Post-Layout Optimization for Deep Submicron Design

Koichi Sato Masamichi Kawarabayashi
Hideyuki Emura Naotaka Maeda

Advanced CAD Development Laboratory
ULSI Systems Development Laboratories
NEC Corporation, Kawasaki, Kanagawa 211 Japan

Abstract

To reduce the number of synthesis and layout iterations, we present a new delay optimization technique, which inserts buffers based on back-annotated detailed routing information. During optimization, inserted buffers are assumed to be placed on the appropriate location of original wires so as to calculate accurate wire RC delay. With forward-annotated location information of inserted buffers, the layout system attempts to preserve patterns of original wires using the ECO technique. Our experimental results show that this technique combined with the conventional gate sizing technique achieves up to 41.2% delay reduction after the initial layout.

1 Introduction

In the case of delay optimization using logic synthesis, delays of paths are calculated using the intrinsic delay of library blocks and the estimated wire delay based on the number of fanouts. The estimated wire delay is usually inaccurate in comparison with the post-layout delay because it is deeply related to wire length based on the location of the connected gates on the circuit.

In the past, the difference between the estimated wire delay in the logic synthesis phase and the routed wire delay was relatively small because the total intrinsic gate delay was much larger than the total wire delay on each path. As design rules have shrunk into deep submicron recently (*e.g.* 0.35 or 0.25 μm), precise consideration of wire delay has become important.

1.1 Previous/Related works

Lin *et al.*([1]), Singh *et al.*([2]) and Yoshikawa *et al.*([3]) proposed algorithms to reduce the delay of the critical paths of the pre-layout circuits.

Related earlier efforts in the area of post-layout delay optimization concentrated on the accurate estimation of wire delay, because of the increasing contribution of wire delay to path delay. Pedram *et al.*([4],[5]) estimated wire length based on the location of gates in the circuit using quick global placement before detailed placement and routing. Vaishnav *et al.*([6]) provided a fanout optimization technique considering routability. Some commercial logic synthesis systems can insert buffers to optimize the delay after back-annotating the wire capacitance and wire RC delay, but they don't take into account the effect of the location of the inserted buffer.

Kannan *et al.*([7]) proposed a buffer insertion and gate sizing technique in the placement phase in order to reduce the number of iterations between the logic synthesis phase and the layout phase when the delay constraints are not satisfied.

These earlier efforts only considered the capacitance of wires and input pins of gates. For half micron design rules, wire capacitance dominates the total wire delay. On the other hand, in the case of deep submicron rules (*e.g.* 0.35 or 0.25 μm), wire RC delay must be considered in delay optimization because wire resistance dominates much more than the previously.

Aoki *et al.*([8]) provided a buffer insertion method taking wire RC delay into account in the layout system. The method creates a graph that consists of the terminals and Steiner points as nodes, and inserts buffers on the edges between the nodes. Since wire resistance and capacitance are estimated by the Manhattan distance between nodes, this approach is less accurate than detailed routing.

Lillis *et al.*([9]) proposed an optimal buffer insertion algorithm using Dynamic Programming which minimizes the maximum delay considering wire capacitance and resistance. But they ignore the wire resistance change upon inserting buffers.

1.2 Our approach

We propose a post-layout optimization algorithm considering wire RC delay exactly.

Not only the wire capacitance but also the resistance for each segment of the wire should be back-annotated to our logic synthesis system in order to insert buffers for delay optimization. We assume that the inserted buffers can be placed at an appropriate location relative to the original wires so as to estimate the wire resistance precisely. After delay optimization, the location of the buffers is forward-annotated to a layout system which attempts to place them at the annotated locations and to preserve the patterns of

33rd Design Automation Conference ®

DAC 96 - 06/96 Las Vegas, NV, USA

the original wires using the ECO technique.

The basic terminology is introduced in Section 2. Section 3 reviews the timing analysis model and equations, especially the wire RC delay calculation equations. Section 4 deals with our approach which consists of the back-annotation, the post-layout optimization process and the engineering change order in the layout system. The experimental results are provided in Section 5. Finally, Section 6 provides the conclusion.

2 Definitions

The definitions of the variables in this paper, are summarized in Table 1.

