# Incremental Gate Sizing for Late Process Changes

John Lee and Puneet Gupta
*Electrical Engineering Department, UCLA*
*lee@ee.ucla.edu, puneet@ee.ucla.edu*

*Abstract*— **Circuit design often runs in parallel with the development of the manufacturing process that will be used to fabricate it. However, as the manufacturing process matures, its models may undergo substantial changes as the design nears production. These changes may cause the design itself to fail its specifications, and in these cases it is necessary to perform an Engineering Change Order (ECO) to correct these problems. We present a new framework to perform incremental gate sizing for process changes late in the design cycle. This includes a method to measure and estimate ECO cost, transform these costs into a linear programming optimization problem, and solve the problem to find the ECO. This method performs well, compared to a leading commercial physical design tool, reducing ECO costs by 18% to 99% in changed area, and 1% to 96% in number of pins with unnecessary pin timing changes.**

## I. Introduction

With the aggressive production schedules in the semiconductor industry, the design of integrated circuits runs concurrently with the development of the manufacturing process itself. This means that the exact manufacturing specifications may not be known when the design team is working on the design. As the design progresses, additional information may become available from the manufacturer. However, substantial changes in the specification would require Engineering Change Orders, commonly referred to as ECOs, on the original design.

The extent of the ECO depends on when the updated information arrives in the product's development cycle. If the information arrives before substantial engineering time is spent, the product may simply be redesigned. In contrast, if significant time has been spent validating the design, it would be wise to employ an ECO that affects a minimum fraction of the design.

In this paper, we focus on late-design cycle ECOs when the changes arrive after the the design has been placed and routed, but before it is sent for fabrication. These changes may also correspond to a different but design-rule compatible process to which the design is being retargeted. We assume that the power of the designs is to be minimized, subject to a timing constraint. Once the new process information is introduced, we would like to minimize the impact of the ECO while maintaining a solution that is reasonably optimal.This is done by first quantifying the ECO cost in terms of the area cost and the timing cost, and then approximating this relation as a function of layout parameters. The resulting model is fed into an optimization loop which minimizes a trade-off of ECO cost and power while meeting the timing constraint.

The contributions of this paper are:

1) measures to quantify ECO cost in terms of timing and area change;
2) a new algorithm to perform discrete sizing with ECO cost estimates using linear programming;
3) comparisons with a commercial physical design tool that illustrates the superiority of the proposed approach.

This paper focuses on the gate sizing problem as it is one of the most flexible and widely used methods available. It is less intrusive than adjusting the placement of the design, and more powerful than rerouting the design. It does not change the logic functionality of the circuit, and works by changing the sizes of the gate to increase the drive strength or to decrease the power.

### A. Background

Research on incremental algorithms has been well studied (see [1]). Problems with existing incremental algorithms are discussed in [2], [3].

Significant progress has been made in the area of incremental and ECO routing [4], placement (see [5]), synthesis mapping [6], and spare cell usage [7]. However, as far as the authors know, the subject of ECO gate sizing has received no attention, especially in the context of minimizing the impact of the ECO.

The general topic of gate sizing itself is a very well studied topic. These approaches can be categorized by continuous sizing approaches (see, for example [8]) and discrete approaches (see for example [9], [10]).

Linear Programming, in the context of gate sizing, has also been well studied [11], [9], [12]. This paper follows a similar formulation as in [12], applied to the case of ECO minimization.

### B. How specifications can change

With the ever shrinking dimensions and complex process control mechanisms, the change in the specifications can be substantial. For example, Figure 1 shows the percentage change from April 2008 to March 2010, for a commercial 45nm process. The difference in these parameters is not negligible – the transistor off current ($I_{\text{off}}$) increases by over 80%, and the gate capacitance increases by approximately 10%. These two changes alone would have a large impact, by increasing the leakage power by over 80%, the dynamic power by approximately 10%, and the delay by approximately 10%.

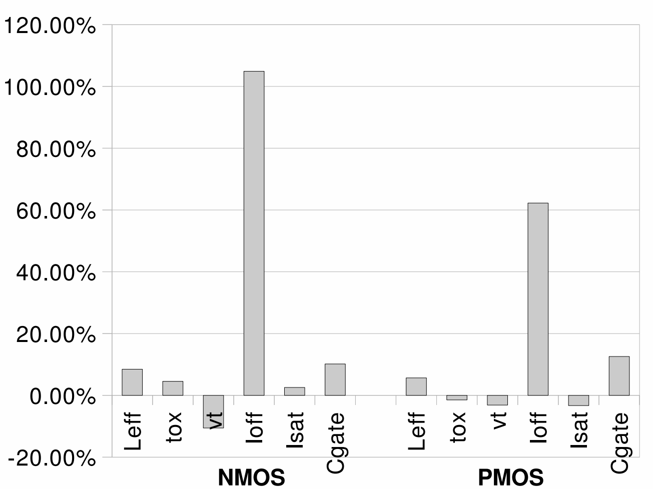


Fig. 1. Comparison of the 2008 and 2010 process specifications for a commercial 45nm process. The graph plots the percentage increase or decrease for several key parameters.

Furthermore, these changes differ for PMOS and NMOS – while the $t_{\text{ox}_e}$ increases for NMOS, it decreases for PMOS, and similarly for the $I_{\text{sat}}$. However, the current $I_{\text{off}}$, along with $L_{\text{eff}}$ and $C_{\text{gate}}$, increases for both transistors, indicating that the process will have a higher leakage power, and may be slower than originally predicted.

