Power Optimization in VLSI Layout: A Survey

Massoud Pedram

Department of EE-Systems
University of Southern California
Los Angeles CA 90089

Hirendu Vaishnav

Synopsys, Inc.
700 East Middlefiled Road
Mountain View, CA-94043

Abstract

This paper presents a survey of layout techniques for designing low power digital CMOS circuits. It describes the many issues facing designers at the physical level of design abstraction and reviews some of the techniques and tools that have been proposed to overcome these difficulties.

1. Introduction

In the past, the major concerns of the VLSI designer were area, performance, cost and reliability; power considerations were mostly of only secondary importance. In recent years, however, this has begun to change and, increasingly, power is being given comparable weight to area and speed. Several factors have contributed to this trend. Portable computing and communication devices demand high-speed computation and complex functionality with low power consumption. Heat generation in high-end computer products limits the feasible packing and performance of VLSI circuits and increases the packaging and cooling costs. Circuit and device reliability deteriorate with increased heat dissipation, and thus the die temperature. Heat pumped into the rooms, the electricity consumed and the office noise diminish with low power LSI chipset.

Our goal in writing this paper is to provide background and outlook for people interested in using or developing low power design methodologies and techniques. Even though we tried to be complete, some significant research work might have been unintentionally left out. The paper is organized as follows. First, we describe sources of power dissipation in CMOS circuits and degrees of freedom in the low power design space. We then present an in-depth survey (and in many cases analysis) of power minimization techniques and describe some of the frontiers of the research currently being pursued. We conclude by summarizing the major low power design challenges that lie ahead of us.

2. Sources of Power Dissipation

Power dissipation in digital CMOS circuits is caused by four sources as follows.

- the *leakage current*, which is primarily determined by the fabrication technology, consists of two components: 1) reverse bias current in the parasitic diodes formed between source and drain diffusions and the bulk region in a MOS transistor, and 2) the subthreshold current that arises from the inversion charge that exists at the gate voltages below the threshold voltage,
- the *standby current* which is the DC current drawn continuously from V_{dd} to ground,
- the *short-circuit (rush-through) current* which is due to the DC path between the supply rails during output transitions,
- the *capacitance current* which flows to charge and discharge capacitive loads during logic changes.

The term *static power dissipation* refers to the sum of leakage and standby dissipations. Leakage currents in CMOS circuits can be made small with proper choice of device technology. Standby currents are important only in CMOS design styles like pseudo-nMOS and nMOS pass transistor logic. In this article, we assume that the standby dissipation is insignificant, thus limiting ourselves to CMOS technologies, logic styles and circuit structures [19] in which this condition holds.

The short-circuit power consumption for an inverter gate is proportional to the input ramp time, the load and transistor sizes of the gate. The maximum short circuit current flows when there is no load; this current decreases with the load. Depending on the approximations used to model the currents and to estimate the input signal dependency, different formulae [55] [17], with varying accuracy, have been derived for the evaluation of the short circuit power. A useful formula was recently derived in [51] that shows the explicit dependence of the short circuit power dissipation on the design and performance parameters, such as transistor sizes, input and output ramp times and the load. The idea is to adopt an alternative definition of the short circuit power dissipation, through an equivalent (virtual) short circuit capacitance C_{SC}.

The dominant source of power dissipation CMOS circuits is the charging and discharging of the node capacitances (also referred to as the capacitive power dissipation) and is given by:

$$P = 0.5 C_L V_{dd}{}^2 E(sw) f_{clk} \tag{1}$$

where C_L is the physical capacitance at the output of the node, V_{dd} is the supply voltage, $E(sw)$ (referred to as the *switching activity*) is the average number of output transitions per $1/f_{clk}$ time, and f_{clk} is the clock frequency.

The term *dynamic power dissipation* refers to the sum of short circuit and capacitive dissipations. Using the concept of equivalent short-circuit capacitance described above, the dynamic power dissipation can be calculated using equation (1) if we add C_{SC} to C_L. Short-circuit currents in CMOS circuits can be made small with appropriate circuit design techniques [55]. In most of this article, we will thus focus on capacitive power dissipation.

In the above, we alluded to the three degrees of freedom inherent in the low-power design space: voltage, physical capacitance, and data activity. Optimizing for power entails an attempt to reduce one or more of these factors.

3. Power Minimization Techniques

To address the challenge to reduce power, the semiconductor industry has adopted a multifaceted approach, attacking the problem on four fronts:

1. **Reducing chip and package capacitance:** This can be achieved through process development such as SOI with partially or fully depleted wells, CMOS scaling to submicron device sizes, and advanced interconnect substrates such as Multi-Chip Modules (MCM).This

approach can be very effective but is also very expensive and has its own pace of development and introduction to the market.

2. **Scaling the supply voltage:** This approach can be very effective in reducing the power dissipation, but often requires new IC fabrication processing. Supply voltage scaling also requires support circuitry for low-voltage operation including level-converters and DC/DC converters as well as detailed consideration of issues such as signal-to-noise margins.
3. **Employing better design techniques:** This approach promises to be very successful because the investment to reduce power by design is relatively small in comparison to the other three approaches and because it is relatively untapped in potential.
4. **Using power management strategies:** The power savings that can be achieved by various static and dynamic power management techniques are very application dependent, but can be significant.

The various approaches interact with one another, for example CMOS device scaling, supply voltage scaling, and choice of circuit architecture must be done judiciously and carefully in order to find an optimum power-area-delay trade-off. In the following, we will focus on CAD algorithms and techniques for low power. These techniques span various levels of the design abstraction from algorithmic and system level down to layout and circuit level. In this paper, we will consider power optimization techniques at the physical design level only.

Once various system level, architectural and technological choices are made, it is the switched capacitance of the logic that determines the power consumption of a circuit. The strategy for designing circuits for low power consumption will therefore be to optimize the circuit to obtain low switching activity factors at nodes which drive large capacitive loads.

3.1. Physical Design Automation

Physical design fits between the gate-level specification and the geometric (mask) representation known as the layout. It provides the automatic layout of circuits minimizing some objective function subject to given constraints. Depending on the target design style (General Cells, Standard Cells, Gate Arrays, FPGAs), the packaging technology (printed circuit boards, multi-chip modules, wafer-scale integration) and the objective function (area, delay, power, reliability), various optimization techniques are used to partition, place, resize and route gates.

