Logic Gates as Repeaters (LGR) for Area-Efficient Timing Optimization

Michael Moreinis, Arkadiy Morgenshtein,
Israel A. Wagner and Avinoam Kolodny

Abstract - LGR (Logic Gates as Repeaters) – a methodology for delay optimization of CMOS logic circuits with RC interconnects is described. The traditional interconnect segmentation by insertion of repeaters is generalized to segmentation by distributing logic gates over interconnect lines, reducing the number of additional, logically useless inverters. Expressions for optimal segment lengths and gate scaling are derived. Considerations are presented for integrating LGR into a VLSI design flow in conjunction with related methods. Several logic circuits have been implemented, optimized and verified by LGR. Analytical and simulation results were obtained, showing significant improvement in performance in comparison with traditional repeater insertion, while maintaining low complexity and small area.

I. Introduction

Interconnect optimization has become a major design consideration in state-of-the-art nanometer CMOS VLSI systems. Traditional design procedures have been developed assuming capacitive interconnect with negligible resistance [1][4][5][6]. In order to handle resistive interconnect, post-routing design steps have been added, involving wire segmentation and repeater insertion (Fig. 1b) such that every segment resistance is much smaller than the on-resistance of the driver [2][9]. Wire sizing and gate sizing have also been applied at this stage [11][14].

Numerous studies explored various facets of the repeater insertion problem [14][8][10][12][13], adding inverters or buffers (double inverters) for amplifying logic signals on resistive wires between stages in a logic path. Besides speed optimization, this amplification reduces noise and restores logic levels [13]. However, the usage of repeaters implies a significant cost in power and area, without contributing to the logical computation performed by the circuit. A recent study [17] claims that in the near future, up to 40% of chip area will be used by inverters operating as repeaters and buffers. The use of numerous logically-redundant repeaters seems to be a waste of area and power, 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 logic gates over interconnect; thus driving the partitioned interconnect without adding inverters to serve as repeaters (Fig. 1c).

The concept 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]. The formulation was based on an extension of the logical effort [1] concept to include the resistive load at the output of the logic gate in addition to capacitive load. Although in [3] logical gates were treated as repeaters, no general methodology was presented for finding the structures where this technique is applicable and efficient. A particular case of logic optimization with resistive interconnect was presented in [19], referring to optimal design of an SRAM address decoder.

This paper presents an analysis of LGR (Logic Gates as Repeaters) as an approach to delay optimization and proposes a methodology for applying LGR in high-performance VLSI design. This method can be combined with inverter-based repeater insertion to augment delay optimization, while improving power dissipation and area.

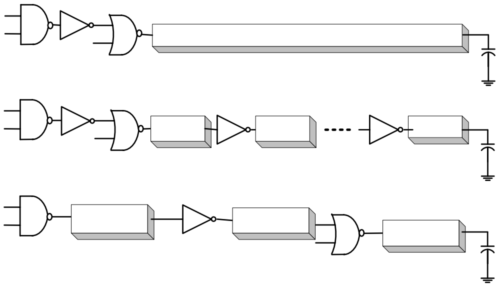

Fig. 1. (a) A logic path driving a long interconnect wire. (b) Repeater insertion on the long interconnect (c) LGR optimization: the logic gates are distributed over the interconnect and serve as repeaters.

II. Delay modeling

The process of interconnect segmentation by logic gates as repeaters is schematically shown in Fig. 2. Before the segmentation, logic gates are concentrated in a single logic block driving a long interconnect load (Fig. 2a). After the distribution of logic gates over interconnect is performed, each logic gate has a related interconnect segment, as presented in Fig. 2b. After segmentation, the delay of each pair of logic-interconnect segment can be calculated separately.

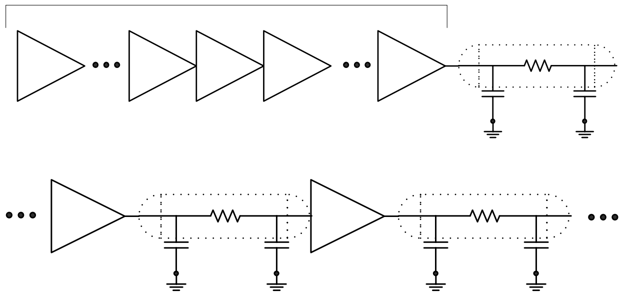

Fig. 2. Logic gates with related interconnect load: (a) before segmentation, (b) sections i and i+1 after segmentation.