Changes can be even more widespread for designs using a variety of different device types. Though we do not explore them here, changes can be layout-dependent as well.

With this uncertainty in manufacturing specifications, it becomes important to research algorithms that adjust designs to account for these changes. It is important to have a method that can modify the design multiple times, for each new set of specifications, with a minimal cost.

## II. ECO Cost

Research on ECO and incremental algorithms has focused on traditional costs – wire-length, timing closure, and the number of changed nets (see for example [7], [13], [14]). These metrics work well when the timing closure and the traditional metrics are important, but they are too general to describe cases where there are many possible timing closure solutions, and it is important to weigh the cost needed to implement the design.

In practice, the ECO cost is determined by the amount of time, in engineering work time and in tool hours, that is required to perform the ECO. This time is the spent in:

1) Checking and correcting the timing: how much of the design must be rechecked for timing validity, and how much time is need to fix any detected errors? Note that in modern system-on-a-chip (SoC) designs, a large fraction of this verification may be manual.
2) Checking and correcting the layout: is the resulting layout manufacturable with high yield?
3) Checking and correcting design rules: are there violations in the electrical or layout characteristics (maximum capacitance, slew, wire density)

Thus, it is important to find a measure of ECO, ECO$(\cdot)$, that correlates to the costs in (1)-(3). We approximate the costs using two measures:

1) $c_{\text{area}}$: The area change from the ECO: the amount of layout area that changes. This includes area that is changed by cell changes, cell movement, and routing changes. This area is computed over all layers of the design.
2) $c_{\text{timing}}$: The number of non-critical pins that are in the fan-in or fan-out cones of the ECO-changed cells. This is used to measure the ECO cost related to unintended timing changes.

The area cost relates to the area that needs to be re-validated for design rule violations, the amount of re-routing that needs to be done, and the amount of parasitic information that needs to be recomputed. The timing cost is a measure of how the ECO affects gates downstream and upstream from it, which will need to be re-timed to ensure that the delay specifications are met. Minimizing unintended timing changes becomes important especially when the place and route timer is different from the sign-off timer. This is also important when there are precise arrival time constraints at the block or design boundaries.

Note that the area cost is also related to the timing cost. The area changes that are captured in $c_{\text{area}}$ also relate to the amount of parasitics that need re-extraction and the amount of routes that need to be rewired. Furthermore, these two measures are complimentary – while the $c_{\text{area}}$ measures the local area change associated with an ECO change, the $c_{\text{timing}}$ measures the effect of the ECO on the gates that may be distant from the change, but affected through the topology of the circuit.

The ECO cost is a function of the circuit layout, the interconnect routing, and the type of change that is needed. The timing ECO cost ($c_{\text{timing}}$) can be predicted by counting the number of non-critical pins in the fan-out and fan-in cones of the changes. In contrast, the area ECO cost is difficult to quantify without performing the ECO itself. This cost is the result of a chaotic interaction between the incremental design tool and the current layout.

For the purposes of guiding the optimization we construct and estimated ECO area cost ($\hat{c}_{\text{area}}$) from the following information about a potential change:

- $m_1$: Number of affected pins
- $m_2$: Number of dislocated pins (old locations and new locations do not overlap)
- $m_3$: Pin bounding box area
- $m_4$: Utilized area over pin bounding box (routing over all layers)

This information is obtained by using a quick legalization-like placement check that finds the amount that cells must be moved to find free space for the potential ECO. These parameters are used in a linear model for $\hat{c}_{\text{area}}$:

$$\hat{c}_{\text{area}} = \sum_{i=1}^{4} a_i m_i + b. \tag{1}$$

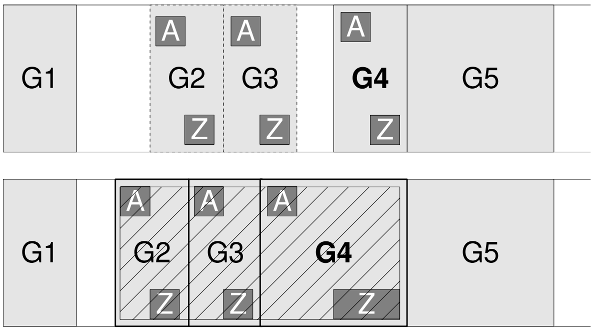


Fig. 2. ECO example to estimate $c_{\text{area}}$. Gate G4 changes from INV size 1 to INV size 2, dislocating cells G2 and G3. All the pins are affected by the change ($m_1 = 6$), but $m_2 = 5$ because pin G4/Z still overlaps with its old location. The cross-hatched area is the pin bounding box area $m_3$.

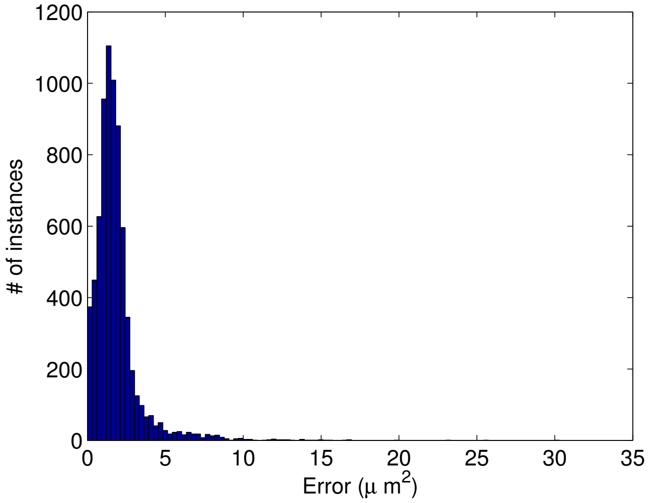


Fig. 3. Error histogram of the difference between the estimated ECO area values ($\hat{c}_{\text{area}}$) and the actual ECO area values ($c_{\text{area}}$) for 7274 data points over the ISCAS '85 benchmarks.