Under a zero-delay (glitch-less) model, the switching activity of gates remains unchanged during layout optimization, and hence, the only way to reduce power dissipation is to decrease the load on high switching activity gates by proper netlist partitioning and gate placement, gate and wire sizing, transistor reordering, and routing. Layout problems become more complicated under a real-delay model, which accounts for glitches in the circuit, because layout optimization operations influence the glitch activity in ways that cannot be accurately and reliably predicted.

In the recent past, post-layout optimization techniques (such as buffer and wire sizing, local restructuring and re-mapping) for power reduction (or area and delay recover given a fixed power budget) have become commonplace. The advantage of these techniques is that re-synthesis tools allow more global changes to the circuit structure compared to layout tools. At the same time, the re-synthesis tools have access to detailed post-layout information that allows accurate estimation of circuit area, delay and power dissipation.

3.1.1 Circuit Partitioning

Netlist partitioning is key in breaking a complex and large design into smaller pieces which are subsequently optimized and implemented as separate blocks. This is often needed to satisfy I/O pin constraints on the blocks, reduce the complexity of subsequent optimization steps, or improve performance. Traditionally, the objective functions for partitioning have been the cut-size and/or the circuit delay while the constraints have been I/O pin count per block and block size. Partitioning for low power has recently become an important problem.

<< **Figure 1 Goes Here.** >>

In general, the off-block capacitances are much higher than the on-block capacitances (one to two orders of magnitude). It is therefore essential to develop partitioning schemes that keep the high switching activity nets entirely within the same block as much as possible. Figure 1 depicts a simple netlist where the edge weights reflect the switching activity of the corresponding nets. Here, the minimum cutsize solution, that is (a), leads to higher switched capacitance while the minimum switched capacitance cut, that is (b), leads to higher number of nets crossing the cutline.

Techniques based on local neighborhood search (e.g., the Kernighan-Lin algorithm [20]) or simulated annealing [21] can be easily adapted to do this. In particular, it is adequate to assign net weights based on the switching activity values of the driver gates and then find a minimum cost partitioning solution. Performance-Driven Circuit Clustering

3.1.2 Node Clustering

As a result of logic extraction, it is possible to increase the circuit depth to such an extent that the circuit delay becomes unacceptably large. This problem is often mitigated by a *reduce_depth* operation that implements a depth optimal node clustering algorithm based on [23]. This algorithm however makes no attempt to explore alternative clustering solutions that result in the same logic depth, but have lower power dissipation.

In [53] a mechanism is described that implicitly generates all non-inferior power-delay clustering solutions and selects the one which has minimum logic depth, but lower power dissipation. This is achieved by enumerating, in postorder, all candidate clusters of up to a maximum cluster size and selecting the power-optimal cluster solution for each delay value at every gate in the circuit. The algorithm which is linear in circuit size but exponential in the maximum cluster size, is provably power- and delay-optimum for trees. The algorithm produces optimum delay solutions for general directed acyclic graphs, but the results are not power-optimum because of the possible logic duplication at the multiple fanout nodes in the circuit. Thus, it is often necessary to perform a delay-constrained power-recovery step as a post-process. Experimental results indicate that, on average, 25% improvement in power dissipation of multi-level Boolean circuits is obtained without any increase in circuit delay (assuming that the physical capacitance on inter-cluster lines is much higher than the capacitance on intra-cluster lines).

<< Figure 2 Goes Here. >>

Two example clustering solutions are shown in Figure 2 where the solution on the left is obtained by Lawler's algorithm while the solution on the right corresponds to power and delay optimal clustering solution (the maximum cluster size is seven). In this example, all input activities are set to 0.5 and the numbers shown beside the nodes represent their switching activities obtained by symbolic simulation of the Boolean network. Both solutions have a depth of two. However, the power cost (switched capacitance) of *inter-cluster* lines in Clustering A is 1.3 while that in Clustering B is 0.65. Experimental results indicate that, on average, 25% improvement in power dissipation of multi-level Boolean circuits is obtained without any increase in circuit delay (assuming that the physical capacitance on inter-cluster lines is much higher than the capacitance on intra-cluster lines).

3.1.3 Floorplanning

Floorplanning is the process of assigning shapes, pin positions and locations to a set of macro-cells or modules so as to minimize the area of the floorplan. One successful floorplanning approach is based on computing the shape functions (height versus width trade-off curves) during a postorder traversal of a cluster tree that captures the connectivity among modules. The optimal floorplan topology, block shapes and room assignments, and pin positions (or block orientations) are determined during a preorder traversal of this tree [60] [32]. The two dimensional shape function curves can be indexed by the power cost, that is, for each distinct power dissipation value, one shape function is built. These indexed shape functions can then be used during the preorder traversal to compute the optimal power solution which also leads to minimum chip area (see [5] for details).

3.1.4 Placement

Placement refers to the process of assigning locations to gates in a circuit netlist. Placement algorithms can be easily modified to minimize the power dissipation. For example, a common placement algorithm for small-cell ICs is to formulate the problem as a constrained mathematical programming problem and then solve it in two phases: global optimization and slot assignment [47] [22]. The objective function is the sum of squares of net lengths while the constraints are center-of-mass and/or path-based timing constraints. The only change needed in the low power formulation is to use the sum of squares of switched capacitances as the objective function during each phase [53] as detailed next.

Let

$$I = \{i_1, i_2, \ldots, i_I\}$$

$$M = \{m_1, m_2, \ldots, m_M\}$$

$$O = \{o_1, o_2, \ldots, o_O\}$$

$$N = \{n_1, n_2, \ldots, n_N\}$$

denote the sets of primary inputs, internal gates, primary outputs, and nets, respectively. The total number of nets N is given by $I+M$. Each gate i has a 4-tuple $<E_i, C_i^{in}, C_i^{wire}, C_i^{gate}>$ associated with it where E_i, C_i^{in}, C_i^{wire}, and C_i^{gate}denote the switching rate of the gate, its input capacitance, the wire load due to output net n_i, and the gate load of n_i, respectively. Let us denote the set of gates connected by net n_i by γ_i. The total power consumption for the circuit is given by:

$$P = \frac{1}{2}\frac{V_{dd}^2}{T_{clk}} \sum_{i \in \{I,M\}} \left(C_i^{wire} + C_i^{gate}\right)E_i$$