Table 1: Terminology

n_i	gate pins and branching points of wire $1 \leq i \leq N$: N is the total number of gate pins and branching points of wire
$S_{i,j}$	wire segment between n_i and n_j
$R_{i,j}$	resistance of the segment $S_{i,j}$
$C_{i,j}$	capacitance of the segment $S_{i,j}$
CI_j	if n_j is pin of the gate, then pin capacitance of n_j, else 0

3 Timing analysis

A path delay is calculated as the summation of the delay for each gate. The delay for each gate is defined in equation (1).

$$tpd = tpd0 + tpd_{CL} + tpd_{RC} \tag{1}$$

where $tpd0$ is the intrinsic gate delay which is defined in a technology library, tpd_{CL} is the load delay at the output pins of the gate, tpd_{RC} is wire RC delay. tpd_{CL} is calculated by equation (2).

$$tpd_{CL} = \gamma \cdot Z \cdot CL \tag{2}$$

where CL is the load at the output pin which is the summation of the capacitance of the wire and the input pins of the fanout gates, Z is the impedance at the output pin of the gate. γ is the constant value which is calculated from the supply voltage and the basic threshold voltage. There is also another load delay calculation method which calculates tpd_{CL} from the load CL and the input slew delay.

tpd_{RC} is calculated from the resistance and capacitance for each segment of the wire. The wire RC delay was much smaller than any other delays until the half micron technology. Thus it was always ignored or specified as a constant value. In deep submicron technology, wire RC delay dominates the delay of the path.

Therefore, it must be re-calculated or re-estimated precisely when the netlist is changed by the logic synthesis system. Our system allows this re-estimation.

3.1 Elmore delay calculation

Calculation methods for wire RC delay, taking wire resistances and branching points of the wires into account, have already been proposed, *e.g.* Elmore delay calculation([10]). In comparison with SPICE simulation, Elmore delay calculation has 30% error in the worst case, but is much faster. Since the logic synthesis system must calculate the delay many times during optimization, we have adopted the Elmore delay calculation method in this paper. We will consider the model which has a source gate connected to sink gates. Let Vt_j be the input threshold voltage, Vdd be the supply voltage. The wire RC delay between source n_i and sink n_j is calculated by equation (3), which is called Elmore delay calculation equation.

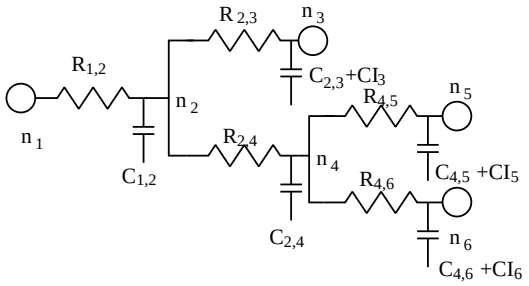


Figure 1: Elmore delay calculation example

$$\begin{aligned} tpd_{RCi,j} &= \beta_j TR_{i,j} && (3)\\ \beta_j &= log(\frac{Vdd}{Vdd - Vt_j}) && (rise)\\ &= log(\frac{Vdd}{Vt_j}) && (fall) \quad (4)\\ TR_{i,j} &= \sum_{S_{k,l} \in P(i,j)} R_{k,l}(\frac{1}{2}C_{k,l} + Cf_{k,l}) && (5)\\ Cf_{k,l} &= \sum_{S_{o,p} \in TFO(k,l)} (C_{o,p} + CI_p) && (6) \end{aligned}$$

$$\begin{aligned} P(i,j) &= set\ of\ segments\ between\ n_i\ and\ n_j\\ TFO(k,l) &= transitive\ fanout\ segments\ of\ S_{k,l} \end{aligned}$$

The capacitance and resistance for each segment of the wire cannot be estimated exactly with only the target technology library, and is strongly dependent on the topology of the wire. Therefore, the capacitance and resistance should be back-annotated to the logic synthesis system. In the case of the circuit in Figure 1, tpd_{RC} at the terminal n_5 is derived from equations (7) through (9).

$$\begin{aligned} tpd_{RC(rise)1,5} &= log(\frac{Vdd}{Vdd - Vt_5})TR_{1,5} && (7)\\ tpd_{RC(fall)1,5} &= log(\frac{Vdd}{Vt_5})TR_{1,5} && (8)\\ TR_{1,5} &= R_{1,2}(\frac{C_{1,2}}{2} + Cf_{1,2}) + \\ &\quad R_{2,4}(\frac{C_{2,4}}{2} + Cf_{2,4}) + \\ &\quad R_{4,5}(\frac{C_{4,5}}{2} + Cf_{4,5}) && (9) \end{aligned}$$