The overall delay is the sum of delays of all the combined logic-interconnect segments. In practice, the logic path can be laid out with wire segments in both x and y directions. The basic concept of matching wire segment lengths to their driver gates is the same. The overall delay of the logic path was derived in [14] for the case where all gates are inverters. Following [14], we use Elmore delay model [7] for wire segment delay. We use the Logical Effort method [1] for gate delay calculation. For the combined i^{th} gate-interconnect segment in Fig. 2b the respective delay components are:

$$D_{gate} = \tau\left(g_i\left(\frac{C_{i+1}+C_{w_i}}{C_i}\right)+p_i\right) \qquad D_{wire} = R_{w_i}\left(0.5C_{w_i}+C_{i+1}\right) \tag{1}$$

where $\tau = R_{inv}C_{inv}$ is a technology-dependent time constant, defined as the delay of an ideal inverter driving another identical inverter. R_{inv} and C_{inv} are effective "on"-resistance and input capacitance of an inverter, respectively. Parameter p_i represents the parasitic delay of the gate and is related to capacitance of source/drain regions within the gate. C_i and C_{i+1} are input capacitance of gates i and $i+1$ respectively. Cw_i and Rw_i are the wire capacitance and resistance of segment i and can be replaced by:

$$C_{w_i} = L_i \cdot C_{\text{int}}, \quad R_{w_i} = L_i \cdot R_{\text{int}} \tag{2}$$

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:

M. Moreinis, A. Morgenshtein and A. Kolodny are with Department of Electrical Engineering, Technion, Israel (moreinis@tx.technion.ac.il, arkadiy@tx.technion.ac.il, kolodny@ee.technion.ac.il)
I. A. Wagner is with IBM Research Labs, Haifa, Israel (wagner@il.ibm.com)

$$D_{tot} = \sum_{i=1}^{N} \left[\tau \left(g_i \left(\frac{C_{i+1} + L_i C_{int}}{C_i} \right) + p_i \right) + \left(0.5 L_i^2 R_{int} C_{int} + L_i R_{int} C_{i+1} \right) \right] \quad (3)$$

where N is the number of gates and C_{N+1} is the load capacitance at the output of the circuit. Note that (3) **Error! Reference source not found.** assumes an ideal voltage source as the driver at the source of the logic path.

The closed-form expression **Error! Reference source not found.** provides a basis for analysis and timing optimization of a critical logic path involving long-distance wiring, using Logic Gates as Repeaters (LGR).

III. Optimization Methods

A. Optimal Segmenting

The total length of the interconnect along the logic path is denoted by L. The goal is to divide L into segments such that the delay expression in **Error! Reference source not found.**(3) will be minimized. The optimal length of each segment is derived by partial differentiation of the delay expression, performed for each of the segment lengths L_i.

There are two constraints on the goal function**Error! Reference source not found.**. The first constraint is:

$$L_1 + L_2 + ... + L_n = L \quad (4)$$

Since the length of each segment must be non-negative due its physical nature, the second constraint applied **Error! Reference source not found.**is:

$$\forall i \quad L_i \geq 0 \quad (5)$$

Applying differentiation on (3) **Error! Reference source not found.** with constraint (4), and equating to zero, the resulting optimal length of the i-th segment is:

$$L_{i_{opt}} = \frac{L}{N} + \frac{\tau \left(\sum_{j=1}^{N} \frac{g_j}{C_j} - (N-1) \frac{g_i}{C_i} \right)}{N \cdot R_{int}} + \frac{\sum_{j=1}^{N} C_{j+1} - (N-1) C_{i+1}}{N \cdot C_{int}} \qquad i \neq j \quad (6)$$

Note that in case where all gates are of the same type and size, an equal segmentation is obtained from (6).