The term C_i^{gate}. E_i is independent of the placement and hence, is dropped from the objective function. After dropping the constant multiplication factor 0.5. V_{dd}^2. f_{clock}, we get the following objective function for low power placement:

$$L_1 = \sum_{i \in \{I,M\}} \{C_i^{wire} \cdot E_i\}$$

Under a quadratic formulation the objective function is the sum over all nets of the square of power consumption due to each net. As usual, net n_i is modeled by a weighted clique. Therefore, the quadratic objective function for low power placement is or after dropping constants and rearranging:

$$L_2 = \sum_{i \in \{I,M\}} \left(C_i^{wire} \cdot E_i\right)^2$$

which can be re-written as:

$$L_2(x,y) = \sum_{j,k \in \{I,M\}} F_{j,k} \cdot \left[\langle x_j - x_k\rangle^2 + \langle y_j - y_k\rangle^2\right]$$

where $F_{j,k}$ is zero if no net contains both gates j and k and is $\sum E_i^2 \cdot \frac{2}{|\gamma_i|}$ where i denotes any net that contains both gates j and k.

The objective function can be formulated using the matrix notation as:

$$L_2(x,y) = \frac{1}{2}\left(X^T BX + Y^T BY\right) + c^T X + d^T Y$$

where **X** and **Y** are vectors of *x* and *y* coordinates of the gate locations and **c** and **d** are constant coefficient vectors. Matrix **B** is a symmetric matrix derived from matrix $F = [F_{j,k}]$ as follows:

$$B = D - F$$

where **D** is a diagonal matrix with

$$d_{j,j} = \sum_k F_{j,k}$$

This objective function is similar to the one derived in PROUD [47] and GORDIAN [42]. It can be shown that as long as the whole circuit forms one connected set and some modules are fixed, **B** is positive definite. This implies that the objective function is convex and that quadratic optimization techniques can be applied to obtain a global optimal solution.

The approach used to solve the above convex programming problem is to interleave quadratic optimization with circuit bi-partitioning of the circuit. After each global optimization step, circuit is further partitioned, and the partitioning information is introduced in the subsequent global quadratic optimization step as center of mass constraints. Unlike the divide-and-conquer mechanisms proposed in literature, this mechanism maintains a global view of the circuits beyond the partition boundaries and allows migration of gates across boundaries.

Long path delay constraints can be written in terms of required times at primary outputs and arrival times at primary inputs of the circuit and added to the above programming problem. These constraints are then integrated with the objective function. Thus, we have a convex objective function with a set of convex constraints. The resulting constrained optimization problem can be formulated as a Lagrangian function optimization and solved efficiently using Lagrangian relaxation techniques. The slot assignment phase for the performance-driven version of placement tool is identical to the area-driven version of the placement tool. However, a simple modification that restricts assignment of gates on critical path to slots that may increase the delay can be incorporated.

With this modification, an average power reduction of about 10% has been obtained compared to the minimum net length solution without any increase in circuit delay. The paper also addresses the problem of hazard minimization during placement by proposing to integrate the gate-level power estimation programs with cell placement programs. The work of [53] can be easily extended to minimize the sum of switched capacitances using a formulation and computational procedure similar to that of [42].

3.1.5 Global Routing

Global routing produces routing trees for all nets in the circuit so as to minimize the interconnect length and/or chip area. The routing trees for multi-terminal nets are often constructed as Rectilinear Spanning or Steiner trees. In routing a single net to achieve lower power dissipation, the goal is to minimize the physical capacitance which coincides with the minimum length objective used in conventional routing. Therefore, there is no new routing problem here. In routing a collection of nets in fixed-size routing channels (e.g., Gate Array or FPGA layouts), in variable-width routing channels (e.g., Standard Cell layout) or in general area (e.g., General Cell layouts), however the difference between minimizing the total physical capacitance and the total switched capacitance comes to surface. In the following, Standard Cell layout will be used as an example.

The main tasks of a global router for Standard Cell layouts are to generate the routing topology for each net and to determine the number of feedthrough cells required on each cell row. Both sequential [39] and parallel [6] [24] routing algorithms for routing in have been proposed.

Sequential routing algorithms route each net separately. They assign a feedthrough penalty to each cell row which characterizes the additional cost (in terms of layout area) that a routing tree edge accrues if it crosses that row. Typically, the longest cell row in the circuit is assigned the highest feedthrough penalty to discourage use of additional feedthrough cells on that row. Sequential routing algorithms can be modified to produce minimum-power routing solution by simple net weighting where the net weights are derived from the switching activity values of the driver gates. Nets with higher weights are given priority during routing and thus tend to assume their smallest possible routes. In contrast, low activity nets will encounter high feedthrough penalties on their most desired routing edges and thus tend to assume longer lengths than is ideally possible. Parallel routing algorithms alleviate the net ordering problem by constructing routing trees for all nets concurrently. One can modify the feedthrough insertion and net segment assignment steps in these routers to generate tree connections with smaller lengths for nets that are driven by gates with higher switching rates [54].

<< **Figure 3 Goes Here.** >>

Feedthrough assignment is the interface between global and detailed routing as it assigns exact positions to the feedthrough cells on each cell row and hence defines the pin positions on the channel boundaries. Figure 3 shows a scenario where nets A and B are competing for their optimum feedthrough location. If, for example, net A has a higher switching activity than net B, then configuration (a) will lead to lower power dissipation as it minimizes the switched capacitance.

One can perform feedthrough assignment during the global routing, but then the result will heavily depend on the net ordering. Consequently, this problem is often formulated as a linear assignment problem for each cell row as in [46] [31] [27]**.** Other techniques that perform feedthrough assignments for all cell rows simultaneously [15] [30] do not show much improvement compared to the linear assignment based technique which are outlined next. The idea of the latter techniques is to set up a linear assignment cost matrix where rows correspond to nets crossing the cell row, columns correspond to feedthrough cells on the cell row, and each entry *(i,j)* in the matrix gives the cost of using feedthrough cell *j* for net *i*. The problem is then solved by finding a minimum-cost cover of all rows in the matrix. For the low power version of this problem, one can modify each entry in the linear assignment cost matrix according to the switching activity of the net, that is, active nets will incur higher cost if they do not occupy their preferred (minimum-cost) feedthrough cell.