A sample of 7274 ECO operations over the ISCAS '85 benchmarks is used to fit the model, and a least-squares fit of the coefficients $a_i$ is made. The values are shown in Table I. The quality of the fit is shown in Figure 3, which we show later to be good enough for solving the ECO problem.

We can use this information to estimate the cost of changing the size of a given cell. For example, consider the case in Figure 2. A quick placement check is done to find the values of $m_1$ to $m_3$, and routing congestion information is used to get $m_4$. With the values $m_1 = 6$ and $m_2 = 5$ (and assuming $m_3 = 0.25$ and $m_4 = .05$) the expression gives the estimate of $3.2\mu\text{m}^2$.

These estimates can be used to help to guide the ECO process. Gates in congested areas will result in large estimated ECO costs, as changing the gate will move many neighboring cells (resulting in large values for $m_1$ - $m_3$), and require rerouting in a congested area ($m_4$). Relying on changes with small ECO cost will help to make changes where free space is high and congestion is low.

TABLE I

ECO Cost Fitting Coefficients

| | value |
|---|---|
| $a_1$ | 0.0367 $\mu\text{m}^2/\text{pin}$ |
| $a_2$ | 0.186 $\mu\text{m}^2/\text{pin}$ |
| $a_3$ | 5.35 |
| $a_4$ | 9.65 |
| $b$ | .264 $\mu\text{m}^2$ |

## III. Solving the redesign problem

Suppose we would like to solve the incremental problem: *given the current set of sizes x, find a suitable adjustment y which is the solution to:*

$$\begin{array}{ll} \text{minimize} & \text{Power}(y) + \gamma\text{ECO}(y;x) \\ \text{subject to} & \text{Delay}(y) \le T_{\max} \end{array} \tag{2}$$

This is the fundamental ECO problem – how can the power and ECO costs be juggled to meet the timing constraint? The term $\text{ECO}(y;x)$ measures the amount of change or the difference in the designs $x$ and $y$, in terms of an ECO cost. As $\gamma$ becomes large, this cost becomes more significant.

To solve this problem, we consider the case where we would like to minimize the ECO modifications, and consider the following *simplifying assumptions*:

1) The ECO costs are additive (the total ECO is the sum of the costs of the individual ECOs)
2) Out of every two connected gates, at most one gate should change its size (see Section III-B)

This results in the following linear programming approximation to (2):

$$\begin{array}{ll} \text{minimize} & \sum_{i,k} p_{ik} y_{ik} + \gamma\text{ECO}(y;x) \\ \text{subject to} & t_i + d_{i0} + \sum_k \delta_{ik} y_{ik} \le t_j, \quad \forall i \in \text{fo}(j) \\ & t_i \le T_{\max}, \quad \forall i \in \text{po} \\ & \sum_k y_{ik} \le 1, \quad \forall i \\ & \sum_k y_{ik} + \\ & \cdots \sum_{j\in\text{fo}(i)} \sum_k y_{jk} + \sum_{j\in\text{fi}(i)} \sum_k y_{jk} \le 1, \quad \forall i \\ & 0 \le y_{ik} \le 1 \end{array} \tag{3}$$

The variables are:

- $t_i$: Arrival time for gate $i$
- $d_{i0}$: Current delay for gate $i$
- $\delta_{ik}$: Change in the delay of gate $i$ under size $k$
- $y_{ik}$: Assignment variable of gate $i$ to size $k$
- $p_{ik}$: Power cost of changing gate $i$ to size $k$

We call this algorithm LPECO. This algorithm finds an assignment of sizes to gates that minimizes a weighted objective of power and ECO cost. The variables $t_i$, $d_{i0}$ and $\delta_{ik}$ are related to the timing of the design, and they propagate the arrival times down the graph, to enforce setup time constraints, as in [12].[1] The variable $y_{ik}$ is the assignment variable that is 1 when gate $i$ is size $k$ in the solution, and 0

[1] This formulation can also consider hold time constraints. In this case, we add a second set of timing variables, where the earliest arrival time of a gate is set to be less than the earliest arrival times of the fan-in gates, plus the minimum gate delay.

otherwise. Note that for a given $i$, if all $y_{ik} = 0$, this indicates that the current gate size is kept, and not changed.

As the number of possible moves is very large, we restrict the search to the gates that have negative slack, and the moves that improve slack (e.g. $\delta_{ik} < 0$). This means that the size of the problem is dominated by the number of possible moves, and not the size of the circuit. Furthermore, to consider the effect of fan-out load, gates are also considered if they are a fan-out of a critical gate. Fan-ins can also be considered but we ignore them in our current experiments as they have little effect on delay for our benchmarks.

Note that design rules such as max transition and max capacitance can be handled in this formulation. This can be done by removing the assignments that violate these rules.

The constraint $\sum_k y_{ik} \leq 1$ prevents the assignments of gate $i$ from adding up to more than 1. The constraint

$$\sum_k y_{ik} + \sum_{j \in \mathrm{fo}(i)} \sum_k y_{jk} + \sum_{j \in \mathrm{fi}(i)} \sum_k y_{jk} \leq 1 \tag{4}$$

is used to help enforce assumption 1, that only one gate out of every two connected gates will change size. However, this does not guarantee that only one gate will be assigned, and we will consider these indeterminate cases in Section III-D.

