Activity-Driven Clock Design for Low Power Circuits *

Gustavo E. Téllez *Amir Farrahi* *Majid Sarrafzadeh*
Department of Electrical Engineering and Computer Science
Northwestern University, Evanston, IL 60208

Abstract

In this paper we investigate *activity-driven clock trees* to reduce the dynamic power consumption of synchronous digital CMOS circuits. Sections of an *activity-driven clock tree* can be turned on/off by gating the clock signals during the active/idle times of the clocked elements. We propose a method of obtaining the switching activity patterns of the clocked circuits during the high level design process. We formulate three novel activity-driven problems. The objective of these problems is to minimize system's dynamic power consumption. We propose an approximation algorithm based on recursive matching to solve the clock tree construction problem. We solve the gate insertion problems with an exact algorithm employing the dynamic programming paradigm. Finally, we present experimental results that verify the effectiveness of our approach. Our work in this paper is a step in understanding how high level decisions (e.g. behavioral design) can affect a low level design (e.g. clock design).

1 Introduction

Modern digital systems are being designed with faster *clock frequencies* which imply increased power consumption and heat dissipation by the system components. Heat dissipation by highly integrated circuits is of concern because virtually all failure mechanisms are boosted at high temperatures [2]. Moreover, with the advent of portable electronics, the design of circuits with low power consumption has gained much importance. Recent work on low power design has focused in the areas of logic and layout synthesis and the area of high-level synthesis. The idea of exploiting sleep mode (or activity-driven) operation to minimize power consumption was proposed recently in [3].
The power consumed by CMOS circuits consists of two components: dynamic and static power. The static power is largely determined by the technology and circuit design. In this paper we only consider minimizing the dynamic power. The average dynamic power consumed by a CMOS gate g with load capacitance C_g is given by $P_{av}(g) = \frac{1}{2} C_g V_{dd}^2 D(g)$, where V_{dd} is the voltage of the power supply and $D(g)$ represents the *transition density* of the signal at the output of g [5]. This suggests that a signal has a high power contribution if it has large load capacitance and/or transition density, which is the case with a clock signal. Recent studies [1] indicate that the clock signals in digital computers consume a large (15%-45%) percentage of the system power. Therefore, to reduce the power consumption of synchronous systems, we would like to minimize the total power consumed by the clock tree subject to performance constraints on the clock signal, such as the operating frequency and maximum clock skew. Most efforts in clock tree power reduction have focused on issues such as reduced voltage swings, buffer insertion and clock routing [4]. In a normal clock tree, the clock signal arrives regularly at all of the clock sinks. However clock signals are not needed when the circuits are idle. Suppose that we know the set of times when the clocked sinks must be active. We refer to the set of active/idle times for the module as *activity patterns*. We obtain the activity patterns

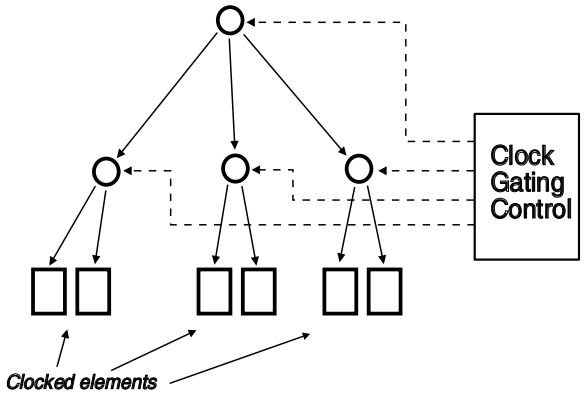

Figure 1: A gated clock tree construction.

*This work was supported in part by NSF grant MIP-9207267 and by the IBM Ph.D. Resident Study Program.

by simulation of the design at the behavioral level using synthetic data (see Section 2). The clock signal must be supplied to the modules only during these active times. If we gate the clock signal such that it is only delivered during these times we may reduce the total power consumed by the clock, and perhaps also by the modules. We call a clock tree thus constructed an *activity-driven clock tree. In this paper we analyze the problems of minimizing the power consumed by a synchronous system by minimizing its activity through the use of an activity-driven clock tree.*

The outline of this paper is as follows. In Section 2, we outline the motivations for this work and show some introductory examples of activity-driven clock trees. Next, in Section 3, we formulate two problems: the activity-driven clock tree construction problem and the minimum-power activity-driven clock gate insertion problem. Sketches of the solutions to the activity-driven problems are given in Section 3. Experimental results and conclusions are shown in Section 4.

2 Motivation

Our first objective is to construct an *activity-driven clock tree* which minimizes power consumption by reducing clock activity. For this purpose, we must have activity patterns for the clocked modules. We now propose a methodology based on high-level synthesis steps that yield these activity patterns:

1. Start from a high-level description of the system. As an example see Figure 2, where we show an HDL description of a differential equation (DE). We then show a transformation of the HDL description to a Control-Data Flow Graph (*CDFG*). Next, schedule and allocate a set of modules into a set of control steps.

2. If a module is assigned to a control step, then the module is active during that control step, otherwise the module is idle. The activity pattern U for a module consists of a set of "1"s and "0"s denoting active and idle times, respectively.

3. The above steps may work well with DSP circuits, where the data activity is well known. In other circuits, such as microprocessors, module activities can be sampled from simulations, which will yield activity probabilities. We can then translate these probabilities into activity patterns.