Experimental results have produced only marginal improvements in power dissipation. This is because global routing is a complex process where the net lengths and channel congestion are dictating the routing solution for each net; an extra weighting factor for the nets can only produce a sizeable difference in the final result if net activities (especially on large nets where global routers have many options to route them) are drastically different. This condition was not met in the examples attempted in [54].

3.1.6 Detailed Routing

Detailed routing produces the wiring geometries and layer assignments within a routing channel, switchbox or general area. Again, we will only consider channel routing techniques commonly used in Standard Cell layouts. Given the channel length, top and bottom terminal lists, left and right connection lists, and the number of routing layers, the channel routing problem is to find interconnections of all the nets in the channel including the connection sets so that the channel achieves minimum height. The objective function for low power routing becomes the switched capacitance within the channel, that is, high activity nets should assume their shortest possible route at the expense of low activity nets. One must however achieve this with no or little increase in channel height, since otherwise, the increase in wire lengths due to larger layout area will more than compensate the reduction of switched capacitances within the routing channels.

Two-layer channel routing algorithms can be classified as net-based (Left-Edge algorithm of [16] and dogleg router of [10]), constraint graph-based (net-merging algorithm of [59], column based (greedy router of [38]), and hierarchical [4]. To reduce power dissipation during channel routing, one can give high priority to active nets in using the available routing resources (e.g., tracks, layers, dogleg positions). This can be achieved in a net-based algorithm by sorting nets and in a greedy router by modifying the rule set to favor high activity nets. The modifications to the net-merging and hierarchical routing algorithms are not so straight-forward. There are no reported experimental results on power savings in VLSI circuits due to use of a low-power channel routing algorithm.

Power dissipation due to cross-talk is also an important concern to today's dense and high performance layouts. This can be minimized by ensuring that wires carrying high activity signals are placed sufficiently far from the other wires.

3.1.7 Transistor and Gate Sizing

If performance was not a design constraint, design for low (capacitive) power would be achieved by using minimum-sized gate versions everywhere. The gate sizing problem is thus to find a minimum power solution subject to meeting a given delay constraint.

An efficient approach to *continuous* (generator-based) gate sizing for low power is to linearize the path-based timing constraints and use a linear programming solver to find the global optimum solution [2]. This work has been extended to handle setup and hold time constraints in [44]. The drawbacks of this approach are the omission of slope factor (input ramp time) for input waveforms from the delay model and use of a simple power dissipation model that ignores short-circuit currents. The LP-based cell selection algorithm can be easily extended to account for the short-circuit power dissipation as described in [33]. More recently, the authors of [40] present a different convex programming approach for solving the gate sizing problem for minimum power dissipation. The authors report that by including the short-circuit power in the objective function (along with capacitive power), the minimum-power solution often corresponds to faster solutions compared to the case where only capacitive power is minimized.

A heuristic technique for *discrete* (library-based) gate sizing for minimum power subject to a given delay constraint is described in [45]. The idea is to start with minimum-sized gate versions, and then size up gates along the paths with negative slacks (that is, critical paths) so as to satisfy the constraints while increasing the switched capacitance of the circuit minimally. Alternatively, one may start with the fastest possible design and then size down the gates along the paths with positive slack (compared to the given delay constraint) so as to maximize the reduction in switched capacitance. Another technique presented in [26], starts with a circuit that satisfies the timing constraint and sizes down certain gates (which are not necessarily on the non-critical paths) to reduce the power dissipation. The shortcoming of these approaches is their greedy nature which leads to sizing one gate a time.

<< Figure 4 Goes Here. >>

Discrete gate sizing problem is a special case of technology mapping problem and thus the dynamic programming technique can be applied to build the power-delay trade-off curves during a postorder traversal of the circuit and then perform the gate selection during a preorder traversal so as to satisfy the delay constraints while minimizing the switched capacitance (see Figure 4 for an example trade-off curve).

In [3], the problem of transistor sizing in a static CMOS layout to minimize the capacitive plus short circuit power dissipation. It is shown that the power-optimal size for the transistors in a gate that is driving a given load, can be larger than minimum size. The authors next derive the power-delay optimal sizes for these transistors and present a greedy algorithm for calculating the optimal power sizing subject to a given delay constraint for all gates in a circuit. This algorithm starts by doing an initial power-optimal transistor sizing on each gate. If the power-minimal layout satisfies the delay constraint, the process is terminated; otherwise, the power-delay optimal sizing is applied to gates on the critical paths until the timing target is met.

Various researchers [3] [26] [40] have reported about 15-20% reduction in total power dissipation as a result of cell selection or transistor sizing.

3.1.8 Transistor Reordering

In general, library gates have pins that are functionally equivalent which means that inputs can be permuted on those pins without changing function of the gate output. These equivalent pins may have different input pin loads and pin dependent delays. It is well known that the signal to pin assignment in a CMOS logic gate has a sizeable impact on the propagation delay through the gate [19].

If we ignore the *parasitic* (internal) power dissipation due to charging and discharging of source/drain to bulk diffusion capacitances inside a CMOS logic gate, it becomes self-evident that high switching activity inputs should be matched with pins that have low input capacitance. This scheme is however not very effective as in the semi-custom libraries, the difference in pin capacitances for logically equivalent pins is small. The parasitic power dissipation varies in turn as a function of the switching activities and pin assignment of the input signals. To find the minimum power pin assignment for a gate that accounts for this internal power dissipation, one must solve a difficult optimization problem as formulated in [49]. As the number of functionally equivalent pins in a typical semi-custom library is not greater that six, it is feasible to exhaustively enumerate all pin permutations to find the minimum power pin assignment.

<< **Figure 5 Goes Here.** >>

One can also use heuristics, for example, one such rule assigns input signal with the largest probability of assuming a controlling value (zero for nMOS and one for pMOS for series-connected transistors in the pull-up or pull-down blocks of a logic gate) to the transistor near the output terminal of the gate [33]. The rationale is that this transistor (e.g., N1 in Figure 5) will switch off more frequently, thus blocking the internal nodes (e.g., nodes x and y in Figure 5) from non-productive charging and discharging. Another rule is presented in [34] where the input that has the highest switching activity when all other inputs are set to their non-controlling values (one for nMOS and zero for pMOS in series-connected transistors) is directed to the input closest to the output terminal. The rationale is that the internal node capacitances (e.g., *Cx* and *Cy* in Figure 5) will have been discharged by the time that the output transition takes place as a result of an input change for the transistor closest to the output terminal (e.g., N1 in Figure 5). The authors of [41] derive similar rules to those mentioned above and point out that if there is a conflict between the two rules, then the transistor ordering should be determined by the ratio of the probability of assuming controlling value over probability of making transitions, that is input with the highest ratio will be placed closest to the output terminal. Experimental results show that about 5% power reduction can be achieved by transistor ordering.

