Logic Gates as Repeaters (LGR) for Timing Optimization of SoC Interconnects

Arkadiy Morgenshtein
Bio-Medical Engineering Department, Technion, Haifa, Israel
arkadiy@tx.technion.ac.il

Michael Moreinis
Electrical Engineering Department, Technion, Haifa, Israel
moreinis@tx.technion.ac.il

Israel A. Wagner
IBM Haifa Labs, Haifa University, Mount Carmel, Haifa, Israel
wagner@il.ibm.com

Avinoam Kolodny
Electrical Engineering Department, Technion, Haifa, Israel
kolodny@ee.technion.ac.il

Abstract

LGR (Logic Gates as Repeaters) – a new methodology for delay optimization of SOC design with RC interconnects is described. Traditional interconnect segmentation by insertion of repeaters is generalized to segmentation by distributing the existing logic gates over interconnect lines, thus reducing the number of additional logically useless inverters. The application methodology of LGR is presented. Several logic circuits have been optimized by LGR and verified for delay and power. Analytical and simulated results were obtained, showing up to 25% improvement in performance, compared with traditional repeater insertion technique.

1. Introduction

Interconnect optimization has become a major design consideration in state-of-the-art sub-micron CMOS VLSI systems-on-chip. The growth of die size together with decreased line width makes wire delay more significant, compared with the active devices delay. In resistive wires, propagation delay increases proportionally to the square of the interconnect length because both the capacitance and resistance of the interconnect increase linearly with length. In modern SOC, characterized by long distances of signal propagation, the interconnect delay is likely to become a bottleneck of high-performance design. In order to handle resistive interconnect, post-routing design steps have been added, involving wire segmentation by repeater insertion [2].

Numerous studies explored various facets of the repeater insertion problem [4][5][6], adding inverters or buffers (double inverters) for amplifying logic signals on resistive wires between stages in a logic path. However, the usage of repeaters implies a significant power and area cost, without contributing to the logical computation performed by the circuit. A recent study [8] claims that in the near future, up to 40% of chip area will be used by inverters operating as repeaters and buffers. In many cases, the use of numerous logically-redundant repeaters seems to be a waste, because the logic gates themselves may function as repeaters due to their amplifying nature. The main idea of LGR (Logic Gates as Repeaters) concept is distribution of existing logic gates across interconnect; thus driving the partitioned interconnect without adding inverters to serve as repeaters.

The idea of overall delay optimization of a circuit path consisting of various CMOS logic gates together with long segments of resistive interconnect was presented by Venkat in [3]. Although logic gates were treated as repeaters, no general methodology was presented for finding the structures where this technique is applicable and efficient. The optimization proposed in [3] fails for circuits with exceptional ratios of input capacitances and logical efforts, and no correction was proposed for these cases.

This paper presents an analysis of LGR (Logic Gates as Repeaters) as an efficient delay optimization concept and proposes a new methodology for applying LGR in high-performance VLSI design. This method is suitable for delay optimization, while improving computational efficiency, power dissipation and area utilization (as compared with inverter-based repeater insertion) and can be efficiently integrated within a synthesis environment for VLSI, thanks to its low complexity.

2. Logic Gates as Repeaters (LGR) – The Concept

The process of interconnect segmentation by logic gates as repeaters is schematically shown in Figure 1. Before the segmentation, logic gates are concentrated in a single logic block driving a long interconnect load

(Figure 1a). After the distribution of logic gates over interconnect is performed, each logic gate has a related interconnect segment, as presented in Figure 1b.

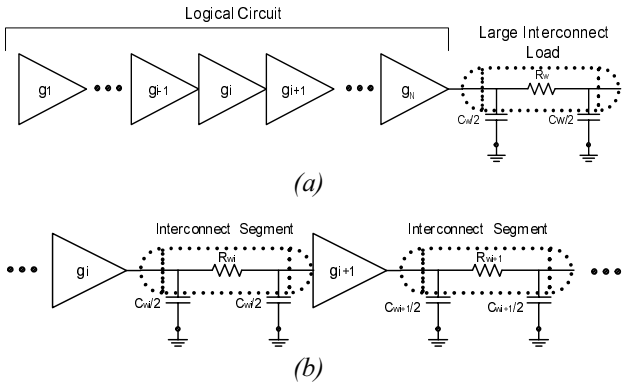

Figure 1. Logic gates with related interconnect load: (a) before segmentation, (b) sections i and i+1 after segmentation.

