Generalized Delay Optimization of Resistive Interconnections Through an Extension of Logical Effort

Kumar Venkat
Silicon Graphics, Inc.
2011 N. Shoreline Blvd.
Mountain View, CA 94039

***Abstract*: Resistance of VLSI interconnections has become significant due to large die sizes and sub-micron geometries in high performance designs. Previous studies have proposed optimal repeater schemes using simple buffers for delay optimization of the interconnection. This paper proposes a more general approach that handles arbitrary logic gates as well as buffers. The methodology is based on an extension of the concept of logical effort. The optimization yields proper spacing of the given logic gates, additional repeaters (buffers) required for a given RC line, and sizing of all the gates. This approach is applicable to many design situations where existing logic gates must be considered in the overall repeater scheme.**

I. Introduction

Resistance of interconnections is becoming increasingly important in state-of-the-art VLSI design, due to large die sizes, long interconnections and sub-micron geometries. It is no longer sufficient to consider only the capacitance of interconnections in optimizing delay and cycle time. The intrinsic RC delay of long metal interconnections is becoming a significant component of the overall circuit delay. Further, interconnect resistance degrades the current drive capability of buffers designed to drive heavy capacitive loads.

There are a number of commonly occurring VLSI circuit structures that must compute complex logic functions while gathering information via long interconnections. An example is a network that computes a single-bit parity on a 64-bit input to the chip. The inputs enter the chip through pads placed around the periphery of the chip, and the computation produces a single-bit result at one common point on the chip. There is complexity in both computation and communication here. Good delay optimization strategies are required that will consider both types of complexity in a one framework.

The motivation for this work is to develop a general methodology for overall delay optimization of a circuit path that consists of various types of CMOS logic gates and long segments of interconnection. The methodology is intended to be easily applicable to real VLSI design scenarios, providing designers with a quick and accurate way of speeding up interconnect-limited paths.

Delay optimization in the presence of resistive interconnections (long uniform lines) has been the subject of considerable research [1,2,3,4]. These studies have proposed optimal solutions, assuming a repeater structure along the RC line, for selecting the number and type of buffers, position of buffers along the line, and sizing of buffers. However, these solutions are limited to simple buffers or inverters, and do not apply to general CMOS logic gates. The main goal of this work is to overcome this limitation and propose a general solution.

The vehicle chosen for development of the solution and the methodology is the concept of *logical effort* [5]. Logical effort is a powerful framework for performance optimization of CMOS circuits. The effects of capacitive load, complexity of the logic function and the number of stages are combined into a single framework for performance optimization. However, the existing framework of logical effort does not consider the effect of any series resistance at the output of a logic gate, such as a resistive interconnection line or a pass transistor.

In this paper, the framework of logical effort is first extended to include series resistance at gate outputs, in addition to capacitive load. This effect is termed *resistive effort*. It is shown that the resistive effort adds to the native logical effort of a gate, thus weakening the drive capability of the gate as it should.

Based on this extension, delay optimization through spacing of logic gates along a long uniform RC line is considered. It is shown that the optimal spacing is, in general, not equal spacing. The equal spacing solution for inverters [1,2,3,4] is just a special case of this general result. Based on this general solution, it is shown that transistor sizing can be used to allow equal spacing of arbitrary gates, which is helpful for practical chip layouts. Further, using equal spacing of appropriately sized logic gates, good approximations are presented for the optimum number of stages as well as the optimum transistor sizes for all the gates.

II. Extension of Logical Effort

The concept of resistive effort will be developed in this section as an extension of the theory of logical effort [5].

0-7803-1254-6/93$03.00 © 1993 IEEE

Delay through a simple logic gate, driving a capacitive load, is modeled in a way that clearly separates four distinct contributions to the delay:

$$d = \tau(gh + p) \qquad (1)$$

g is the logical effort of the gate, h is the electrical effort, and p is the parasitic or intrinsic delay of the gate. τ is a technology constant defined as the delay of an ideal inverter with no intrinsic delay, driving another identical inverter.

The logical effort g represents the computational complexity of the gate and measures how much weaker it is in current drive compared to an inverter with the same input capacitance. It compares the characteristic time constant (product of output resistance and input capacitance) of a gate with that of an inverter. Logical effort is a function of the topology of the transistor interconnections, but not of transistor sizes.

$$g = (R_tC_t)/(R_{tinv}C_{tinv}) \qquad (2)$$

Subscript t represents a minimum-sized template of the logic gate and subscript $tinv$ represents a minimum-sized inverter.