In general, pin permutation for minimum delay produces results that are very different from those obtained for minimum power. Therefore, pin permutation for low power should take place on non-critical gates.

3.1.9 Super Buffer Design

Super buffer design is a chain of inverters designed to derive a large capacitive load with minimal signal propagation time [19]. A power-optimal buffer sizing technique applicable to the design of super buffers at high speed is presented in [61]. This work is based on an analytic relationship among signal delay, power dissipation, driver size and interconnect load which is in turn derived from the *I-V* characteristics of CMOS transistors. This work shows that optimal-power sizing requires a variable tapering (scaling) factor for the inverter chain.

3.1.10 Wire Sizing, Driver Sizing and Buffer Insertion

Wire and/or driver sizing are often needed to reduce the interconnect delay on time-critical nets. Wire sizing however tends to increase the load on the driver and hence increase the power dissipation. A simultaneous wire and driver sizing approach can reduce the interconnect delay with only a small increase in the power dissipation. The approach in [7] and [8] uses the properties of *monotonicity*, *separability* and *dominance* (which apply to Elmore delay model) to determine lower and upper bounds on the wire and driver sizing solution. The delay is measured using the distributed Elmore delay model and power estimations include both capacitive and short circuit power components. In the following, we review the work described in [7].

<< Figure 6 Goes Here. >>

Assume that we are given a tree T implementing a signal net which consists of a source $N+$ and a set of m sinks $\{N_1, N_2, ..., N_m\}$. Assume that $\{E_1, E_2, ..., E_n\}$ is the set of segments forming the tree T, where n is the number of segments in the tree. Each wire segment has a set of discrete choices of wire widths $\{W_1, W_2, ..., W_r\}$ ($W_i < W_j$ for $i < j$). We use e_i to denote the width of wire segment E_i. Furthermore, assume that the signal is driven by a chain of cascaded drivers of k stages at the source as shown in Figure 6. We use $\{d_1, d_2, ..., d_k\}$ to denote the driver sizing solution. We assume (after normalization) that $d_1=1$. Given a routing tree T, the simultaneous driver and wire sizing for both delay and power minimization is to determine the number of stages k, a driver sizing solution and a wire sizing solution on T, such that a linear combination of delay and power costs is minimized. Delay cost is measured as the weighted sum of source-sink Elmore delays while power cost accounts for both capacitive and short-circuit power dissipations.

Experimental results show that for the same delay constraint, this approach reduces the power by about 15-20% when compared to the conventional method of driver sizing only.

An optimal gate and wire sizing approach based on convex programming techniques which avoids the monotonicity and separability assumptions of the delay model is presented in [28]. This method can be easily extended to determine the optimal gate size and wire widths so as to minimize the power dissipation instead of the area required for the circuit layout.

A power- and delay-optimal algorithm for discrete buffer insertion and wire sizing under the Elmore delay model was presented in [25]. Because of the dynamic programming approach used, the entire power-delay trade-off curve for simultaneous wire sizing and buffer insertion could be generated. The authors show that using five discrete buffer sizes (1X to 8X) and wire widths ranging from 0.5 µm to 5 µm, the unsized/unbuffered solution for a 20 sink net yields a delay of 7 ns and a total switched capacitance of 4.7 nF while the delay-optimal solution gives a delay of 2.7 ns and a switched capacitance of 5.5 pF.

3.1.11 Clock Tree Generation

Clock is the fastest and most heavily loaded net in a digital system. Ideally, clock signals should have minimum rise/fall times, specified duty cycles and zero skew. Power dissipation of the clock net contributes a large fraction of the total power consumption in a digital circuit [11], thus, it is also desirable to minimize the total capacitive load seen by the clock source.

Many zero-skew clock routing algorithms have been proposed. In one approach, a chain of drivers is introduced at the source and zero-skew is achieved by wire (length) elongation [48] [12] or wire (width) sizing [62] [13]. Work of [35] proposes construction of initial non-zero skew clock routing solutions which can be sized to achive a prescribed skew bound. In another approach, buffers are inserted at internal points in the clock tree [36] [58] for satisfying source-sink path delay constraints and for minimizing the area of the clock net. The rationale is that instead of increasing wire widths and lengths to reduce the skew which will result in increased power dissipation, one can use a balanced buffer insertion scheme to partition a large clock tree into a small number of subtrees with minimum wire widths. Authors of [36] propose concurrent buffer insertion and wire width adjustment in a clock tree and report power-delay product figures as a function of the number of buffer insertion levels. Assuming a fixed clock tree topology, authors of [58] propose a technique for inserting buffers into an equal path-length clock tree. These buffers are subsequently sized to minimize the power dissipation of the clock tree under the specified clock skew constraint. The buffer sizing problem in turn is formulated as a posynomial programming problem and solved optimally. This techniques results in 60-70% power savings in the clock tree compared to the single driver scheme with wire sizing that achieves the same clock skew.

In [56] a technique for low power clock synthesis that simultaneously inserts buffers and generates the clock tree topology is presented. The main result of this paper is that by simultaneous buffer insertion and clock tree topology generation, one can reduce the total wire length (and hence power dissipation) needed to achieve zero-skew in the clock tree by 50% compared to the scheme which separates the topology generation and buffer insertion steps. Experimental results show improvements in terms of area, rise/fall times and power dissipation compared to the case where buffers are inserted into clocks as a post-processing step. The paper also reaffirms that inserting buffers at internal nodes of the clock tree leads to better results compared to inserting buffers at the root of the clock tree only.