If we assume that the delay models are sufficiently accurate (to some minimum tolerance), the solution will give a lower bound on the optimal assignment. Namely, if $y^\star$ is the optimal solution, no assignment can produce a solution better than:

$$\sum_{i,k} p_{ik} y^\star_{ik} + \gamma \mathrm{ECO}(y^\star; x) \tag{5}$$

This algorithm, however, is not guaranteed to return a solution where all $y_{ik} \in \{0,1\}$, and some extra assignment must be done (see Section III-D). This assignment process introduces suboptimality into the process; however, there are ways to mitigate this (see Section III-D).

### *A. Incorporating ECO costs*

Introducing $c_{\mathrm{area}}$ into the optimization problem 3 is straightforward, as the model in Section II can be used to estimate the area cost of moving gate $i$ to size $k$ ($e_{ik}$). This gives the expression

$$c_{\mathrm{area}} = \sum_{\forall i,k} e_{ik} y_{ik} \tag{6}$$

which can be added to the objective of (3).

Incorporating the timing cost is more involved, and requires the introduction of extra constraints in addition to an extra term in the objective. The extra complexity stems from the need to measure the fan-out and fan-in contributions, while avoiding double-counting. For example, if gate $k$ is in the fan-out cone of gates $i$ and gate $j$, the timing cost of $k$ should only be counted once even if both of gates $i$ and $j$ change. This is done using the variables:

- $\tau_i^{\mathrm{fo}} \in [0,1]$, which indicates that gate $i$ is in the fan-out cone of a gate that changes size,
- $\tau_i^{\mathrm{fi}} \in [0,1]$, which indicates that gate $i$ is in the fan-in cone of a gate that changes size, and
- $\tau_i \in [0,1]$, which indicates that gate $i$ is in the fan-in cone or fan-in cone of a gate that changes size.

These variables are set using the following constraints:

$$\begin{array}{ll} \sum_{\forall k} y_{ik} \leq \tau_i^{\mathrm{fi}}, \ \ \forall i & \sum_{\forall k} y_{ik} \leq \tau_i^{\mathrm{fo}}, \ \ \forall i \\ \tau_i^{\mathrm{fo}} \leq \tau_j^{\mathrm{fo}}, \ \ \forall i \in \mathrm{fo}(j) & \tau_k^{\mathrm{fi}} \leq \tau_j^{\mathrm{fi}}, \ \ \forall j \in \mathrm{fi}(k) \\ \tau_i^{\mathrm{fi}} \leq \tau_i, \ \ \forall i & \tau_i^{\mathrm{fo}} \leq \tau_i, \ \ \forall i. \end{array} \tag{7}$$

The first two constraints involving $y_{ik}$ ensure that for any changed gate, the fan-out and fan-in indicators are marked. The second two constraints ensure that the timing fan-out and fan-in cones are propagated. The last two constraints ensure that $\tau_i$ is greater than $\tau_i^{\mathrm{fo}}$ and $\tau_j^{\mathrm{fi}}$, and thus marked to be counted in the ECO timing cost.

With these constraints, $\hat{c}_{\mathrm{timing}}$ can be approximated as:

$$\hat{c}_{\mathrm{timing}} = \sum_i r_i \tau_i \tag{8}$$

where $r_i$ is the number of *non-critical* pins on gate $i$. Non-critical pins are defined as pins that have positive slack under the new manufacturing specifications.

### *B. Restrictions on neighboring gates*

The assumption that “out of every two connected gates, at most one gate should change its size” is made because we assume that the ECO sizing changes are small changes over the entire circuit. In other words, because the ECO sizing depends on minimizing the ECO costs in area and timing, we may assume that connected gates are not likely to change.

This assumption is useful because it simplifies the gate size vs. delay relations. The delay of a gate $i$ is a function of the input slew, gate size and the output load. Thus, the size of gate $i$ alone does not determine the new delay; the change in the sizes of the output gates also affects the gate delay, and when neighboring gates change, the resulting delay is generally not equal to the sum of the individual changes. However, by introducing the restriction that neighboring gates do not change allows for the straightforward delay expressions in (3).

### *C. Maximizing slack*

Problem (3) may be infeasible when the amount of negative slack is large. In these cases, the slack must be maximized iteratively, until problem (3) becomes feasible. This leads to the following problem:

$$\begin{array}{ll} \text{minimize} & t_{\max} \\ \text{subject to} & t_i + d_{i0} + \sum_k \delta_{ik} y_{ik} \leq t_j, \quad \forall i \in \mathrm{fo}(j) \\ & t_i \leq t_{\max}, \quad \forall i \in \mathrm{po} \\ & \sum_k y_{ik} \leq 1, \quad \forall i \in \mathrm{fo}(j) \\ & 0 \leq y_{ik} \leq 1 \end{array} \tag{9}$$

In this paper, we iterate (9) until a timing feasible solution is found, with a maximum number of iterations set at 10.

### D. Indeterminate assignments

The solution to (3) or (9) may have indeterminate assignments, e.g. the $y_{ik}$ may be greater than 0, but less than 1. In these cases, a decision must be made as to whether a gate should be changed, and if so, which size it should be assigned to.

A guideline for the indeterminate assignments in the problem (3) can be derived from the lower bound equation (5). As this equation is linear, we can approximate the suboptimality for the case of $d_{ik} > 0$[2] as

$$\sum_{\forall y_{ik}>0} (p_{ik} + \gamma_{\text{area}} e_{ik})(1 - y^{\star}_{ik}) + \gamma_{\text{timing}} \sum r_i (1 - \tau_i). \qquad (10)$$

This suboptimality comes from the difference between the continuous and the integer solutions to the problem.