The optimal segment length can also be expressed using (2) and (6) as:

$$L_{i_{opt}} = \frac{L}{N} + \frac{L\left(R_{av} - R_i\right)}{R_w} + \frac{L\left(C_{av} - C_{i+1}\right)}{C_w} \quad (7)$$

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 loading on the driving gate and wire segment.

Closed-form expression (7) 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 (5). Such a violation can be determined in a simple way by comparing the expression in (7) to zero. Once the violation is determined, a different value should be chosen as optimum. The following Lemma defines a property of the total delay function, used to select a non-negative length as optimum.

Lemma 1: *The function $D_{tot}(L_1, L_2, ..., L_n)$ in* ***Error! Reference source not found.*** *under the constraint (4) is convex.*

Proof: *We first observe that D_{tot} 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 (4) is an (n-1)-dimensional hyper-plane which is perpendicular to the hyper-plane $D_{tot}=0$, we get that the intersection of D_{tot} and (4) 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 (7). 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 **Error! Reference source not found.**(7) 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 due to constraint (5). The physical meaning of zero-length segment is that gates are placed in close proximity to each other, or "merged". All violations may be determined and avoided previous to optimization using this technique.

B. Scaling and Segmenting

Additional speed-up 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. The delay expression for a logic chain with gates enlarged by factor s is:

$$D_{tot} = \sum_{i=1}^{N} \left[\tau \left(g_i \left(\frac{sC_{i+1} + L_i C_{int}}{sC_i} \right) + p_i \right) + \left(0.5 L_i^2 R_{int} C_{int} + L_i R_{int} s C_{i+1} \right) \right] = \quad (8)$$
$$= \sum_{i=1}^{N} \left[\tau \left(g_i \left(\frac{C_{i+1}}{C_i} \right) + p_i \right) + 0.5 L_i^2 R_{int} C_{int} \right] + \frac{1}{s} \sum_{i=1}^{N} \frac{\tau g_i L_i C_{int}}{C_i} + s \sum_{i=1}^{N} L_i R_{int} C_{i+1}$$

The optimal scaling factor s, obtained by differentiation of (8), is:

$$s = \sqrt{ \frac{\tau C_{int}}{R_{int}} \left(\sum_{i=1}^{N} \frac{g_i \cdot L_i}{C_i} \right) \Big/ \left(\sum_{i=1}^{N} L_i C_{i+1} \right) } \quad (9)$$

Note the special case where all gates are inverters and the interconnect is equally segmented. Equation (9) yields the scaling factor:

$$s = \sqrt{\left(C_{int} R_{inv}\right) / \left(C_{inv} R_{int}\right)} \quad (10)$$

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 (7) and (9). In our experiments, convergence to within 1% of the optimal delay was reached in a few steps, usually less than 3.

The scaling factor in expression (9) obtains the global timing optimum. However, for power-efficient design, a more moderate scaling should be considered. This is due to the fact that the curve of delay vs. scaling factor is almost flat around its minimum. Hence a significant reduction in power and area can be gained by a slight increase in delay [20].

IV. Applicability of LGR within a VLSI design flow

LGR should be integrated into a complete VLSI design flow that is oriented towards interconnect optimization [21]. The integration involves various complications, such that LGR may be applicable for only a subset of the paths in a given circuit. A first applicability criterion can be based on the additional wirelength produced by LGR optimization. The objective is to improve delays while making sure that the total wirelength in the circuit does not grow too much. Consider for example a critical path within a datapath block, such that gate location changes are all along one dimension. Besides wires on the critical path, all the other inputs of each relocated gate must be connected, thus increasing or decreasing the total length of wires in the circuit. This impacts the area and power dissipation. Thus, variation of wirelength cost can be one of the main concerns in LGR applicability analysis. Fig. 3 exemplifies the effect of LGR segmenting on wire cost. The initial total interconnect length of a 3-to-8 decoder is *8L*. After a uniform segmenting to optimize timing, the resulting interconnect length is reduced to *5.67L*. On the other hand, by performing a similar LGR timing optimization on an 8-to-1 multiplexer structure, total wire length is increased from *L* to *3.33L*. Thus, when segmenting for minimum delay, the change in wire cost depends on circuit topology. A Placement optimizer targeted at minimum total wirelength would typically lead to different results, with some long unsegmented wires. For example, a minimum-wirelength placer could move all the logic in Fig. 3a all the way to the right, hence wirelength would be reduced even more than in Fig. 3b. However, the critical path delay would be worse than in Fig. 3b. Therefore, a method is needed to optimize the timing of the critical path using LGR, while keeping a limit on the extra wiring this might cause.