Zero-skew is imposed to ensure correct circuit operation. In practice, circuits function correctly within a tolerable clock skew. The objective of low power clock routing is thus to minimize the load on the clock drivers (and hence the clock tree length) subject to meeting a tolerable clock skew (or more precisely, the source-sink path delay constraints). Algorithms for minimum cost bounded skew clock and Steiner tree routing are described in [9], [18] and [29]. TMethods in [9] and [18] are based on the observation that with some skew bounds, the feasible locations for the Steiner points in the routing tree become octilinear convex polygons. The tree topology is generated by successively joining the two nearest feasible regions while the placement of Steiner points in the feasible region is done heuristically. These works however only consider the skew bound and do not control the maximum source-sink delay. Consequently, excessively long wires may be generated which then requires more buffers and leads to slower rise/fall times. More buffers and slower transition times in turn results in higher power dissipation. In [29], a linear programming based algorithm for solving the lower and upper bounded delay routing tree (LUBT) construction problem is presented. LUBT is a Steiner tree rooted at the source such that the delay from the source to each sink lies between the given lower and upper bounds for that sink. The proposed method produces minimum cost LUBT for the given tree topology. Unlike the previous works which merely control the amount of skew, this solution constructs trees with distinct lower and upper bounds on the source-sink delays, hence, it can exploit all the flexibility that is present in low power, high performance clock routing tree design.

3.1.12 Power Distribution

As the supply voltage is reduced, the noise margins are diminished, thus, small voltage drop in the power distribution may have a relatively big impact on the circuit speed. Careful power distribution is thus becoming more important at lower supply voltages. In [57], a technique for concurrent topology design and wire sizing in power distribution networks is presented. The objective is to minimize the layout area while limiting the average current density to avoid electromigration-induced reliability problems and large resistive voltage drops. This technique is based on the observation that when two sinks do not draw currents at the same time, narrow wires can be used for power distribution to those sinks, thus reducing the layout area. The authors report up to 30% area savings compared with the “star” routing scheme.

4. Summary

The need for lower power systems is being driven by many market segments. There are several approaches to reducing power, however the highest

Return-On-Investment approach is through designing for low power. Unfortunately designing for low power adds another dimension to the already complex design problem; the design has to be optimized for Power as well as Performance and Area.

Optimizing the three axes necessitates a new class of power conscious CAD tools. The problem is further complicated by the need to optimize the design for power at all design phases. The successful development of new power conscious tools and methodologies requires a clear and measurable goal. In this context the research work should strive to reduce power by 5-10x in three years through design and tool development.

In summary, low power design requires a rethinking of the conventional design process, where power concerns are often overridden by performance and area considerations. This article presented a detailed coverage of low power design methodologies and techniques ranging from technology and devices to circuits and systems. In addition to offering a broad introduction to low power electronics, the article offers an extensive set of references that can be used by researchers.

5. Acknowledgment

This work was performed in part under ARPA contract No. F33615-95-C1627, SRC contract No. 94-DJ-559, and NSF NYI award No. MIP-9457392.

6. References

[1] H. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, Menlo Park, CA, 1990.

[2] M. Berkelaar and J. Jess. " Gate sizing in MOS digital circuits with linear programming. " In *Proceedings of the European Design Automation Conference*, pages 217-221, 1990.

[3] M. Borah, R. M. Owens and M. J. Irwin. " Transistor sizing for minimizing power consumption of CMOS circuits under delay constraint. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 167-172, April 1995.

[4] M. Burstein and R. Pelavin. " Hierarchical channel router. " In *Proceedings of the 20th Design Automation Conference*, pages 591-597, 1983.

[5] K. Y. Chao and D. F. Wong. " Low-power consideration in floorplan design. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 45-50, April 1994.

[6] J. Cong and B. T. Preas. " A new algorithm for standard cell global routing. " In *Pro-*

ceedings of the IEEE *International Conference on Computer Aided Design*, pages 176-180, November 1988.

[7] J. Cong, C-K. Koh and K-S. Leung. " Simultaneous driver and wire sizing for performance and power optimization. " IEEE *Transactions on VLSI Systems*, 2(4):408-425, December 1994.

[8] J. Cong and K-S. Leung. " Optimal wire sizing under Elmore delay model. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(3):321-336, March1995.

[9] J. Cong and C-K. Koh. " Minimum-cost bounded-skew clock routing. " In *Proceedings of the International Symposium on Circuits and Systems*, pages 215-218, 1995.

[10] D. N. Deutch. " A dogleg channel router. " In *Proceedings of the 13th Design Automation Conference*, pages 425-433, 1976.

[11] D. Dobberpuhl et al, " A 200MHz, 64b, dual issue CMOS microprocessor. " *Digest of Technical Papers, ISSC '92*, pages 106-107, 1992.

[12] M. Edahiro. " A clustering-based optimization algorithm in zero-skew routings. " In *Proceedings of the 30th Design Automation Conference*, pages 612-616, June 1993.

[13] M. Edahiro. " Delay minimization for zero-skew routing." In *Proceedings of the International Conference on Computer Design,* pages 563-566, November 1993.

[14] B. J. George, D. Gossain, S. C. Tyler, M. G. Wloka, and G. K. H. Yeap. " Power analysis and characterization for semi-custom design. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 215–218, April 1994.

[15] X. L. Hong, J. Huang, C. K. Cheng, and E. S. Kuh. " FARM: an efficient feed-through pin assignment algorithm. " In *Proceedings of the 29th Design Automation Conference*, pages 530-535, June 1992.

[16] A. Hashimoto and J. Stevens. " Wire routing by optimization channel assignment within large apertures. " In *Proceedings of the 8th Design Automation Workshop*, pages 155-163, 1971.

[17] N. Hedenstierna and K. Jeppson. " CMOS circuit speed and buffer optimization. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 6(3):270-281, March 1987.

[18] D. J. Huang, A. B. Kahng and C. W. Tsao. " On the bounded-skew clock and Steiner tree problems. " In *Proceedings of the 32nd Design Automation Conference*, pages 508-513, June 1995.

[19] S. M. Kang and Y. Leblebici. *CMOS Digital Integrated Circuits: Analysis and Design.* McGraw-Hill Companies, Inc. 1996.

[20] B. W. Kernighan and S. Lin. " An efficient heuristic procedure for partitioning graphs. " *Bell System Technical Journal*, 49(2):291-307, February 1970.

[21] S. Kirkpatrick and C. D. Gelatt and M. P.Vecchi. " Optimization by simulated annealing. " *Science,* 220(4598):671-680, May 1983.

[22] J. M. Kleinhans, G. Sigl, F. M. Johannes and K. J. Antreich. " GORDIAN: VLSI placement by quadratic programming and slicing optimization. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(3):356-365, March

1991.