After segmentation, the delay of each pair of logic-interconnect segment can be calculated separately. The overall delay is the sum of delays of all the combined logic-interconnect segments.

For a capacitive load, the gate delay is expressed by the Logical Effort method [1]:

$$D_{gate} = t \cdot (gh + p) \tag{1}$$

where $\tau = R_{tinv}C_{tinv}$ is a process-dependent time constant, defined as the delay of an ideal inverter driving another identical inverter. R_{tinv} and C_{tinv} are effective resistance and input capacitance of an inverter. Parameter p is the parasitic delay of the gate and is related to capacitance of source/drain regions within the gate.

$$p = \left(R_t C_{pt}\right) \Big/ \left(R_{tinv} C_{tinv}\right) \tag{2}$$

where R_t, C_t and C_{pt} are output resistance, input capacitance and parasitic capacitance of the gate, respectively, and parameter g is called logical effort. It is independent of transistor sizes (depends only on the topology of the gate) and presents the relative ability of the gate to produce driving current.

$$g = \left(R_t C_t\right) \Big/ \left(R_{tinv} C_{tinv}\right) \tag{3}$$

The logical effort of an inverter is *1*. The logical effort g of a logic gate tells how much worse it is at producing output current than an inverter, given that each of its inputs may present only the same input capacitance as the inverter. Parameter h is called electrical effort and is the ratio of load capacitance of the gate to the capacitance of one of its inputs.

$$h = C_{load} / C_i \tag{4}$$

The interconnect delay of the wire segment, denoted as D_w, can be added to the gate delay using an Elmore delay model [10]:

$$D_w = x \cdot R_w \cdot C_w \tag{5}$$

where x is a fitting parameter. For the combined gate-interconnect segment the respective delay components of the ith segment are:

$$D_{gate} = \tau \cdot \left(g_i \cdot \left(\frac{C_{i+1} + C_{w_i}}{C_i} \right) + p_i \right) \tag{6}$$

$$D_{interconnect} = R_{w_i} \cdot \left(C_{w_i} \cdot x + C_{i+1} \right) \tag{7}$$

where

$$C_{w_i} = L_i \cdot C_{\text{int}}, \quad R_{w_i} = L_i \cdot R_{\text{int}} \tag{8}$$

L_i is the length of the wire segment, C_{int} and R_{int} are the capacitance and resistance per unit length, respectively. The overall delay for the logic path is therefore:

$$D_{tot} = \sum_{i=1}^{N} \left[\begin{array}{l} \tau \cdot \left(g_i \cdot \left(\dfrac{C_{i+1} + L_i C_{int}}{C_i} \right) + p_i \right) + \\ + \left(x \cdot L_i^2 R_{int} C_{int} + L_i R_{int} C_{i+1} \right) \end{array} \right] \tag{9}$$

where N is a number of gates and the capacitance C_{N+1} is the load capacitance of the circuit. Note that (9) assumes an ideal voltage source as the driver at the source of the logic path, e.g. the input to g_1 in Figure 1.

The closed-form expression (9) provides a basis for analysis and timing optimization of a critical logic path involving long-distance wiring, using Logic Gates as Repeaters (LGR).

3. Optimization Methods

3.1 Optimal Segmenting

The total length of the interconnect is L. The goal is to divide the total length into segments such that the delay expression in (9) will be minimized. The optimal length of each segment is derived by differentiation of the delay expression, performed for each of the interconnect segment lengths L_i.