We can reduce this gap by considering other sizes that may reduce the suboptimality in (10). Although the term $r_i(1-\tau_i)$ is unavoidable for any gate assignment to gate $i$, the left hand size can be reduced by considering alternate assignments. Formally, if we are given an indeterminate assignment $y_{ik}$, the suboptimality is minimized over $k$ by choosing the size $s$ as:

$$s = \text{argmin}_{\{j|\ \delta_{ij} \geq y_{ik}\delta_{ik}\}} p_{ij} + \gamma_{\text{area}} e_{ij} \qquad (11)$$

For example, consider the case where $\gamma_{\text{area}} = 1$ and gate $i$ has assignment $y_{i4} = .5$. The available options $(p_{ik}, e_{ik}, \delta_{ik})$ are given by

- $p_{i2} = 1$, $e_{i2} = 2$, $\delta_{i2} = -1$ for $(k = 2)$
- $p_{i4} = 2$, $e_{i4} = 2$, $\delta_{i4} = -2$ for $(k = 4)$

The assignment $y_{ik} = .5$ indicates that a feasible option must have a slack of 1 $(= y_{ik} \cdot \delta_{i4})$ or greater, which is satisfied by $k = 2$. Utilizing $k = 2$, however, adds only 1 in (10), rather than the 2 that would be added by the original assignment of $k = 4$. Thus, using the alternate assignment of $k = 2$ will reduce the suboptimality gap by 1.

In the case of the slack maximization problem (9), we can also use (11), although there is no lower bound analysis available in this case. However, this can help the slack minimization algorithm choose solutions that have smaller power and ECO costs.

In a small minority of cases, the algorithm will assign neighboring gates. We circumvent this difficulty by using a greedy algorithm to apply the assignments to the gates in order of increasing sensitivity ($\Delta$objective/$\Delta$slack). In addition, only values of $y > 0.01$ are applied in (3), and values of $y > 0.1$ are applied in (9). If a gate has a neighbor that has already been mapped, the gate is skipped and left unmapped.

## IV. Experimental Results

This algorithm is tested on the ISCAS '85 benchmarks and the Open Cores ALU[15], which are synthesized to the Nangate 45nm Library[16], and optimized using a leading commercial design tool. Table II gives information about these benchmarks for the nominal process parameters. The library is then adjusted for the following parameter changes, using a different commercial tool:

- $v_t$: nmos -10%, pmos -5%
- $t_{ox}$: nmos +5%, pmos -5%
- $c_{\text{gate}}$: nmos +10%, pmos +10%
- $l_{\text{eff}}$: nmos +5%, pmos +5%

These changes are derived from a 2 year change in a commercial 45nm process, and these library changes create a negative slack, or timing violation, that is repaired using the algorithm LPECO in Section III and the commercial design tool in *post-route* mode and the optimization effort set to high. All timing data in this paper is generated using this commercial design tool. The flowchart for this experiment is shown in Figure 4.

$l_{\text{eff}}$ and $v_{\text{th}}$ assignment are not considered in this work, as our gate library does not support these options. However, these options would be easy to integrate into our framework, and for these modalities, only the $c_{\text{timing}}$ would be affected.

The algorithm LPECO is implemented using C++ and the linear programming solver in MOSEK [17]. The ECO cost estimates are also programmed in C++, and the final ECO design is created using the commercial design tool.

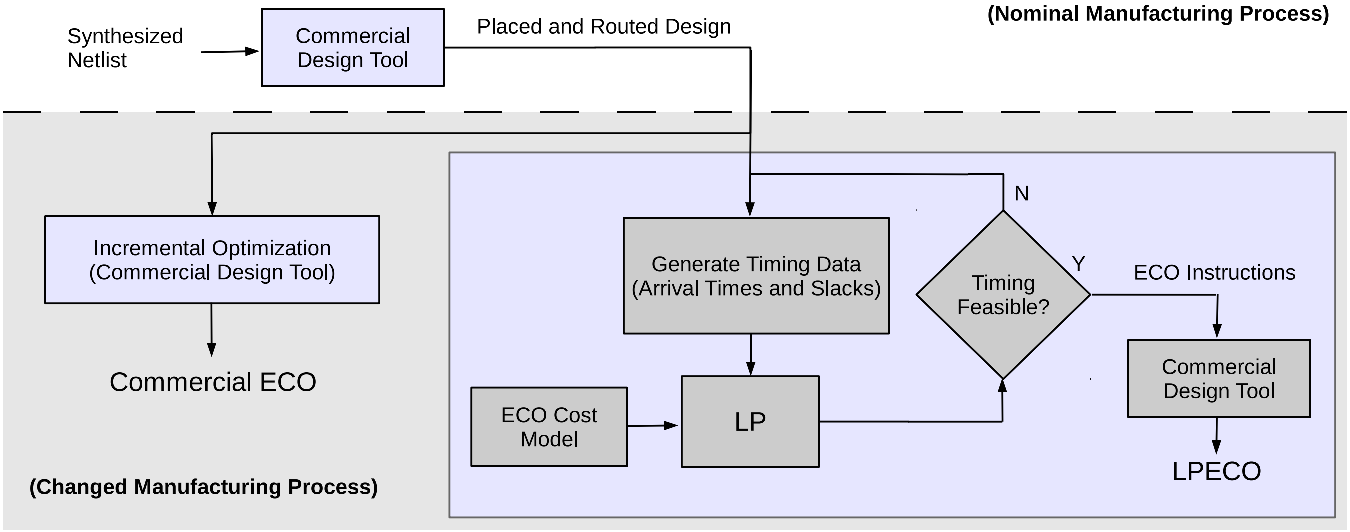
Results are shown in Table III for the three different congestion targets 70%, 80% and 90%. The $c_{\text{area}}$, $c_{\text{timing}}$ and $p_l$ represent the actual ECO area cost, ECO timing cost and leakage power, respectively. The slacks in the table are computed after the parameter change. In the cases where the a timing feasible design cannot be found, the algorithm LPECO reduces the negative slack better than the commercial design tool in all but two cases (the 70% alu, and the 80% c6288 benchmark). However, the ECO and power costs may suffer as the algorithm tries to close the timing gap.