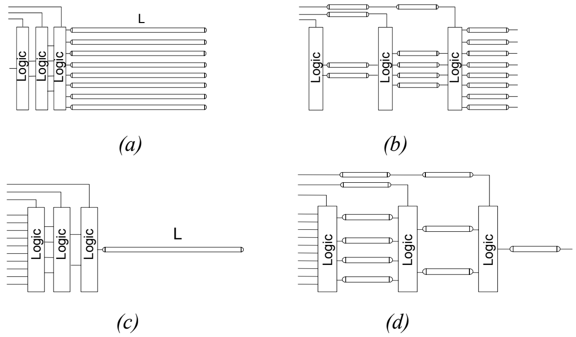

Fig. 3. Segmenting of decoder (a,b) and multiplexer (c,d).

A simple wirelength cost heuristic for LGR applicability analysis in a one-dimensional circuit is illustrated in Fig. 4, assuming a rectangular circuit block with ports on the left and right sides only. The heuristic serves as a pre-optimization step, prior to LGR. It merges gates into clusters, such that the number of wires between adjacent clusters will not exceed a predefined limit β. Afterwards, LGR segmentation will be used to move entire clusters of gates, for minimizing the critical path delay. The application of this procedure on a logic block is demonstrated in Fig. 4 for $\beta = 1$, resulting in a chain with three combined clusters, such that LGR can relocate the second cluster for minimal delay.

For more general kinds of circuits, integration of LGR into the design flow involves further issues. The evaluation of additional wirelength must be extended to two-dimensional floorplans. Also, other applicability criteria should be considered: in practice, gates may not be relocated and distributed over long interconnect without violating design hierarchy. Ideal gate location may not be realizable because it might be occupied by a large block. Relocation and resizing of gates may necessitate to change placement of other cells. Thus, interaction with placement is required, e.g. iterative placement, routing and LGR optimization.

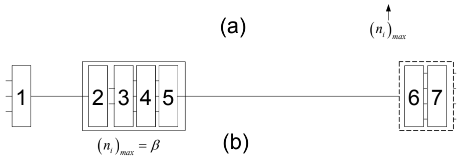

Fig. 4. Example of gate clustering heuristic to reduce additional wiring cost. (a) Initial clustering. (b) In the final clustering, the middle cluster can be relocated along the critical path to obtain minimal delay.

In the context of a whole circuit, improvement of one path can result in timing degradation of another path. The optimization presented in section III is based on single critical path optimization. However, it can be extended to treat more than one critical path. The solution for multiple-path optimization involves the definition of a more general goal function, using Equation (3) or (8), in which several delay paths make their contribution.

V. Power and Area Comparison

As a result of aggressive sizing, the circuit area and the power dissipated by up-scaled gates are considerably increased. Hence, repeater insertion may be preferred over LGR for power and area considerations, because an inverter consumes the smallest possible area in comparison with other gates having the same current drive capability. In this section, an analytical comparison between the LGR and repeater insertion is presented for dynamic power considerations, assuming that similar path delay is obtained by both techniques.

The dynamic power is related to total capacitance of the system. Hence, the comparison between total capacitances of LGR method and traditional repeater insertion technique provides an estimation of power dissipation.

