer Science, University of Birmingham, Birmingham, B15 2TT, U.K.

S. Shukla is with the Bradley School of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060 USA.

Digital Object Identifier 10.1109/TCAD.2005.852033