In the cases where LPECO finds a timing feasible solution, it outperforms the commercial design tool in all metrics. The $c_{\text{area}}$ metric is much better – in the 90% congestion case, it is half of the commercial tool's value, on average. In the same set of benchmarks, the $c_{\text{timing}}$ is 12% less than the commercial tool, and the power is 5% less. This shows that the formulation in (3) is effective.

However, the effect on the power is small in these cases. The change in the power from the initial power ($p_{\text{initial}}$) is much smaller than the change in the ECO timing and area measures. This means that the ECO measures are important to gauge the quality of an ECO, as the resulting power change is near-negligible.

The designs with higher congestion fare better in the results for both the commercial tool and LPECO. This is counter-intuitive, as it would seem that a larger die area would give the tools more flexibility. However, in these designs, the congestion increases the potential impact for sizing. For example, in the case of the 70% congestion c7552 circuit, the delay ranges from .82ns to .96ns, a difference of .14ns. However, for the 80% congestion version, the range is from .84ns to 1.4ns. This is because the placement is much better with 70% congestion, and there is less of a need for

[2] When all $d_{ik} < 0$, only slack-improving moves are considered. A similar expression can be derived for cases where the $d_{ik}$ are unrestricted.



(Nominal Manufacturing Process)
Synthesized Netlist
Commercial Design Tool
Placed and Routed Design
Incremental Optimization (Commercial Design Tool)
Commercial ECO
Generate Timing Data (Arrival Times and Slacks)
N
Y
Timing Feasible?
ECO Instructions
Commercial Design Tool
ECO Cost Model
LP
LPECO
(Changed Manufacturing Process)

Fig. 4. Flowchart for the experiment in Section IV.

cell sizing to improve the delay. However, as the placement density grows and interconnect loads and delays increase, cell sizing is needed to recover timing.

To get an idea of how the results would change for different manufacturing changes, the benchmark c7552 is run for the four manufacturing changes in Table IV. The results in Table V show that the changes on the pmos affect the timing more than changes on the nmos. In these cases, the negative slack cannot be corrected by either LPECO or the commercial design tool. However, the LPECO gives 43% less negative slack than the commercial tool, showing that the formulation (9) is effective.

The runtime for this algorithm is dominated by the interface from the commercial tool to LPECO, which is needed to transfer timing information and gate sensitivity information. This sensitivity information is needed for any sizer, as the comparisons between competing gates must be made in the process of optimization. The linear program in LPECO takes between .01 to 103 seconds for all benchmarks (excluding the time used by the commercial physical design tool).

## V. Conclusion

In this paper, we present the idea of ECO cost, to quantify the amount of time that is needed to validate an ECO operation. We then propose a novel method for performing ECO gate sizing, and give models for the ECO that can be incorporated into the optimization procedure. This leads to results that outperform a leading commercial design tool in the timing closure, and the resulting cost of the ECO for nearly all benchmark examples.

Further research will be made to extend this algorithm to the cases of layout-transparent changes such as $v_t$ assignment and gate-length biasing, and to modify the initial design to minimize future ECO costs.

## References

[1] J. Cong and M. Sarrafzadeh, "Incremental physical design," in *Proc. Int. Conf. Physical Design*. ACM, 2000, p. 92.

[2] O. Coudert, J. Cong, S. Malik, and M. Sarrafzadeh, "Incremental cad," in *Proc. Int. Conf. Computer-Aided Design*, 2000, pp. 236–244.

[3] A. Kahng and S. Mantik, "On mismatches between incremental optimizers and instance perturbations in physical design tools," in *Proc. Int. Conf. Computer-Aided Design*. IEEE Press, 2000, p. 22.

[4] M. Moffitt, J. Roy, and I. Markov, "The coming of age of (academic) global routing," in *Proc. Int. Conf. Physical Design*. ACM, 2008, pp. 148–155.

[5] C. J. Alpert, C. Chu, and P. G. Villarrubia, "The coming of age of physical synthesis," in *Proc. Int. Conf. Computer-Aided Design*. Piscataway, NJ, USA: IEEE Press, 2007, pp. 246–249.

[6] S. Krishnaswamy, H. Ren, N. Modi, and R. Puri, "Deltasyn: an efficient logic difference optimizer for eco synthesis," in *Proc. Int. Conf. Computer-Aided Design*, 2009, pp. 789–796.

[7] Y.-P. Chen, J.-W. Fang, and Y.-W. Chang, "Eco timing optimization using spare cells," in *Proc. Int. Conf. Computer-Aided Design*. Piscataway, NJ, USA: IEEE Press, 2007, pp. 530–535.