The electrical effort h is the ratio of the load capacitance to input capacitance, and clearly depends on transistor sizes.

$$h = C_{load}/C_i \qquad (3)$$

The parasitic delay p occurs primarily due to source/drain diffusion capacitance C_{pt} at the output of the gate. It depends on the layout geometry, but is independent of transistor sizes.

$$p = (R_tC_{pt})/(R_{tinv}C_{tinv}) \qquad (4)$$

The method of logical effort works with first-order delay equations, and applies to any static CMOS gate whose pullup and pulldown networks qualify as valid *transistor groups* [7] or *transistor stages* [8]. Further, it assumes that rise and fall times are made equal through transistor sizing of pullup and pulldown networks in a gate.

Let us now consider the case of a gate driving a uniform line of total resistance R_w and total capacitance C_w. Fig. 1 illustrates this using a lumped Π model of the interconnection. The subscript i indicates a logic gate that is a scaled version of its template. Using first-order delay equations, the total delay can be expressed as

$$d = \tau((g + r)h + p) \qquad (5)$$

The *resistive effort* r that adds to the logical effort of the gate to make it a weaker driver, is defined as

$$r = M\,(R_wC_i)/(R_{tinv}C_{tinv}),\ \text{where}\ M = \frac{1}{2}. \qquad (6)$$

C_i is the input capacitance of the logic gate based on actual transistor sizes. The M factor comes from the fact that only half of the wire capacitance is charged through the wire resistance in the lumped Π model. The resistive effort depends on the fraction of the load capacitance driven through series resistance at the gate output. Equation (6) also shows that a gate with higher drive (higher C_i) is weakened more by a given series resistance than a low-drive gate in relative terms.

Equation (6) can be easily modified for any other lumped wire model, as well as for RC tree networks driven by a gate [6,7]. The M factor would simply be the ratio of capacitance driven by external series resistance on a given path to the total external load capacitance (including capacitance on branches) at the gate output.

III. Optimal spacing of gates

An optimal solution for spacing of arbitrary logic gates along a uniform RC line will be developed in this section.

Let us start with the problem of placing two logic gates optimally along a fixed length of interconnection that must be driven by the gates. The two gates have logical efforts g_1 and g_2, and input capacitances C_1 and C_2. Table I presents the logical effort and parasitic delay values of some typical gates. Let the total length of interconnection be L, with the resistance and capacitance defined in terms of a minimum sized inverter as follows:

$$R_w = K_rR_{tinv}L,\quad C_w = K_cC_{tinv}L. \qquad (7)$$

K_r and K_c are constants dependent on technology and layout.

The total length L is divided into two segments L_1 and L_2, representing the portions that the two gates would drive:

$$L = L_1 + L_2,\quad R_w = R_{w1} + R_{w2},\quad C_w = C_{w1} + C_{w2}. \qquad (8)$$

The total delay is just the sum of the delays of the gates based on the interconnect segment driven by each:

$$D_2 = \tau\ ((g_1 + r_1)h_1 + p_1 + (g_2 + r_2)h_2 + p_2) \qquad (9)$$

The resistive and electrical efforts of the two gates are as follows (assuming gate 2 drives a load C_3 at the end):

$$r_1 = ((C_2 + C_{w1}/2)/(C_2 + C_{w1}))(R_{w1}C_1)/(R_{tinv}C_{tinv}) \qquad (10a)$$
$$h_1 = (C_{w1} + C_2)/C_1 \qquad (10b)$$
$$r_2 = ((C_3 + C_{w2}/2)/(C_3 + C_{w2}))\,(R_{w2}C_2)/(R_{tinv}C_{tinv}) \qquad (11a)$$
$$h_2 = (C_{w2} + C_3)/C_2 \qquad (11b)$$

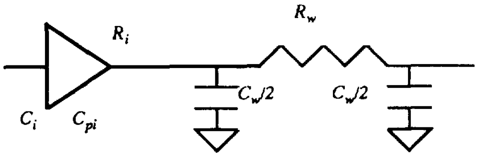

Fig. 1. Gate driving a lumped Π model of interconnect.

TABLE I

Logical Efforts and Parasitic Delays [5]

Gate Type	Logical Effort (g)	Parasitic Delay (p)
INVERTER	1	p_{inv}
2-input NAND	4/3	$2p_{inv}$
2-input NOR	5/3	$2p_{inv}$
2-input XOR	4	$4p_{inv}$

Expressing L_2 in terms of L_1 from (8), and setting $dD_2/dL_1=0$, optimal segment length L_1 is determined to be:

$$L_{1opt} = L/2 + (C_{tinv}/(2K_r)) ((g_2/C_2) - (g_1/C_1)) + (1/ (2K_cC_{tinv}))(C_3 - C_2) \tag{12}$$

The second term indicates that the weaker of the two gates (higher g) should drive a smaller segment of the interconnection, unless the gate is sized up (higher C) to achieve a smaller g/C ratio. The third term, which is due to the difference in gate input capacitances at the ends of the two segments, is significant only when this difference is comparable to the capacitance of a unit length (1 mm) of interconnection.