[23] E. L. Lawler and K. N. Levitt and J. Turner. " Module clustering to minimize delay in digital networks. " IEEE *Transactions on Computers*, pages 45-57, January 1969.

[24] K. W. Lee and C. Sechen. " A new global router for row-based layout. " In *Proceedings of the* I EEE *International Conference on Computer Aided Design*, pages 180-183, November 1988.

[25] J. Lillis, C-K Cheng and T-T. Y. Lin. " Optimal wire sizing and buffer insertion for low power and a generalized delay model " In *Proceedings of the International Conference on Computer Design*, pages 138-143, November 1995.

[26] H-R. Lin and T-T. Hwang. " Power reduction by gate sizing with path-oriented slack calculation. " In *Proceedings of the 1st Asia-Pacific Design Automation Conference*, pages 7-12, August 1995.

[27] G. Meixner and U. Lauther. " A new global router based on a flow model and linear assignment. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 44-47, November 1990.

[28] N. Menezes, R. Baldick and L. T. Pileggi. " A sequential quadratic programming approach to concurrent gate and wire sizing. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 144-151, November 1995.

[29] J. Oh and M. Pedram. " Constructing lower and upper bounded delay routing trees using linear programming. " To appear in *Proceedings of the 33rd Design Automation Conference*, June 1996.

[30] T. Okamoto, M. Ishikawa, and T. Fujita. " A new feed-through assignment algorithm based on a flow model. " In *Proceedings of the 30th Design Automation Conference*, pages 775-778, June 1993.

[31] T. M. Parng and R. S. Tsay. " A new approach to sea-of-gates global routing. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 52-55, November 1989.

[32] M. Pedram, M. Marek-Sadowska and E. S. Kuh. " Floorplanning with pin assignment. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 98-101, November 1990.

[33] M. Pedram. " Power estimation and optimization at the logic level. " Int'l Journal of High Speed Electronics and Systems, 5(2):179-202, June 1994.

[34] S. C. Prasad and K. Roy. " Circuit optimization for minimization of power consumption under delay constraint. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 15-20, April 1994.

[35] S. Pullela, N. Menezes and L. T. Pillage. " Reliable non-zero skew clock tree using wire width minimization. " In *Proceedings of the 30th Design Automation Conference*, pages 165-170, June 1993.

[36] S. Pullela, N. Menezes, J. Omar and L. T. Pillage. " Skew and delay optimization for reliable buffered clock trees. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 556-562, November 1993.

[37] J. Rabaey and M. Pedram (Editors). *Low Power Design Methodologies*. Kluwer Aca-

demic Publishers, 1996.

[38] R. Rivest and C. Fiduccia. " A greedy channel router. " In *Proceedings of the 19th Design Automation Conference,* pages 418-424, 1982.

[39] K. A. Roberts. " Automatic layout in the Highland system. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 224-226, November 1984.

[40] S. Sapatnekar and W. Chuang. " Power versus delay in gate sizing: conflicting objectives? " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages , November 1995.

[41] W-Z. Shen, J-Y. Lin and F-W. Wang. " Transistor reordering rules for power reduction in CMOS gates. " In *Proceedings of the 1st Asia-Pacific Design Automation Conference*, pages 1-5, August 1995.

[42] G. Sigl and K. Doll and F. M. Johannes. " Analytical placement: a linear or a quadratic objective function?. " In *Proceedings of the 28th Design Automation Conference,* pages 427-432, June 1991.

[43] C. Small, " Shrinking devices put the squeeze on system packaging. " *EDN*, vol. 39, no. 4, pages 41-46, Feb. 17, 1994.

[44] Y. Tamiya, Y. Matsunaga and M. Fujita." LP based cell selection with constraints of timing, area and power consumption. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 4378-381, November 1994.

[45] C-H. Tan and J. Allen. " Minimization of power in VLSI circuits using transistor sizing, input ordering and statistical power estimation. " In *Proceedings of the 1994 International Workshop on Low Power Design*, pages 75-80, April 1994.

[46] B. S. Ting and B. N. Tien. " Routing techniques for gate array. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2(1):301-312, January 1983.

[47] R. S. Tsay, E. S. Kuh and C. P. Hsu. " PROUD: A sea-of-gates placement algorithm. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 318-323, November 1988.

[48] R. S. Tsay. " An exact zero-skew clock routing algorithm. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 12(3):242-249, March 1993.

[49] C-Y. Tsui, M. Pedram, and A. M. Despain. " Power efficient technology decomposition and mapping under an extended power consumption model. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(9):1110-1122, September 1994.

[50] A. Tyagi. " Hercules: A power analyzer of MOS VLSI circuits. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 530–533, November 1987.

[51] S. Turgis, N. Azemard and D. Auvergne. " Explicit evaluation of short circuit power dissipation for CMOS logic structures. " In *Proceedings of the 1995 International Symposium on Low Power Design*, pages 129-134, April 1995.

[52] H. Vaishnav and M. Pedram. " PCUBE: A performance driven placement algorithm for low power designs. " In *Proceedings of the European Design Automation Conference,*

pages 72-77, September 1993.

[53] H. Vaishnav and M. Pedram. "Delay optimal partitioning targeting low power VLSI circuits. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, November 1995.

[54] H. Vaishnav. *Optimization of Post-Layout Area, Delay and Power Dissipation.* Ph.D. Dissertation, Computer Engineering, University of Southern California, August 1995.

[55] H. J. M. Veendrick. " Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits. " IEEE *Journal of Solid State Circuits*, 19:468–473, August 1984.

[56] A. Vittal and M. Marek-Sadowska. " Power optimal buffered clock tree design. " In *Proceedings of the 32nd Design Automation Conference*, pages 497-502, June 1995.

[57] A. Vittal and M. Marek-Sadowska. " Power distribution topology design. " In *Proceedings of the 32nd Design Automation Conference*, pages 503-507, June 1995.

[58] J. G. Xi and W-M. Dai. " Buffer insertion and sizing under process variations for low power. " In *Proceedings of the 32nd Design Automation Conference*, pages 491-496, June 1995.

[59] T. Yoshimura and E. S. Kuh. " Efficient algorithms for channel routing. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1(1):25-33, January 1982.

[60] G. Zimmermann. " A new area and shape function estimation technique for VLSI layout. " In *Proceedings of the 25th Design Automation Conference*, pages 60-65, June 1988.