[8] C. Chen, C. Chu, and D. Wong, "Fast and exact simultaneous gate and wire sizing by lagrangian relaxation," *IEEE Trans. on Computer-Aided Design*, vol. 18, no. 7, pp. 1014–1025, 1999.

[9] D. Nguyen, A. Davare, M. Orshansky, D. Chinnery, B. Thompson, and K. Keutzer, "Minimization of dynamic and static power through joint assignment of threshold voltages and sizing optimization," in *ISLPED '03: Proceedings of the 2003 international symposium on Low power electronics and design*. New York, NY, USA: ACM, 2003, pp. 158–163.

[10] Y. Liu and J. Hu, "A new algorithm for simultaneous gate sizing and threshold voltage assignment," in *ISPD '09: Proceedings of the 2009 international symposium on Physical design*. New York, NY, USA: ACM, 2009, pp. 27–34.

[11] M. R. C. M. Berkelaar and J. A. G. Jess, "Gate sizing in mos digital circuits with linear programming," in *EURO-DAC '90: Proceedings of the conference on European design automation*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1990, pp. 217–221.

[12] D. G. Chinnery and K. Keutzer, "Linear programming for sizing, vth and vdd assignment," in *Proc. Int. Conf. Low Power Electronics and Design*, 2005, pp. 149–154.

[13] S. Dutt and H. Arslan, "Efficient timing-driven incremental routing for vlsi circuits using dfs and localized slack-satisfaction computations," in *Proc. Design, Automation and Test in Europe*, 2006, pp. 768–773.

[14] J. Roy and I. Markov, "ECO-system: Embracing the Change in Placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 12, pp. 2173–2185, 2007.

[15] Available from http://www.opencores.org.

[16] "Nangate Open Cell Library v1.3," available from http://www.si2.org/openeda.si2.org/projects/nangatelib.

[17] MOSEK ApS, "The MOSEK Optimization Tools Version 5.0," available from http://www.mosek.com.

TABLE II

INFORMATION ON THE BENCHMARKS FOR NOMINAL PROCESS PARAMETERS

| | 70% Congestion | | | | 80% Congestion | | | | 90% Congestion | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cells | delay [ns] | power [$\mu W$] | die area [$\mu$m$^2$] | cells | delay [ns] | power [$\mu W$] | die area [$\mu$m$^2$] | cells | delay [ns] | power [$\mu W$] | die area [$\mu$m$^2$] |
| c1355 | 560 | .464 | 8.68 | 4417.0 | 533 | .513 | 6.53 | 4193.2 | 533 | .505 | 6.69 | 4013.2 |
| c1908 | 627 | .736 | 9.47 | 4780.0 | 578 | .854 | 6.69 | 4523.6 | 578 | .903 | 6.43 | 4317.3 |
| c2670 | 960 | .556 | 12.97 | 5546.4 | 906 | .656 | 10.08 | 5222.6 | 906 | .666 | 9.82 | 4959.2 |
| c3540 | 1420 | .996 | 19.60 | 7192.8 | 1415 | .989 | 19.59 | 6711.9 | 1332 | 1.228 | 14.32 | 6325.0 |
| c5315 | 1874 | .936 | 22.49 | 8518.2 | 1799 | 1.045 | 21.03 | 7909.3 | 1799 | 1.075 | 20.50 | 7420.8 |
| c6288 | 2777 | 1.912 | 39.92 | 10466.5 | 2777 | 1.914 | 40.31 | 9661.3 | 2733 | 2.323 | 29.13 | 9018.3 |
| c7552 | 2728 | .817 | 33.30 | 10091.7 | 2645 | .907 | 28.66 | 9324.6 | 2645 | .976 | 34.82 | 8713.2 |
| alu | 11370 | 1.644 | 125.71 | 28102.3 | 11200 | 1.908 | 106.38 | 25409.1 | 11201 | 1.872 | 107.95 | 23281.2 |