The total capacitance in the logic-interconnect system is calculated as:

$$C_T = C_w + C_{devices} \tag{11}$$

where C_w represents the wire capacitance and $C_{devices}$ represents the gates capacitance (logic chain and repeaters). The total capacitance of the circuit optimized by LGR and Repeater Insertion is:

$$C_{LGR} = C_w + C_{gates} s_{LGR} \qquad C_{rep} = C_w + C_{gates} + C_{inv} N_{rep} s_{rep} \tag{12}$$

where s_{LGR} is the optimal scaling factor for gates in LGR technique (9), and s_{rep} is optimal scaling factor for inverter-based repeaters by (10), N_{rep} is the optimal number of optimally scaled repeaters for a wire of length L, as derived in [2], C_{gates} is the total capacitance of the initial circuit (prior to scaling) and C_w is a wire capacitance assumed to be the same for both optimizations (considering the critical path). The two alternatives are equivalent in terms of power if the expressions in (12) are equal:

$$C_{gates} \sqrt{\frac{\tau C_{int}}{R_{int}} \left(\sum_{i=1}^{N} \frac{g_i \cdot L_i}{C_i} \right) \Big/ \left(\sum_{i=1}^{N} L_i C_{i+1} \right)} = C_{gates} + C_{inv} N_{rep} \sqrt{\frac{C_{int} R_{t_{inv}}}{C_{inv} R_{int}}} \tag{13}$$

Finally, LGR is preferable in terms of power if:

$$N_{rep} > \sum_{i=1}^{N} C_i \cdot \left[\sqrt{\tau \left(\sum_{i=1}^{N} L_i \frac{g_i}{C_i} \right) \Big/ \left(\sum_{i=1}^{N} L_i C_{i+1} \right)} - 1 \right] \Bigg/ \sqrt{\frac{R_{t_{inv}}}{C_{t_{inv}}}} \tag{14}$$

In particular, for a chain of *N* identical gates with logical effort *g*, LGR is preferable in terms of power if:

$$N_{rep} > N \cdot \sqrt{g} \tag{15}$$

In terms of delay, it would be beneficial to combine the two techniques: use smaller wire segments and add some repeaters. For short interconnect with a substantial number of gates N in the logic chain, LGR will be less efficient than repeater insertion in terms of dynamic power. In this case the scaling of all gates will be a waste of area and power. Still, LGR can be modified to be advantageous over classical Repeater Insertion, if a subset of the gates in the chain are used as the repeaters.

In the discussion above we ignored short-circuit power dissipation and slew-dependent delay in gates. These effects were accounted for in Spice simulations of test circuits (shown in the next section).

VI. Results and Application

In this Section LGR optimization is characterized and compared with Repeater Insertion. We compare the two techniques in three different categories of circuit optimization: In the first category the designer tries to reduce delay without increasing circuit area and power. In the second category the purpose is to obtain the smallest possible path delay by optimal driver scaling, at any cost in circuit area (and power). The third category is a trade-off, where delay reduction is important, but area and power are also valuable. Hence, a pragmatic scaling factor is chosen, smaller than the optimal value given by (9) [20].

A. A simple example

To validate our delay model and optimization results, and to demonstrate the advantages of using logic gates as repeaters, we use the simple circuit shown in Fig. 1. The Berkeley parameter extraction tool (BPTM) [16] was used to predict parameters of 0.07µm process for both interconnect and device BSIM3v3 models, and fidelity analysis was performed by comparison between LGR analytical results and Spice simulation results on the test circuit in Fig. 1c. LGR segmenting and scaling parameters were first analytically obtained from (6) and (9), and then the optimization was verified using Spice simulation.

In order to test the optimality of the analytically obtained values, L_1 and L_2 were swept near their analytically obtained optimal values, while L_3 was computed as the complement to the total length (1500µm). The results are presented in Fig. 5. The segmenting parameters obtained from analytical expressions (emphasized in Fig. 5 at L_1=370µm, L_2=280µm) produce a near-optimal solution, as compared to Spice results (L_1=300µm, L_2=225µm) with 75µm resolution. A scaling factor sweep around the analytically obtained optimum is shown in Fig. 6. The optimal scaling factor obtained from expression (6) is ×70, while the global optimum of Spice simulations is at scale factor of ×66. This difference corresponds to less than 2% in delay.