Equation (12) clearly shows how the optimal spacing is influenced by the g/C ratios of the two gates. It is now possible to take two arbitrary gates and size their transistors in order to achieve equal spacing, which may be important in practical chip layouts. For the special case of identical buffers considered in other studies, the optimal solution simply reduces to equal spacing.

The solution for the problem of spacing N logic gates along an interconnection (Fig. 2) is a simple extension of (12). Considering only the first two terms of (12) with the notion that the g/C ratio of each gate must now be compared with the average g/C ratio of the other $(N - 1)$ gates, optimal length for segment i is given by

$$L_{iopt} = L/N + (C_{tinv}/(NK_r))((\sum_j g_j/C_j) - ((N - 1) g_i/C_i)),$$

$$\text{where } j = 1 \ldots N \text{ and } j \neq i. \tag{13}$$

The third term of (12) can be generalized as follows:

$$\Delta L_{iopt} = (1/(NK_cC_{tinv}))(\sum_j C_{j+1} - (N - 1)C_{i+1}) \tag{14}$$

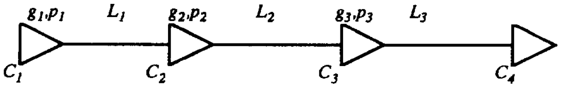

Fig. 2. Non-identical logic gates placed along a long interconnection.

IV. Optimal number of stages

The next step in the delay optimization methodology is to find the optimum number of repeater stages. We will start with the assumption that the sizes of the given set of logic gates have been initially adjusted such that equal spacing is optimal. Then, based on the optimum number of stages, designers have the option of adding additional inverters to minimize delay.

Extending (9) to a general N-stage network, and noting that the assumption of optimal/equal spacing implies equal g/C ratios for all stages, the optimum number of stages is easily computed by setting $dD_N/dN = 0$:

$$N_{opt} = (K_rK_c/ (2 (P_{av} + (g/C)C_{iav})))^{1/2}L \tag{15}$$

P_{av} is the average parasitic delay for the original (given) set of logic gates. C_{iav} is the average input capacitance of all the gates, and contributes to delay by adding load capacitance at the end of each interconnect segment. Therefore, P_{av} and C_{iav} contribute to the delay overhead due to repeaters.

It is clear from (15) that additional repeater stages are required as the line gets longer, but this must be traded-off against the overhead of parasitic delays. The general form of (15) matches the results of Dhar and Franklin [3] for inverters.

The total delay for an N-stage network, where N is the optimal number, is:

$$D_{Nopt} = \tau(K_cC_{tinv}g/C + K_rC_{iav}/C_{tinv} + (2K_rK_c(P_{av} + C_{iav}g/C))^{1/2}) L \tag{16}$$

As expected, the optimal delay is proportional to L rather than L^2. N_{opt} equalizes the intrinsic RC delay of the interconnect segments and the average delay overhead of repeaters.

If $N_{opt} > N$, inverters can be added to the original network to increase the number of stages by comparing the overhead of inverters with the average overhead of the original gates:

$$N_{add} = ((P_{av} + C_{iav}g/C)/(P_{inv} + C_{inv}g/C))^{1/2}(N_{opt} - N) \tag{17}$$

Under the constraint of maintaining a constant g/C at all repeater stages to allow equal spacing, C_{inv} is typically less than the optimal C_{iav} (since inverters have the lowest g). This means that the N_{add} predicted by (17) is slightly less than the optimum value, since the average delay overhead of repeaters is lower due to the added inverters. Section V discusses the optimal value for C_{iav}.

V. Optimal gate sizes

The final optimization step in this methodology is to size the transistors in the gates such that the gates can optimally drive the interconnect capacitance while not posing too large a load to the previous stage. It is assumed that each repeater stage consists of a single gate that must be sized.

Setting $dD_N/dC_{iav} = 0$, the optimum value for C_{iav} is:

$$C_{iav(opt)} = (g_{av}K_c/K_r)^{1/2} C_{tinv} \qquad (18)$$

g_{av} is the average logical efforts of all the gates. Equation (18) is independent of the number of stages is used. Again, the general form of (18) matches the results of Dhar and Franklin [3], but is not restricted to inverters.

Gate sizes must be adjusted using (18) under the constraint that g/C is equal for all stages at the end of the optimization. This simply means that the input capacitances (that were adjusted previously for equal spacing) must all be multiplied by $C_{iav(opt)}/C_{iav}$. Equation (15) is still valid because $(g/C)C_{iav}$ does not change.