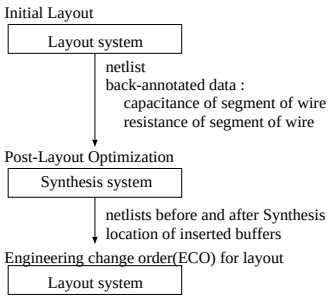


Figure 2: Design flow

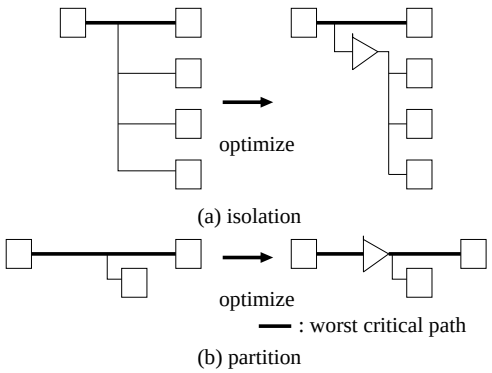


Figure 3: Buffer insertion

```
buffer-insertion()
/* step 1 */
while (critical paths exist & no area constraints violations &
       candidate wires on the critical paths exist) {
  foreach(wire on the worst critical path) {
    /* step 2 */
    foreach(segment n_i-n_j (S_i,j) of the wire) {
      pre_slack ← worst slack of the current fanout wire;
      post_slack ← -∞;
      /* step 3 */
      BUF ← the best buffer inserted on S_i,j
      /* step 4 */
      α ← best location of the buffer BUF;
      slack(BUF, α) ← worst slack when BUF is inserted;
      if (post_slack < slack(BUF, α)) {
        post_location ← α;
        post_slack ← slack(BUF, α);
        post_buffer ← BUF;
      }
      if (pre_slack < post_slack) {
        insert buffer(post_buffer) at post_location;
        timing analysis;
      }
    }
  }
}
```

Figure 4: Buffer insertion algorithm considering wire RC delay

4 Post-Layout optimization considering wire RC delay

The design flow with post-layout optimization considering wire RC delay (Figure 2) is the same as the current ASIC design using back-annotation. In this section we'll present the procedure for each step of the design flow.

4.1 Back-annotation

When the delay constraints specified by designers are not satisfied after placement and routing, delay optimization must be applied to the circuit using the logic synthesis system. The capacitance and resistance for each wire segment is extracted precisely from physical layout patterns. They are then back-annotated to the logic synthesis system.

4.2 Post-Layout optimization

Buffer insertion and gate sizing are applied in post-layout optimization. The wire RC delay is taken into account in both techniques.

4.2.1 Buffer insertion

Two types of buffer insertions cause delay reduction. They are isolation(Figure 3(a)) and partition(Figure 3(b)) of the worst critical path.

The entire algorithm is shown in Figure 4. Let n_1 be an output pin of the source gate, n_3, n_5 and n_6 be input pins of the sink gates and n_2 and n_4 be the branching points of the wire in Figure 5. The optimization steps are summarized as follows.

Step.1 selects a wire on the worst critical path.

A wire on the worst critical path is selected. A worst critical path is defined as a path with the minimum slack in the circuit. We continue the following steps until no more critical paths exist, the area constraints are violated or all candidate wires are checked. The delay of critical paths is re-calculated after each buffer insertion.

Step.2 selects a segment of the wire selected in step 1.

We continue the following steps until all segments are checked. Say that a segment between n_2 and n_3 is selected.

Step.3 selects the type of inserted buffer.

A candidate buffer type BUF is selected and the improvement of the delay is checked. The buffer types which cause fanout violation are removed from the candidates.

Step.4 searches the best location of the segment to insert buffers.