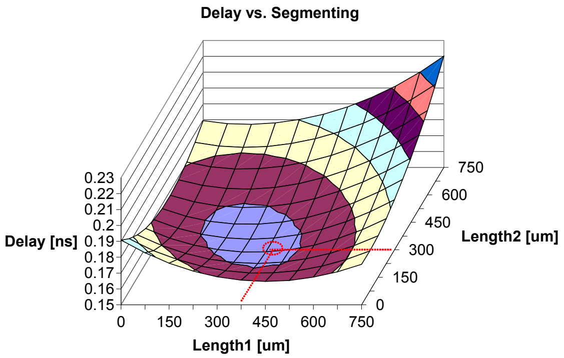

Fig. 5. Delay vs. Segmenting variations

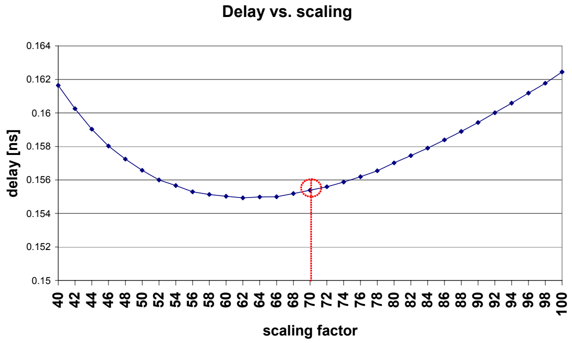

Fig. 6. Delay vs. Scaling factor variation

It is interesting to compare LGR optimization for this circuit (Fig. 1c) with traditional Repeater Insertion (Fig. 1b). In the first category of optimization, where small circuit area is the primary goal, LGR reduces the wire delay due to segmentation while keeping the same gate sizes as in Fig. 1a. On the other hand, repeater insertion involves additional area for inverters. Furthermore, when the inserted inverters are small, their added stage delay typically exceeds their contribution to reducing wire delay. Hence, LGR is better than repeater insertion for the first category of optimization.

In the second category, where delay reduction is achieved at the cost of additional area, insertion of four inverters is required on a low-tier metal wire of 1500µm using the derivation of [2], and including output capacitances of repeaters in the model. The line is uniformly segmented as shown in Fig. 7, and each inverter is sized by ×70 [2]. A cascade of tapered drivers containing two stages (with tapering factor of ×3.8) was constructed before the first repeater [15], to avoid an under-sized gate as driver of the first repeater. The resulting propagation delay obtained for the circuit in Fig. 7 is 0.265ns, while LGR segmenting and scaling (Fig. 1c) provides a delay of 0.155ns. The contribution of wire segment delays is approximately the same for both LGR and Repeater Insertion (about 0.1nsec). Moreover, the delay across wire segments and their direct drivers is about the same in both circuits. However, there are extra CMOS stages in the circuit of Fig. 7 which do not drive wire segments, and they contribute a total additional propagation delay of about 0.11nsec. Hence, LGR is preferable for this circuit.

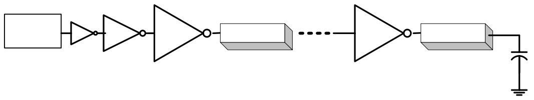

Fig. 7. Repeater insertion solution for the simple circuit in Fig. 1

As a representative case for the third optimization category we chose to compare optimal repeater insertion to LGR segmenting and scaling, where the scaling factor (for both techniques) was reduced (×7) rather than optimal. In this case the cascade of tapered drivers contained only one stage, with tapering factor of ×3.8.

For this circuit example LGR outperforms repeater insertion in both speed and power (Delay: *0.355nsec* for LGR and *0.397nsec* for Repeater Insertion., Power: *561nW* for LGR and *692nW* for Repeater Insertion, Power-Delay: *0.2fJ* for LGR and *0.25fJ* for Repeater Insertion)

B. A decoder circuit

Another circuit that was analyzed is a 8 to 256 decoder. The decoder should be placed within a typical cross-bar structure, which contains long interconnect. Fig. 8 shows the cross-bar design and internal structure of the decoder before and after LGR optimization.