TABLE III

EXPERIMENTAL RESULTS

| 70% Congestion | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Commercial | | | | LPECO | | | |
| | slack$_{\text{initial}}$ [ns] | $p_{\text{initial}}$ [$\mu W$] | slack [ns] | $c_{\text{timing}}$ [pins] | $c_{\text{area}}$ [$\mu$m$^2$] | $p_{\text{l}}$ [$\mu W$] | slack [ns] | $c_{\text{timing}}$ [pins] | $c_{\text{area}}$ [$\mu$m$^2$] | $p_{\text{l}}$ [$\mu W$] |
| c1355 | -.022 | 12.21 | -.026 | 78 | 70.19 | 13.11 (1.07) | -.016 | 78 | 51.97 | 12.68 (1.03) |
| c1908 | -.031 | 12.88 | -.045 | 286 | 16.18 | 12.98 (1.01) | -.029 | 113 | .1 | 12.88 (1.00) |
| c2670 | -.011 | 17.69 | -.015 | 550 | 71.21 | 18.24 (1.03) | .000 | 479 | 53.21 | 17.84 (1.01) |
| c3540 | -.049 | 26.51 | -.070 | 657 | 76.65 | 26.72 (1.01) | -.056 | 660 | 27.24 | 26.92 (1.02) |
| c5315 | -.043 | 29.24 | -.055 | 659 | 20.38 | 29.38 (1.00) | -.049 | 423 | 6.03 | 29.41 (1.01) |
| c6288 | -.083 | 53.28 | -.099 | 425 | 79.81 | 53.63 (1.01) | -.092 | 404 | 55.08 | 54.11 (1.02) |
| c7552 | -.036 | 45.09 | -.037 | 1139 | 76.33 | 45.44 (1.01) | -.034 | 1388 | 40.47 | 45.63 (1.02) |
| alu | -.123 | 168.46 | -.045 | 8733 | 279.8 | 168.63 (1.00) | -.097 | 7861 | 65.36 | 168.92 (1.00) |
| 80% Congestion | | | | | | | | | | |
| c1355 | -.026 | 9.10 | -.012 | 276 | 82.6 | 11.94 (1.31) | .003 | 272 | 27.5 | 9.22 (1.01) |
| c1908 | -.024 | 9.09 | .002 | 453 | 30.2 | 9.72 (1.07) | .001 | 435 | 23.8 | 9.22 (1.01) |
| c2670 | -.022 | 13.64 | .001 | 549 | 27.7 | 13.86 (1.02) | .010 | 165 | 3.5 | 13.66 (1.00) |
| c3540 | -.050 | 26.53 | -.062 | 663 | 90.0 | 26.83 (1.01) | -.050 | 464 | 36.2 | 26.83 (1.01) |
| c5315 | -.049 | 27.26 | -.005 | 919 | 71.7 | 27.74 (1.02) | .000 | 781 | 58.8 | 27.11 (.99) |
| c6288 | -.092 | 53.75 | -.099 | 413 | 98.9 | 54.02 (1.01) | -.100 | 409 | 64.9 | 54.40 (1.01) |
| c7552 | -.019 | 38.67 | -.001 | 2267 | 93.0 | 39.48 (1.02) | .000 | 417 | 14.9 | 38.70 (1.00) |
| alu | -.016 | 142.31 | .044 | 10860 | 609.2 | 143.2 (1.01) | .010 | 480 | 6.6 | 142.3 (1.00) |
| 90% Congestion | | | | | | | | | | |
| c1355 | -.018 | 9.29 | -.009 | 330 | 63.3 | 10.65 (1.15) | .002 | 327 | 34.3 | 9.43 (1.02) |
| c1908 | -.036 | 8.73 | .001 | 417 | 43.0 | 9.50 (1.09) | .008 | 329 | 27.5 | 8.78 (1.01) |
| c2670 | -.015 | 13.27 | .000 | 510 | 34.5 | 13.73 (1.03) | .002 | 167 | 17.3 | 13.26 (1.00) |
| c3540 | -.038 | 19.23 | .005 | 931 | 106.31 | 20.52 (1.07) | .025 | 790 | 43.5 | 19.45 (1.01) |
| c5315 | -.048 | 26.56 | -.004 | 1030 | 83.12 | 27.16 (1.02) | .005 | 964 | 48.2 | 26.64 (1.00) |
| c6288 | -.085 | 38.87 | -.004 | 1413 | 149.48 | 41.00 (1.05) | .002 | 1002 | 102.5 | 38.69 (1.00) |
| c7552 | -.048 | 34.82 | .003 | 1742 | 172.39 | 36.24 (1.04) | .003 | 1282 | 135.3 | 34.74 (1.00) |
| alu | -.056 | 144.40 | .010 | 10198 | 586.57 | 146.12 (1.01) | .005 | 9633 | 129.32 | 143.91 (1.00) |

TABLE IV

EXPERIMENTAL MANUFACTURING CHANGES

| | nmos | | | | pmos | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $v_{\text{t}}$ | $t_{\text{ox}_e}$ | $c_{\text{gate}}$ | $l_{\text{eff}}$ | $v_{\text{t}}$ | $t_{\text{ox}_e}$ | $c_{\text{gate}}$ | $l_{\text{eff}}$ | comments |
| **base** | **-10%** | **+5%** | **+10%** | **+5%** | **-5%** | **-5%** | **+10%** | **+5%** | baseline setup used in Table III |
| case 1 | +2% | +2% | +2% | +2% | +2% | +2% | +2% | +2% | +2% case |
| case 2 | +5% | +5% | +5% | +5% | +5% | +5% | +5% | +5% | +5% case |
| case 3 | +5% | +5% | +5% | +5% | 0% | 0% | 0% | 0% | 5% nmos only |
| case 4 | 0% | 0% | 0% | 0% | +5% | +5% | +5% | +5% | 5% pmos only |

TABLE V

EFFECTS OF THE DIFFERENT MANUFACTURING CHANGES IN TABLE IV (C7552)

| (80% Congestion) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Commercial | | | | LPECO | | | |
| | slack$_{\text{initial}}$ [ns] | $p_{\text{initial}}$ [$\mu W$] | slack [ns] | $c_{\text{timing}}$ [pins] | $c_{\text{area}}$ [$\mu$m$^2$] | $p_{\text{l}}$ [$\mu W$] | slack [ns] | $c_{\text{timing}}$ [pins] | $c_{\text{area}}$ [$\mu$m$^2$] | $p_{\text{l}}$ [$\mu W$] |
| case 1 | -.086 | 22.56 | -.067 | 1434 | 16.18 | 23.34 (1.03) | -.033 | 1442 | 110.03 | 23.26 (1.03) |
| case 2 | -.122 | 16.00 | -.096 | 1166 | 71.21 | 16.54 (1.03) | -.065 | 1229 | 105.95 | 16.44 (1.03) |
| case 3 | -.086 | 16.76 | -.063 | 1435 | 76.65 | 17.33 (1.03) | -.032 | 1441 | 104.84 | 17.19 (1.03) |
| case 4 | -.099 | 27.91 | -.079 | 1315 | 20.38 | 28.82 (1.03) | -.046 | 1314 | 102.52 | 28.52 (1.02) |