[61] D. Zhou and X. Y. Liu. " Optimal drivers for high speed low power ICs. " *To appear in Int'l Journal of High Speed Electronics and Systems*, 1996.

[62] Q. Zhu, W. M. Dai and J. G. Xi. " Optimal sizing of high speed clock network based on distributed and transmission line models. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 628-633, November 1993.

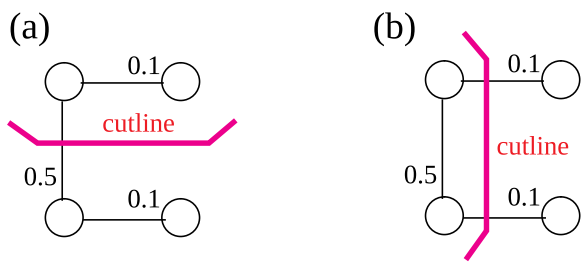

Figure 1 Effect of net activity on partitioning result.

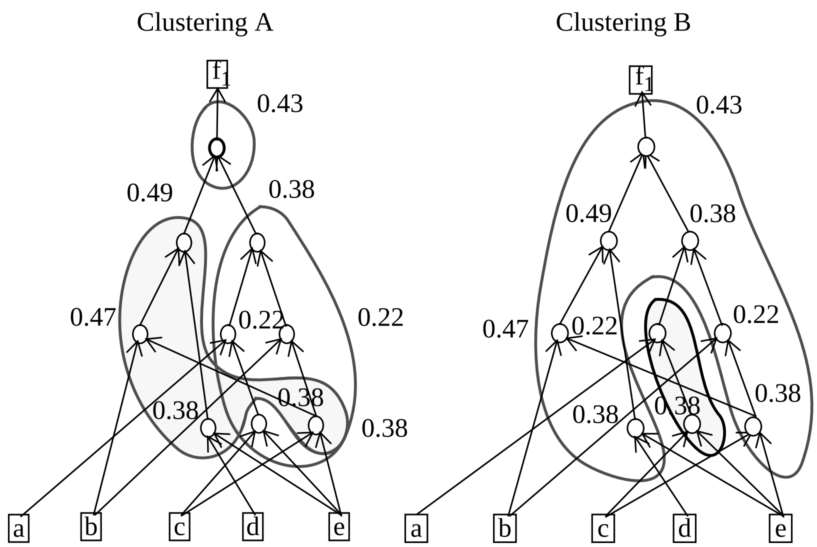

Figure 2 Clustering solutions with equal logic depth but different power.

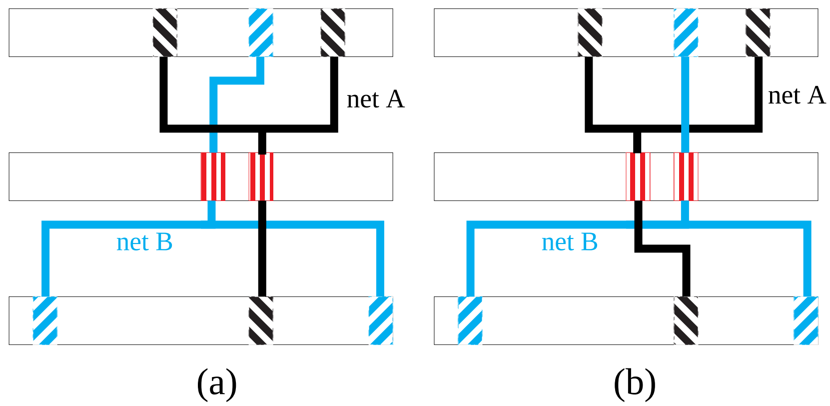

Figure 3 Effect of switching activity on Feedthrough assignment

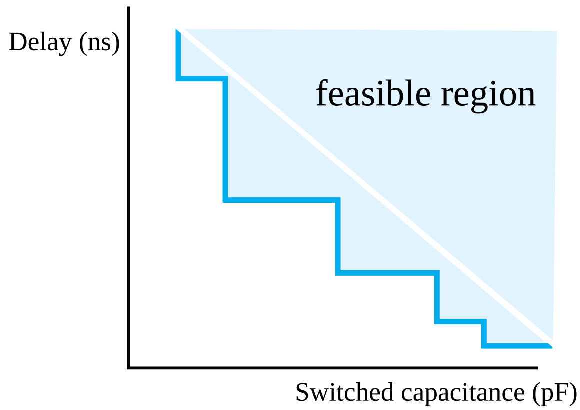

Figure 4 A power-delay tradeoff curve

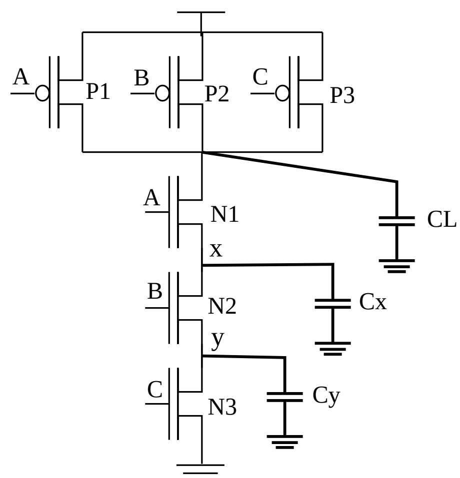

Figure 5 Effect of signal reordering on power dissipation

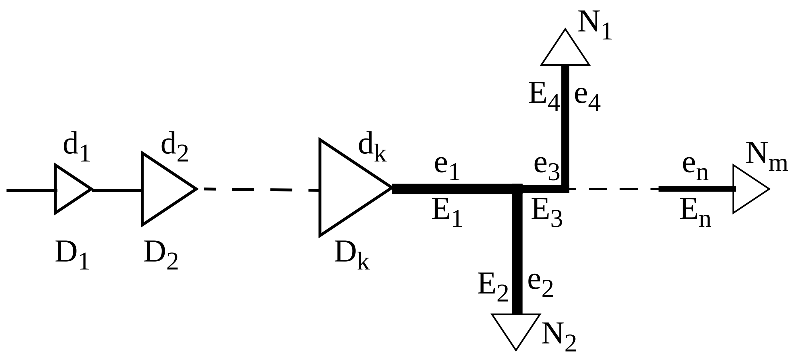

Figure 6 A k-stage cascaded driver driving an tree with m sinkes