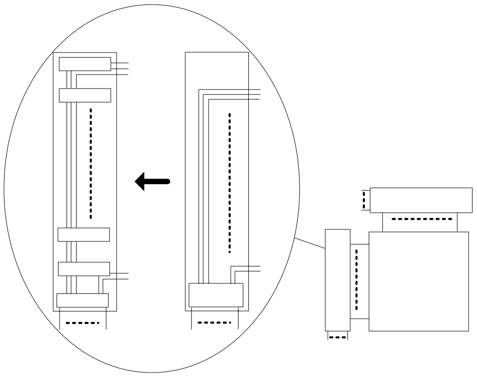

Fig. 8. Decoder Structure before and after LGR optimization

The critical path of the decoder contains several gates to be distributed over outgoing long interconnect along the vertical axis. The logic gates are originally placed close to inputs. Using LGR optimization methodology, the propagation delay over the critical path can be improved by segmenting the interconnect. The symmetrical structure of the decoder is suitable for LGR, since all the paths are simultaneously improved. The critical path of the decoder was optimized according to the methodology proposed in Section III. The results of segmenting optimization are presented in Table 1. The simple distribution of the critical path logic gates over the interconnect obtains timing improvement of up-to 27%.

The LGR segmenting and scaling results are compared with traditional repeater insertion and presented in Table 2. For intermediate lengths of interconnect the LGR shows up to 55% improvement over Repeater Insertion. For long interconnect, where a significant number of additional repeater stages are required, the Repeater Insertion outperforms LGR by up to 70%. However, it requires 44 additional functionally useless repeaters. Generally, in case of a short logic chain, the LGR optimization technique is preferred for intermediate interconnect length. For long interconnect, where many repeaters are required, LGR can be combined with addition of some repeater stages.

Table 1. 8-to-256 Decoder critical path delay for segmenting

	Unoptimized	*LGR Segmenting*
Low-tier 1.5mm	*2.28 nsec*	*2.15 nsec*
Low-tier 15mm	*34.6 nsec*	*25.2 nsec*
High-tier 1.5mm	*3.62 nsec*	*3.47 nsec*
High-tier15mm	*36.4 nsec*	*34.9 nsec*

Table 2. 8-to-256 Decoder critical path delay for segmenting and scaling

	LGR	*Repeater Insertion*
Low-tier 1.5mm	*0.188 nsec*	*0.268 nsec*
Low-tier 15mm	*5.45 nsec*	*1.65 nsec*
High-tier 1.5mm	*0.086 nsec*	*0.194 nsec*
High-tier15mm	*0.557 nsec*	*0.542 nsec*

C. *Synthesized random logic circuits*

In order to demonstrate the potential benefit of LGR technique for random logic blocks, several test circuits were synthesized using standard commercial tools, in a long and narrow rectangular floorplan, where significant wire lengths are involved. The produced layout was analyzed to extract the circuit physical parameters as well as to determine the physical delay of the critical path. Subsequently, LGR optimization was manually applied to the most critical path of each circuit, in order to assess the potential of LGR to gain further timing improvement. The results are presented in Table 3, where: *cmp* – 36-bit comparator, *mul* – 16-bit multiplier, *alu* – 6-bit ALU. The index denotes different floorplan size ratios (1:5 and 1:25). The LGR segmenting improves the critical path propagation delay by up to 41% as compared to the initial synthesis. The LGR scaling applied together with segmenting obtains further optimization of the critical path propagation delay and obtains improvement of up to 56% as compared to the initial circuit produced by the commercial physical synthesis tool. The results were verified by Spice simulations, showing that the timing improvement obtained by LGR is consistent with the analytically obtained results. These examples indicate that LGR can provide timing improvement if integrated into the design flow of random logic synthesis.

VII. Conclusions

Timing optimization based on distribution of logic gates over resistive-capacitive interconnect has been presented. The logic gates thus serve also as repeaters, driving wire segments. Closed-form expressions for timing-optimal segment lengths and scaling factor were obtained for gates. The applicability of Logic Gates as Repeaters (LGR) within a design flow was analyzed by defining heuristics based on wire cost and power dissipation parameters. Guidelines were presented for combining LGR with traditional repeater insertion. The analytically obtained parameters were verified by simulation, showing close-to-optimal solution. Results of design experiments indicate that LGR can provide viable improvement to traditional Repeater Insertion for VLSI interconnect optimization. The technique should be useful whenever gates can be relocated and distributed over long interconnect without violating design hierarchy and modularity. In particular, it can be useful when synthesis is used on very large logic blocks, containing long internal wires with significant wire delays. Such integration requires interaction with physical design automation steps.