There are two constraints on the goal function (9). The first constraint is

$$L_1 + L_2 + \ldots + L_n = L \tag{10}$$

Since the length of each segment must be non-negative due its physical nature, the second constraint applied to (9) is:

$$\forall i \quad L_i \geq 0 \tag{11}$$

Applying differentiation on (9) with constraint (10), and equating to zero, the resulting optimal length of the *i*th segment is:

$$L_{i_{opt}} = \frac{L}{N} + \frac{\tau\left(\sum_{j=1}^{N}\frac{g_j}{C_j} - (N-1)\frac{g_i}{C_i}\right)}{2\cdot N\cdot x\cdot R_{int}} + \frac{\sum_{j=1}^{N} C_{j+1} - (N-1)C_{i+1}}{2\cdot N\cdot x\cdot C_{int}} \qquad i \neq j \tag{12}$$

Note that in case where all gates have same g/C and same size (same input capacitance), an equal segmentation is obtained from (12).

The optimal segment length can also be expressed as:

$$L_{i_{opt}} = \frac{L}{N} + \frac{L\left(R_{av} - R_i\right)}{2\cdot x\cdot R_w} + \frac{L\left(C_{av} - C_{i+1}\right)}{2\cdot x\cdot C_w} \tag{13}$$

where the R_{av} and C_{av} are the average output resistance and input capacitance of the gates.

The first term represents equal partitioning of the total length, and the other terms represent corrections required because of different driving abilities and different input capacitances of the gates. If the driving gate is large (its R_i is small) the segment to be driven will be increased. Similarly, when the driven gate is large (C_{i+1} is large) the segment should be decreased to reduce the loading on the driving gate.

Closed-form expression (13) may fail when a weak gate drives a large gate. In this case the resulting segment length may be negative and thus violate the constraint in (11). Such a violation can be determined in a simple way by comparing the expression in (13) to zero. Once the violation is determined, a different value should be chosen as optimum. The following Lemma 1 defines a property of the delay function, used to select a non-negative length as optimum.

Lemma 1: *The function Dtot(L_1, L_2, …, L_n) in (9) under the constraint (10) is convex.*

Proof: *We first observe that Dtot is an n-dimensional paraboloid in R^{n+1} (i.e. the n+1 dimensional Euclidean space) with positive coefficients, hence it is convex. Since the constraint in (10) is an (n-1)-dimensional hyper-plane which is perpendicular to the hyper-plane Dtot=0, we get that the intersection of Dtot and (10) is an n-dimensional paraboloid which is again convex.*

According to Lemma 1, the resulting function has a single global minimum that is presented in (13). Supposing the global minimum is negative and thus invalid, we seek the closest-to-minimum point, where the expression does not violate the constraints. Hence, if the global minimum of (9) occurs for some negative L_i, the function must be monotonic in L_i in the range from 0 to the global minimum, and the constrained minimum must occur when this segment length L_i is set to 0 (11). The physical meaning of zero-length segment is placing two gates in close proximity to each other. All violations may be determined and avoided previous to optimization using this technique.

3.2 Scaling and Segmenting

Additional optimization in terms of segmenting may be obtained if we enlarge each of the gates in the logic chain by a constant factor *s*. We assume a uniform value of s for all the gates, to preserve the initial relative gate sizing performed by pre-layout methods such as Logical Effort. By performing this operation we enhance the driving ability of the gates in the logic chain. This optimization is similar to optimal repeaters, when not only the number of repeaters is optimized, but also their sizes. The delay expression for an *s*-enlarged gate chain is as follows:

$$D_{tot} = \sum_{i=1}^{N}\left[\begin{array}{l}\tau\cdot\left(g_i\cdot\left(\frac{sC_{i+1} + L_iC_{int}}{sC_i}\right) + p_i\right) + \\ +\left(x\cdot L_i^2 R_{int}C_{int} + L_iR_{int}sC_{i+1}\right)\end{array}\right] =$$

$$= \sum_{i=1}^{N}\left[\tau\cdot\left(g_i\cdot\left(\frac{C_{i+1}}{C_i}\right) + p_i\right) + x\cdot L_i^2 R_{int}C_{int}\right] + \tag{14}$$

$$+\frac{1}{s}\sum_{i=1}^{N}\frac{\tau g_i\cdot L_iC_{int}}{C_i} + s\sum_{i=1}^{N} L_iR_{int}C_{i+1}$$

The optimal scaling factor *s*, obtained by differentiation of (14), is:

$$s = \sqrt{\frac{\tau C_{int}}{R_{int}}\left(\frac{\sum_{i=1}^{N}\frac{g_i\cdot L_i}{C_i}}{\sum_{i=1}^{N} L_iC_{i+1}}\right)} \tag{15}$$

Note that assuming all gates are inverters and the interconnect is equally segmented as a result of (12), the resulting scaling factor is:

$$s = \sqrt{\frac{C_{int}R_{t_{inv}}}{C_{t_{inv}}R_{int}}} \tag{16}$$

which is similar to the scaling factor presented by Bakoglu [2] in the context of optimally sized repeaters.