Let $AT(m, \alpha)$ and $RT(m, \alpha)$ be the arrival and required time at the point of n_m respectively when a buffer is inserted at the location of $\alpha : 1-\alpha$ on wire segment $S_{i,j}$. The resistance and capacitance of the wire are calculated by equations (10) and (11).

$$R = R0 \cdot L/W \tag{10}$$

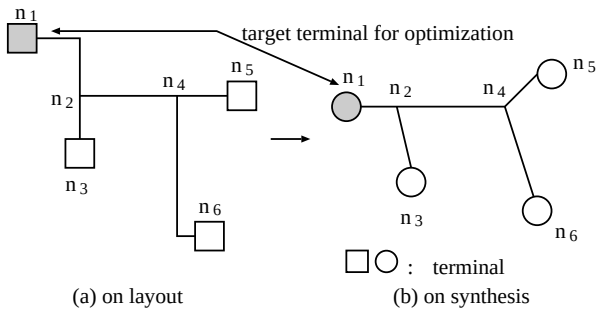


Figure 5: Buffer insertion example(before optimization)

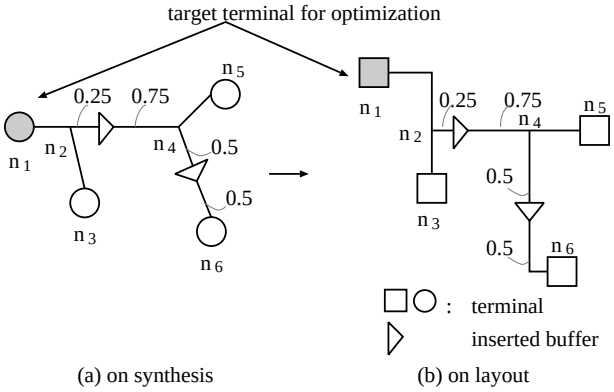


Figure 6: Buffer insertion example(after optimization)

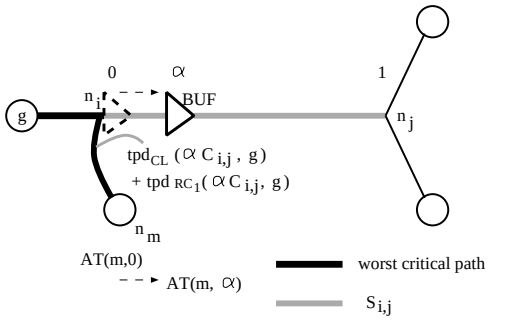
$$C = C0 \cdot L \cdot W \tag{11}$$

where $R0$ and $C0$ are sheet resistance and capacitance per unit area respectively, which are defined in the technology library, L is the length of the segment and W is the width of the segment. It is supposed that the resistance and capacitance between n_i and the inserted buffer are $\alpha R_{i,j}$ and $\alpha C_{i,j}$, the resistance and capacitance between the inserted buffer and n_j are $(1-\alpha)R_{i,j}$ and $(1-\alpha)C_{i,j}$ respectively when the buffer is inserted at the location of $\alpha : 1-\alpha$ $(0 \leq \alpha \leq 1)$ of the segment $S_{i,j}$. This assumption holds when the segment doesn't have vias and is laid out on one layer. But, when the segment has vias and is laid out on some layers, R0, C0 and W are different on some layers, so this assumption isn't always true. In that case, we should regard the wire between vias as a segment.

When the buffer is inserted at the location of α:$1-\alpha$ of the segment $S_{i,j}$, $AT(m,\alpha)$ is calculated as follows.

When the input pin n_m is connected to the fanin wire of the buffer, $AT(m,\alpha)$ is the summation of $AT(m,0)$, load delay calculated from the increasing capacitance $\alpha C_{i,j}$ and the wire RC delay calculated from the increasing capacitance $\alpha C_{i,j}$ of the segment $S_{i,j}$.

When the input pin n_m is connected to the fanout wire of the buffer, $AT(m,\alpha)$ is the summation of $AT(m,0)$, increasing wire delay on the fanout wire of the gate g, decreasing wire delay on the fanout wire of the gate BUF. The increasing wire delay is the summation of the load delay calculated from the increasing capacitance $\alpha C_{i,j}$



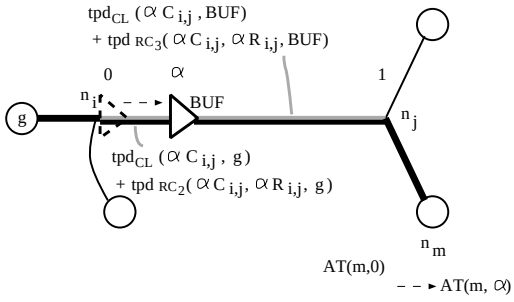


Figure 7: Arrival time calculation

and the wire RC delay calculated from the increasing capacitance $\alpha C_{i,j}$ and resistance $\alpha R_{i,j}$ of the segment $S_{i,j}$. The decreasing wire delay is the summation of the load delay calculated from the decreasing capacitance $\alpha C_{i,j}$ and the wire RC delay calculated from the decreasing capacitance $\alpha C_{i,j}$ and resistance $\alpha R_{i,j}$ of the segment $S_{i,j}$.