Table 3. LGR results applied in Physical Synthesis Flow

Test Circuit	*Critical Path [μm]*	*As synthesized [nsec]*		*LGR Segmenting [nsec]*		*LGR Scaling [nsec]*
		Analytical	*Spice*	*Analytical*	*Spice*	*Analytical*
cmp1	*1665*	*5.07*	*5.88*	*4.71*	*5.50*	*4.59*
cmp2	*7335*	*6.84*	*7.48*	*5.32*	*6.53*	*4.27*
mul1	*6590*	*4.51*	*7.35*	*3.96*	*6.80*	*3.71*
mul2	*16256*	*10.2*	*11.7*	*6.63*	*8.90*	*4.53*
alu1	*2320*	*1.82*	*2.75*	*1.48*	*2.30*	*1.36*
alu2	*11600*	*3.95*	*4.98*	*2.32*	*3.19*	*1.74*

VIII. 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, pp. 194-219.

[3] K. Venkat, “Generalized Delay Optimization of Resistive Interconnections Through an Extension of Logical Effort”, *ISCAS*, NJ, USA, vol. 3, pp 2106-2109, 1993.

[4] H.C. Lin and L.W. Linholm, “An optimized output stage for MOS integrated circuits,” *IEEE J. Solid-State Circuits*, vol. SC-10, no.2, pp.106-109, Apr.1975.

[5] R.C. Jaeger, “Comments on ‘An optimized output stage for MOS integrated circuits,’” *IEEE J. Solid-State Circuits*, vol. SC-10, no.2, pp.185-186, Apr.1975.

[6] B.S. Cherkauer and E.G Friedman, “Design of Tapered Buffers with Local Interconnect Capacitance,” *IEEE J. Solid-State Circuits*, vol. 30, no. 2, pp. 151-155, February 1995.

[7] 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.

[8] V. Adler and E. G. Friedman, "Uniform Repeater Insertion in RC Trees," *IEEE Tran. on Circuits and Systems I: Fundamental Theory and Applications,* Vol. 47, No. 10, pp. 1515-1523, October 2000.

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

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

[11] J. Lillis, C. K. Cheng and T. T. Y. Lin, "Optimal Wire Sizing and Buffer Insertion for Low Power and a Generalized Delay Model," *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, pp. 138-143, Nov. 1995.

[12] C. J. Alpert and A. Devgan, "Wire segmenting for improved buffer insertion, " *in Proc. Design Automation Conf*., Anaheim, CA, June 1997

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

[14] C. Chu and D. F. Wong, “Closed Form Solution to Simultaneous Buffer Insertion / Sizing and Wire Sizing,”. *ACM Trans. on Design Automation of Electronic Systems*, vol. 6, no. 3, pp. 343-371, July 2001.

[15] S. Dhar and M. A. Franklin, “Optimum buffer circuits for driving long uniform lines,” *IEEE J. Solid-State Circuits*, vol. 26, pp. 32–40, Jan. 1991.

[16] Berkeley Predictive Technology Model (BPTM), *www-device.eecs.berkeley.edu/~ptm/introduction.html*.

[17] 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. of the Int’l Workshop on System Level Interconnect Prediction*, San Diego, April 8-9 2000, pp. 147-148.

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

[19] B. S. Amrutur and M. A. Horowitz, "Fast Low-Power Decoders for RAMS," *IEEE J. Solid State Circuits*, vol. 36, pp. 1506-1514, Oct. 2001.

[20] Yu Cao, Chenming Hu, Xuejue Huang, Andrew B. Kahng, Sudhakar Muddu, Dirk Stroobandt, Dennis Sylvester, "Effects of Global Interconnect

Optimizations on Performance Estimation of Deep Submicron Designs", *Digest of Tech. Paper, ICCAD*, pp. 56-61, Nov. 2000.

[21] J. Cong, "An Interconnect-Centric Design Flow for Nanometer Technologies," *Proc. of the IEEE*, vol. 89, No. 4, April 2001, pp 505-528