The optimal segment lengths and optimal scaling factor can be obtained by iterative calculation of (13) and (15). In our experiments, convergence was reached in a few steps, usually less than 3 (for convergence within 1%).

4. Applicability

How suitable is the LGR method for a given circuit? An applicability criterion can be defined as the additional wiring produced by the LGR optimization. This wiring directly impacts the area and power dissipation. Thus, wire cost should be the main issue of LGR applicability analysis.

Figure 2 exemplifies the effect of LGR segmenting on wire cost. The initial interconnect length of a 3-to-8 decoder is 8L. After a uniform segmenting the resulting interconnect length is reduced to:

$$L_{resulting} = \left(4 \cdot \tfrac{1}{3} + 5 \cdot \tfrac{1}{3} + 8 \cdot \tfrac{1}{3}\right) \cdot L = 5.67 \cdot L$$

On the other hand, by performing LGR optimization on an 8-to-1 multiplexer structure, wire cost is increased from L to 3.33L. Thus, the additional wiring depends on circuit topology.

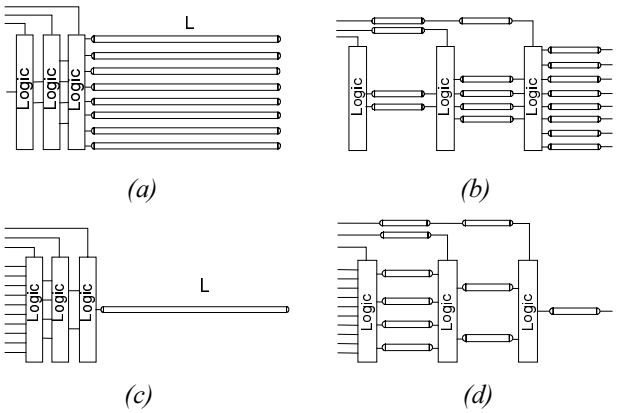

Figure 2. Segmenting of decoder (a,b) and multiplexer (c,d).

The decision about LGR applicability is carried out according to affordable wire cost slack. In order to describe the heuristic for LGR applicability analysis, the following terms are defined, assuming a rectangular circuit block with ports on the left and right sides only: *Clustering* is the division of the circuit into clusters by levelizing; each cluster contains all the gates at the same logic level-count starting from the circuit's primary inputs, as shown in the example of random logic in Figure 3. $Cluster_0$ contains the left I/O ports and $cluster_{n+1}$ contains the right I/O ports. Parameter Δ is a scalar defining the difference between the number of wires on opposite sides of each cluster. Parameter β defines the constraint on the maximal affordable value of Δ and is directly related to the total wire cost slack.

In order to minimize the total wire cost, the following pre-optimization procedure is defined:

If $\Delta > \beta$, the cluster should be shifted right

If $\Delta < \beta$, the cluster should be shifted left.

If $\Delta = \beta$, LGR optimization should be performed.

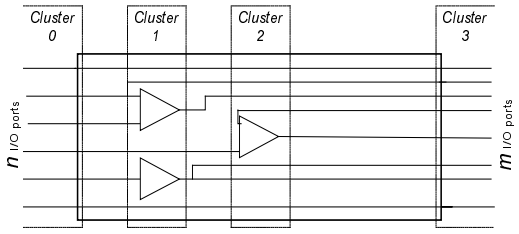

Figure 3. Example of Logic Clustering

By shifting the cluster we adjoin it with its left/right neighbor and thus produce a new combined cluster. The procedure is performed iteratively. At each step Δ is calculated for each cluster and the absolute maximum value indicates the cluster to be shifted. This heuristic is contained in the algorithm in Figure 4. In addition, a sweep of the constraining value β is performed, mapping the variety of solutions. This enables the designer to obtain the trade-off between delay and wire cost The complexity of this heuristic is $O(n \cdot k)$, where n is number of available values of β and k is the number of initial clusters in the design.