From the above consideration and equations (1) through (6), the arrival time $AT(m,\alpha)$ is calculated as follows :

- input pin n_m connected to the fanin wire of segment $S_{i,j}$ (Figure 7(a))

$$\begin{aligned} AT(m,\alpha) &= AT(m,0) + tpd_{CL}(\alpha C_{i,j}, g) + \\ & \quad tpd_{RC_1}(\alpha C_{i,j}, g) \end{aligned} \tag{12}$$

- input pin n_m connected to the fanout wire of segment $S_{i,j}$ (Figure 7(b))

$$\begin{aligned} & AT(m,\alpha) = AT(m,0) + \\ & (tpd_{CL}(\alpha C_{i,j}, g) + tpd_{RC_2}(\alpha C_{i,j}, \alpha R_{i,j}, g)) + \\ & (-tpd_{CL}(\alpha C_{i,j}, BUF) - tpd_{RC_3}(\alpha C_{i,j}, \alpha R_{i,j}, BUF)) \end{aligned} \tag{13}$$

where $tpd_{CL}(C,g)$ is the load delay of the gate g which has the capacitance of the load C. $tpd_{RC_1}(C,g)$ is the increasing wire RC delay, which is the function of increasing capacitance C, at the input pin of the gates except for BUF on the fanout wire of gate g. $tpd_{RC_2}(C,R,g)$ is the increasing wire RC delay, which is the function of increasing capacitance C and resistance R, at the input pin of BUF on the fanout wire of gate g. $tpd_{RC_3}(C,R,BUF)$ is the decreasing wire RC delay, which is the function

of decreasing capacitance C and resistance R, at the input pin of gate on the fanout wire of BUF. The slack $ST(m,\alpha)$ for each pin n_m, when the buffer is inserted at the location α, is calculated by the following equation.

$$ST(m,\alpha) = RT(m,\alpha) - AT(m,\alpha) \quad (14)$$

The worst slack after the buffer insertion is calculated by equation (15).

$$slack(\alpha) = min_{n_m \forall terminals}(ST(m,\alpha)) \quad (15)$$

There are two methods to calculate the location of the inserted buffer from the above equations (12) through (15).

Method 1: The worst slack is calculated from equation (15) when the buffers are inserted at constant intervals on the target segment. The location α where the worst slack is most improved is maximum is selected. For example, if the segment is partitioned into four parts of the same length, the locations α=0,0.25,0.5,0.75,1.0 are checked one by one.

Method 2: The location α can be directly calculated in order to improve the worst slack. There are two cases when inserting buffers as shown in Figure 3.

One case is when the worst critical path is isolated by the inserted buffer(Figure 3 (a)). Note that the equation (14) is a monotonous increasing function of α which is derived from equations (1) through (6) and (12). It means that the worst slack is improved when α is equal to 0.

The other case is when the worst critical path is partitioned in the middle of the wire by the inserted buffer(Figure 3 (b)). Note that the equation (14) is a quadratic function of α which is derived from equations (1) through (6) and (13). Then the partial differential equation (16) of α is solved. Because the coefficient value of α^2 on equation (13) is positive, the worst slack is most improved when the equation (16) is equal to 0. Finally, the location α is calculated by equation (16).

$$\frac{\partial ST(m,\alpha)}{\partial \alpha} = -\frac{\partial AT(m,\alpha)}{\partial \alpha} \rightarrow 0 \quad (16)$$

If α is less than 0, then let α be 0 and if α is larger than 1, then let α be 1.

If the worst slack after inserting the buffer at the calculated location α is better than the previous one (*i.e.* the worst slack is improved by the buffer insertion), the buffer insertion is really executed and the circuit is changed.