Using $C_{iav(opt)}$, (16) can be simplified further and used to predict the minimum achievable delay with this methodology, subject to the inaccuracy mentioned in the previous section:

$$D_{NCopt} = \tau\, (2K_cK_r)^{1/2}((2g_{av})^{1/2} + (P_{av} + g_{av})^{1/2})\, L \qquad (19)$$

This final form of the delay equation equalizes the first two terms of (16) which represent the following components of delay: (a) delay due to gate output resistance and interconnect capacitance, and (b) delay due to interconnect resistance and gate input capacitance.

Another approach is to use cascaded buffers at the output of the logic gate at each repeater stage. The average parasitic delay P_{av} of the repeater stages will increase while the g/C ratio can be reduced greatly. Since no direct mathematical relationship has been observed between these two quantities, an iterative solution must be used. It does have the advantage that complex logic gates need not be made arbitrarily large.

VI. Examples

We will now present two examples that illustrate the application of the optimization methodology developed in this paper. The following technology/layout parameters are assumed:

p_{inv} (parasitic delay of an inverter) = 3.8.

C_{tinv} (input capacitance of minimum-sized inverter) = 3.0.

τ = 0.05 ns, K_r = 0.1, K_c = 10.0.

Both examples assume a total line length of 30 mm and start with an initial set of four logic gates that are needed to perform the computation. The sequence of the logic gates is listed below for each example (with input capacitances in parenthesis):

Example 1: 2-input NAND (3), 2-input NOR (5), INVERTER (1), 2-input XOR (4).

Example 2: 2-input XOR (8), 2-input XOR (8), 2-input XOR (8), 2-input NAND (3).

Table II summarizes the results, and clearly shows the substantial reduction in delay through optimization. The difference between equal spacing and optimal spacing is more significant in the first example where the logic gates are substantially different.

TABLE II

Delays Computed for Examples 1 and 2

Delay Case	Example 1	Example 2
Equal Spacing	24.4	30.98
Optimal Spacing	22.95	30.92
Optimal Delay	**9.53**	**11.16**
Minimum Delay	9.33	10.91

The "optimal delay" was computed using (15, (17) and (18), with N_{opt} and N_{add} rounded up. N_{add} was computed as 4 for both examples. This delay is within 2.5% of the potential minimum delay computed by varying the number of stages (N_{add} = 7). The main source of suboptimality in the "optimal delay" solution is that it depends on parameters that are averaged over the repeater stages (i.e., P_{av}, C_{iav} and g_{av}).

VII. Conclusion

This paper has presented a new methodology for optimizing delay of circuits that consist of general CMOS logic gates and long interconnections. The methodology is based on an extension of the concept of logical effort. This work has demonstrated that it is possible to find simple solutions for delay optimization in the presence of resistive interconnections, even when all the gates are not simple inverters. This is the most important contribution of this research.

A simple methodology has been presented that designers can use to obtain the optimal number of repeater stages and the optimal gate sizes, starting with a given network of logic gates. An important topic for further research is finding optimal solutions for structures other than single-gate repeaters (i.e., repeaters built with cascaded gates/buffers).

References

[1] H.B. Bakoglu and J.D. Meindl, "Optimal Interconnection Circuits for VLSI", IEEE Trans. Elec. Devices, Vol. 32, No. 5, pp. 903-909,May 1985.

[2] H.B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI.* Reading, MA: Addison-Wesley Publishing Co., 1990, Chapter 5.

[3] S. Dhar and M.A. Franklin, "Optimal Buffer Circuits for Driving Long Uniform Lines", IEEE JSSC, Vol. 26, No.1, pp. 32-40, January 1991.

[4] M. Nekili and Y. Savaria, "Optimal Methods of Driving Interconnections in VLSI Circuits", Proc. Intl. Symp. Circuits and Systems, pp. 21-24, May 1992.

[5] I.E. Sutherland and R.F. Sproull, "Logical Effort: Designing for Speed on the Back of an Envelope", in C.H. Sequin, Ed., *Advanced Research in VLSI.* Cambridge,MA: MIT Press, 1991.

[6] J. Rubinstein, P. Penfield and M. Horowitz,"Signal Delay in RC Networks", IEEE Trans. CAD, Vol. 2, No. 3, pp.202-211, July 1983.

[7] T. Lin and C.A. Mead, "Signal Delay in General RC Networks", IEEE Trans. CAD, Vol. 3, No. 4, October 1984.

[8] J.K. Ousterhout, "A Switch-Level Timing Verifier for Digital MOS VLSI", IEEE Trans. CAD, Vol. 4, No. 3, pp.336-349, July 1985.