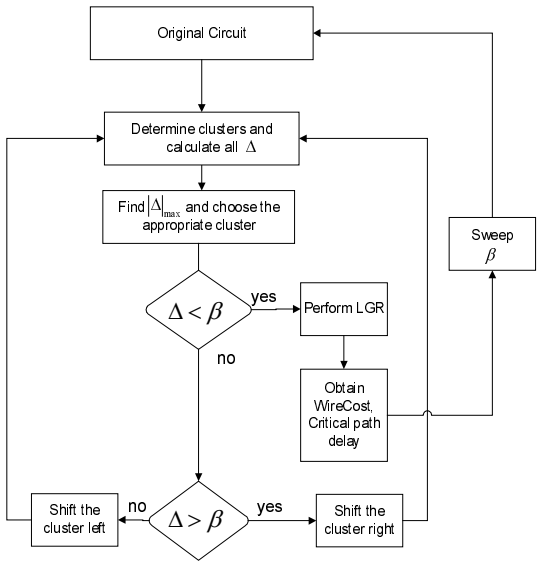

Figure 4. LGR algorithm.

The application of the algorithm on the logic block, containing six clusters, is exemplified in Figure 5 for $\beta = 0$. Performing LGR on the resultant clustering will maintain total wirelength unchanged.

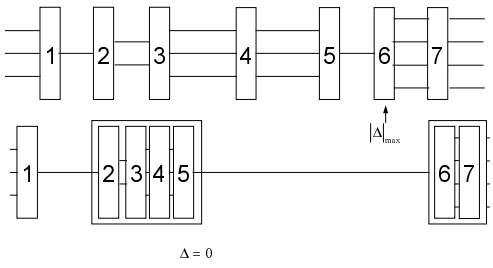

Figure 5. Application of LGR algorithm.

5. Results

In this Section LGR optimization is characterized and compared with Repeater Insertion technique. The Berkeley parameter extraction tool (BPTM) [7] was used to predict the parameters of 0.07 μm process for both interconnect and device BSIM3v3 models.

Fidelity analysis is performed by comparison between the LGR analytical results and SpectreS simulations results on the test circuit in Figure 6 (with minimal sizes of 0.07 μm /0.07 μm for NMOS and 0.28 μm /0.07 μm PMOS). LGR parameters were first analytically obtained from (12) and (15), and then the optimization was verified using SpectreS simulation.

In order to test the optimality of the analytically achieved values, $Length_1$ and $Length_2$ were swept near the analytically obtained parameters, while $Length_3$ was assumed to be the complement to the total length (1500 μm). The results are presented in Figure 7. The segmenting parameters obtained from analytical expressions (emphasized at $Length_1$=370µm, $Length_2$=280µm) produce a near-optimal solution, as compared to SpectreS results ($Length_1$=300µm, $Length_2$=225µm) with 75 μm resolution.

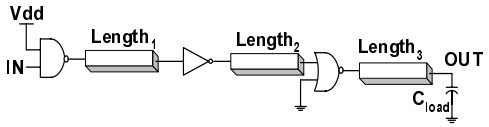

Figure 6. Fidelity analysis circuit

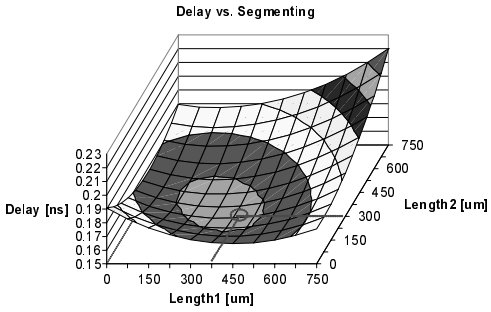

Figure 7. Results of Segmenting fidelity analysis

A scaling factor sweep around the analytically obtained optimum is shown in Figure 8. The optimal scaling factor (emphasized at ×70) obtained from analytic expression (12), is within 2% of the global optimum (×66) of SpectreS simulation.