Figure 6 (a) shows an example of how to calculate the exact delay after the buffer insertion. The resistance and capacitance between n_2 and the buffer are $0.25R_{2,4}$ and $0.25C_{2,4}$ respectively, and the resistance and capacitance between the buffer and n_4 are $0.75R_{2,4}$ and $0.75C_{2,4}$ respectively from equations (10) and (11) when the buffer type BUF is inserted at the location of $\alpha = 0.25$ on the segment $S_{2,4}$

Therefore the arrival time $AT(3,0.25)$ at location n_3 is calculated by equation (12). Similarly, $AT(5,0.25)$ at location n_5 and $AT(6,0.25)$ at location n_6 are calculated respectively by equation (13).

The computation time order of our buffer insertion algorithm is $O(S)$, where S is the number of processed segments.

4.2.2 Gate sizing

Another delay optimization algorithm is gate sizing. It replaces some gates on the critical paths with higher power gates which are equivalent in functionality so as to improve worst slack. Furthermore, it reduces area by replacing some gates on the non-critical path with lower power gates which usually have less area and are equivalent in functionality. Typical gate sizing algorithm([7]) has been applied . Gate sizing doesn't affect the wire RC delay.

4.3 Engineering change order for layout

After optimizing delay by logic synthesis, only the changed parts of the circuit are placed and routed again using the engineering change order (ECO) technique.

Note that each replaced gate must be placed at the closest location of the previous one, and the routing pattern for each new wire must be almost the same as the previous one. Layout system also should refer to and preserve the location of inserted buffers as much as possible.

Figure 6 (a) shows an example of the circuit after the delay optimization of Figure 5 (b) in the logic synthesis system. Figure 6 (b) shows the results of ECO which can realize the same locations of the inserted buffers as the estimated ones in a logic synthesis system.

It is assumed that our approach places buffers on routed wires and that after inserting buffers the topologies of wires vary little. If the inserted buffers are not placed on the wires, the topologies of the wires change significantly. When the topologies of the wires change, it is more difficult to estimate the locations of branching points of wires, and the delay after optimizing can not be estimated accurately. The logic synthesis system can estimate the delay accurately during optimizations only if the layout system can re-route to preserve the same topologies of wires after placing buffers on the routed wires.

5 Experimental results

The experiments should be done in the deep submicron technology for validating our approach.

Three experimental data are created from the real 0.5 μm rule gate array designs to meet intrinsic and wire RC delay ratio of 0.35 μm rule. The size of three data are summarized in Table 2. The first column shows the number of gates and the second column shows the utilization ratio of gates in the chip.

The maximum delay improvement, area, number of inserted buffers, CPU time are shown in Table 3. BO, ARGL, ABIS show the maximum delay on the worst critical path and area before post-layout optimization, after gate sizing, after buffer insertion respectively. The values in parentheses are the ratio of the maximum delay and area normalized by those of BO.

At first, we back-annotate wire capacitance and wire RC delay of the initial laid out circuit from the layout system to the logic synthesis system. This initial circuit is laid out as usual. The circuit is then optimized by gate sizing. Next, we re-layout the optimized circuit. The re-laid out circuit and wire RC data are back-annotated from the layout system to the synthesis system again. The circuit is then optimized by buffer insertion considering wire RC delay.

Table 2: Experimental Data

data	# of gates	utilization [%]
TEST1	4989	11.3
TEST2	19623	48.7
TEST3	36427	69.5

Table 3: Experimental Result

data		TEST1	TEST2	TEST3
maximum delay [ns]	BO	22.96	52.31	61.33
	ARGL	22.96(1.000)	37.93(0.725)	54.55(0.889)
	ABIS	21.07(0.918)	30.78(0.588)	50.89(0.823)
area [# of gates]	BO	4989	19623	36427
	ARGL	4989(1.000)	19697(1.004)	36501(1.002)
	ABIS	5017(1.006)	19808(1.009)	36685(1.007)
# of inserted buffers		16	41	97
CPU time [sec.]		85	662	1843