In order to perform a comparison between the LGR and Repeater Insertion, the ripple-carry adder in Figure 9 has been selected as a test circuit. A common implementation of the adder embedded between two non-uniform logic structures in an ALU circuit [9], creates a critical path which drives long interconnect and is applicable for LGR optimization. The critical path of a 4-bit adder contains 17 CMOS gates (including the output inverters of OR and AND gates) and is indicated by the dashed line.

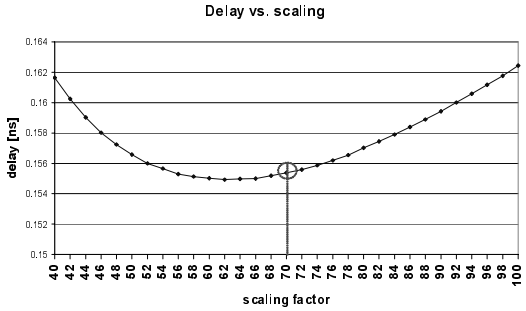

Figure 8. Results of Scaling fidelity analysis

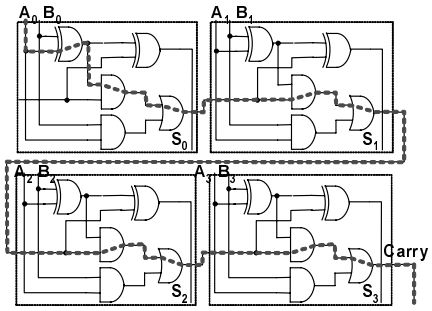

Figure 9. Ripple carry adder

The optimization parameters of LGR and Repeater Insertion techniques were extracted for the critical path of the adder and applied to SpectreS simulations. Four typical test cases were chosen: Low-tier and hjgh-tier metal interconnects with 1.5mm and 15mm lengths each. Timing optimization can be classified into three different categories. Each category refers to different design considerations and is separately analyzed and simulated.

The first category is specified by the need for timing improvement, without any gate modification (neither device sizing, nor repeater insertion). The basic LGR concept introduces timing optimization without increasing device count and sizes, by simple distribution of the initial circuit gates over the interconnect. Table 1 presents the timing gain obtained by LGR segmenting as compared to the Un-optimized circuit.

LGR technique provides a delay reduction of up to 18.5% by simple distribution of the logic gates over the interconnect.

Table 1. LGR Segmenting Optimization

	Un-optimized	*LGR Segmented*
Low-tier 1.5mm	*1.67 nsec*	*1.58 nsec*
Low-tier 15mm	*23.7 nsec*	*19.3 nsec*
High-tier 1.5mm	*2.54 nsec*	*2.45 nsec*
High-tier15mm	*25.4 nsec*	*21.5 nsec*

A second category is the optimization for *absolute* optimal timing. Traditionally, it refers to Optimal Repeater Insertion. LGR alternative is based on a combination of Segmenting & Scaling. The only important term is the delay, while other design considerations, like area and power can be ignored. The

results are presented in Table 2. Optimal Repeater Insertion scheme contains a tapered buffer at the input to reduce the delay of driving the large first repeater. LGR provides timing improvement of up to 98% as compared to the un-optimized circuit. LGR outperforms the Optimal Repeaters by 25% on average. In case of low-tier 15mm interconnect, the number of required repeaters (44) exceeds the number of logic gates (17). Thus the resulting delay of Optimal Repeaters is 6% better than LGR. In order to obtain further delay reduction by LGR, additional repeater stages can be inserted.

Table 2. LGR Segmenting & Scaling vs. Optimal Repeater Insertion

	LGR	*Repeater Insertion*
Low-tier 1.5mm	*0.3 nsec*	*0.45 nsec*
Low-tier 15mm	*2.0 nsec*	*1.92 nsec*
High-tier 1.5mm	*0.2 nsec*	*0.37 nsec*
High-tier15mm	*0.5 nsec*	*0.75 nsec*

Although the improvement is significant, the device scaling was unreasonably large, for both LGR and Repeater Insertion (about ×35 for low-tier metal wires and ×240 for high-tier metal wires), and a more practical approach is required. Thus, the third category is a trade-off. where delay reduction is an important term, but other design considerations, like power and area, are also valuable. This optimization category involves arbitrary (smaller than (15) but practical) sizes for both the gates and repeaters. A scaling factor of 4 was used in simulations. The timing results are presented in Table 3. Consistent advantage of the LGR over Repeater Insertion is observed.