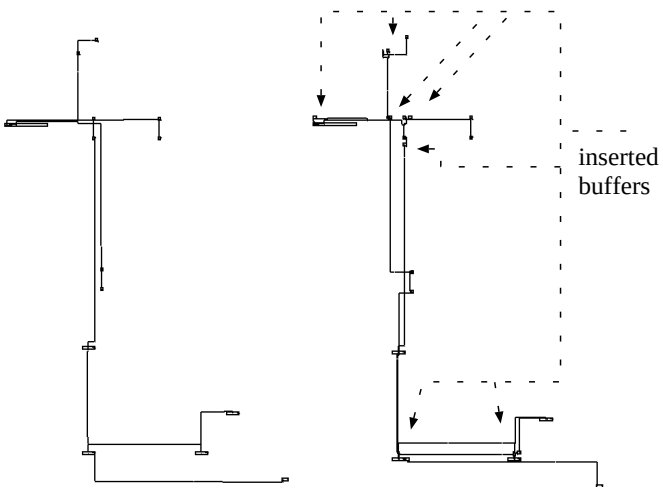


Figure 8: Optimized wire before(left)/after(right) buffer insertion

In this experiment, we use the location decision rule of *Method 1* which checks the slack of equally partitioned segments of the wire.

From Table 3, the average and maximum improvements in maximum delay (ARGL) are 10.4% and 27.5% respectively. The average and maximum improvements in maximum delay (ABIS) are 18.6% and 41.2% respectively.

To check the delay accuracy of ABIS, we manually placed buffers and routed automatically for TEST1 data. The maximum delay after ECO is 21.08 ns. The delay difference before and after ECO is 0.01ns. Therefore, we see our method estimates delay accurately at the synthesis phase. We show the worst critical path on the layout before and after buffer insertion in Figure 8. These wire patterns are almost the same.

The total CPU times for buffer insertion are shown in Table 3. Those include the time for timing analysis (about 60% - 90%).

6 Conclusion

In this paper, we proposed a buffer insertion technique considering wire RC delay. In our approach, the resistance and capacitance of the segments of wires are back-annotated from the layout system to the synthesis system and the buffers are inserted at the proper location on the routed wires. New wires are routed almost the same as the previous ones. Consequently the estimated maximum delay is extremely accurate and the error rate between estimated and real routed delay is very low. The combination of our buffer insertion and the conventional gate sizing algorithm has resulted in the reduction of the maximum delay up to 41.2 %.

There are some possibilities to improve our approach *e.g.* using two inverters instead of a buffer or inserting a series of buffers at the same time.

Acknowledgements

Authors would like to thank the General Manager A.Morino, Senior Manager Y.Fujinami and co-workers for several discussions about the post-layout optimization.

References

[1] S.Lin, M.M-Sadowska and E.S.Kuh. "Delay and Area Optimization in Standard-Cell Design". In *Proc. of the 27th Design Automation Conf.*, pages 349–352, 1990.

[2] K.J.Singh and A.Sangiovanni-Vincentelli. "A Heuristic Algorithm for the Fanout Problem". In *Proc. of the 27th Design Automation Conf.*, pages 357–360, 1990.

[3] K.Yoshikawa, H.Ichiryuu *et al.* "Timing Optimization on Mapped Circuits". In *Proc. of the 28th Design Automation Conf.*, pages 112–117, 1991.

[4] M.Pedram and N.Bhat. "Layout Driven Technology Mapping". In *Proc. of the 28th Design Automation Conf.*, pages 99–105, 1991.

[5] M.Pedram and N.Bhat. "Layout Driven Logic Restructuring/Decomposition". In *Proc. of the Intl. Conf. on Computer-Aided Design*, pages 134–137, 1991.

[6] H.Vaishnav and M.Pedram. "Routability-Driven Fanout Optimization". In *Proc. of the 30th Design Automation Conf.*, pages 230–235, 1993.

[7] L.N.Kannan, P.R.Suaris and H-G.Fang. "A Methodology and Algorithms for Post-Placement Delay Optimization". In *Proc. of the 31st Design Automation Conf.*, pages 327–332, 1994.

[8] T.Aoki, M.Murakata, T.Mitsuhashi and N.Goto. "Fanout-tree Restructuring Algorithm for Post-placement Timing Optimization". In *Proc. of the ASP-DAC'95 / CHDL'95 / VLSI'95*, pages 417–422, 1995.

[9] J.L.Lillis, C.K.Cheng and T.T.Y.Lin. "Optimal and Efficient Buffer Insertion and Wire Sizing". In *Proc. of the Custom Integrated Circuits Conf.*, pages 259–262, 1995.

[10] W.C.Elmore. "The transient response of damped linear networks with particular regard to wide-band amplifiers". In *Journal of Applied Physicals*, pages 55–63, 1948.