Table 3. LGR vs. Repeater Insertion with reduced scaling factor

	LGR	*Repeater Insertion*
Low-tier 1.5mm	*0.60 nsec*	*0.79 nsec*
Low-tier 15mm	*4.96 nsec*	*6.13 nsec*
High-tier 1.5mm	*0.64 nsec*	*0.90 nsec*
High-tier15mm	*10.0 nsec*	*11.7 nsec*

The power-delay performance results of trade-off optimization are shown in Table 4. LGR technique improves the circuit performance by up to 61%, while Repeater Insertion obtains up to 47% improvement. LGR outperforms Repeater Insertion by 4% in average.

Table 4. LGR vs. Repeater Insertion – Power-Delay

	Unoptimized	*LGR*	*Repeater Insertion*
Low-tier 1.5mm	*0.83 f*	*0.77 f*	*0.60 f*
Low-tier 15mm	*60.0 f*	*23.0 f*	*32.0 f*
High-tier 1.5mm	*1.68 f*	*0.98 f*	*1.10 f*
High-tier15mm	*105 f*	*72.0 f*	*79.0 f*

The results of LGR in Table 4 are inferior only in case of short low-tier interconnect, where a small number of repeaters is required (4 in Repeater Insertion). In this case the scaling of all the 17 logic gates in LGR results in increased dynamic power dissipation, as compared to Repeater Insertion, where only four repeaters are scaled and logic gates remain in their initial sizes.

6. Conclusions

Logic Gates as Repeaters timing optimization was presented, based on distribution of core logic gates over resistive interconnect. Closed-form expressions for timing optimal segment lengths and scaling factor were obtained. The analytically obtained parameters were verified in SpectreS simulation, showing close-to-optimal solution. Simulation results of a ripple-carry adder show up to 18.5% improvement of delay by Segmenting and up to 98% improvement by Segmenting & Scaling compared to un-optimized circuit and up to 25% improvement over traditional repeater insertion technique. The results indicate that LGR is a viable improvement to traditional Repeater Insertion technique for VLSI interconnect optimization.

7. References

[1] I. Sutherland, B. Sproull, D. Harris, "*Logical Effort - Designing Fast CMOS Circuits*", Morgan Kaufmann Publishers, 1999.

[2] H.B. Bakoglu, "*Circuits, Interconnections and Packaging for VLSI*", Adison-Wesley, 1990.

[3] K. Venkat, "Generalized Delay Optimization of Resistive Interconnections through an Extension of Logical Effort", *ISCAS* 1993, NJ, USA, vol. 3, p 2106-2109.

[4] V. Adler and E.G. Friedman, "Repeater Design to Reduce Delay and Power in Resistive Interconnect," *IEEE Trans. on Circuits and Systems II,* Vol. CAS II-45, No. 5, May 1998, pp. 607-616.

[5] L.V. Ginneken, "Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay," *Proc. IEEE Int. Symp. on Circuits and Systems*, May 1990, pp. 865 - 868.

[6] C.J. Alpert, A. Devgan, S.T. Quay, "Buffer Insertion for Noise and Delay Optimization," *Proc. 34th ACM/IEEE DAC*, 1999, pp. 362-367.

[7] Berkeley Predictive Technology Model (BPTM).

[8] J.A. Davis, R. Venkatesan, K.A. Bowman and J.D. Meindl, "Gigascale integration (GSI) interconnect limits and n-tier multilevel interconnect architectural solutions," *Proc. Int. Workshop on System Level Interconnect Prediction (SLIP)*, San Diego, April 8-9 2000, pp. 147-148.

[9] E. Hokenek, R.K. Montoye, and P.W. Cook, "Second-Generation RISC Floating Point with Multiply-Add Fused," *IEEE J. of Solid-State Circuits*, vol. 25, October 1990,
pp. 1207-1213.

[10] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide band amplifiers," *J. Appl. Phys*., vol. 19, no. 1, 1948.