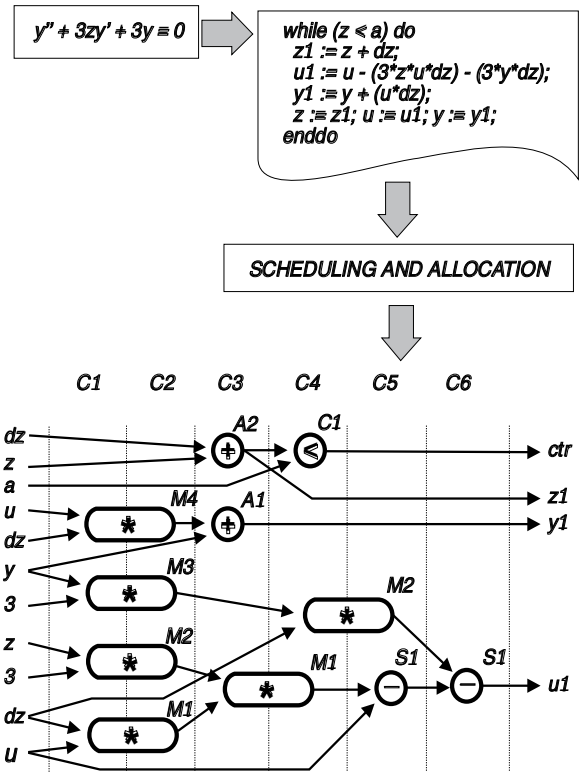

Figure 2: High level design transformation of the DE from HDL to a CDFG (Control-Data Flow Graph).

We will further assume that the modules have registers on their inputs that will have to be clocked. Therefore, if the clock signal is not fed to the inputs of the module's registers, then the outputs of the register will not change and thus the module will not consume dynamic power. Given the activity patterns the proposed gate clock tree construction stage may be executed during two phases in the design:

1. Before floorplanning/placement of the modules has occurred: we then propose to add the resulting gated clock tree to the placement problem.

2. After floorplanning/placement of the modules is completed. At this time, power consumption can be estimated from the placement and routing information. The objective of the clock construction is to come up with a suitable power trade-off, taking wiring geometry into account.

Consider Figure 3, where we illustrate a possible arrangement of a binary clock tree with the clocked modules from the **DE Example**. Each module is drawn according to its activity pattern. The activity pat-

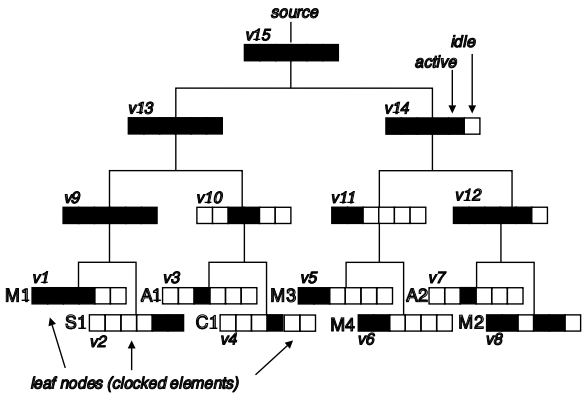

Figure 3: **DE Example**: a clock tree circuit for the modules of the differential equation circuit.

terns are obtained from Figure 2. The tree also contains possible activity patterns of its internal nodes. These activity patterns are derived assuming that a gate can be placed on every node of the clock tree, and they represent the times when these gates must allow propagation of the clock signal. *A gate at the root of any clock tree must be active during a time period if any of its sinks is also active.* Thus, the activity of the internal nodes is obtained by ORing the activity patterns of the sinks that belong to the corresponding sub-tree. Using Figure 3, consider the activity pattern of sub-tree rooted at node v_{12} (111110) which contains modules A2 and M2 as sinks (001000 $\vee$ 110110 = 111110). The tree in Figure 3 has a total of 40 idle time periods This can be improved to 52 idle periods by placing modules of similar activity closely. This heuristic is the basis of our tree construction algorithm.

The total power consumed by circuit clocked by a gated tree results from the several contributions. Power is added by the input and output clock gate capacitance and activity patterns. The control logic associated with the clock gates also adds to the power. The clock tree wiring with its capacitances and activity patterns contributes to the power. Finally, the activity patterns of the clock signals at the sinks dictate an average power consumption of the clocked modules.

Our second task is to locate the clock gates, given a clock tree topology, such that the total power is minimized. The gate insertion problem requires detailed information about the parasitic capacitances of the clock tree and the control lines of the gates, hence we model the clock tree topology using an H-Tree construction.

3 Formulation and Algorithms

We first define an activity pattern for an element i of the system, $U(i) = \{a_{ij} | j = 1, \ldots, u, a_{ij} \in \{0, 1\}\}$, with u time periods, and activity a_{ij}. Let element i consume total *active circuit power* $P_A(i)$, during periods where the circuits are active, and thus the clock must be supplied for proper function. Also let $P_I(i)$ denote the total *inactive or idle circuit power*, whence clock supply is unnecessary. The power consumed by clocked element i is $P(U(i), i) = \sum_{j=1}^{u} [a_{ij} P_A(i) + (1 - a_{ij}) P_I(i)]$.
The clock tree $T(V, E)$, consists of a set of nodes $v_i \in V$, such that the leaf nodes, or sinks, connect to the inputs of the clocked elements. The nodes are connected by set of edges $e_{ij} \in E$, where edge e_{ij} connects node v_i to node v_j. We associate activity patterns with each element of the clock tree: a module i has activity pattern $U_M(i)$, a clock tree edge e_{ij} has an activity pattern $U_T(j)$. The active power of the tree edge is defined by the parasitic capacitance due to its length. A tree edge e_{ij} with no inserted gate has $U_T(i) = U_T(j)$. A gate g added to an edge e_{ij} changes the edge activity pattern to the pattern obtained by ORing the patterns of the leaves. The clock gate g is placed as close to vertex v_i on edge e_{ij} as possible, to maximize the benefit of the gate. The gate g has three power contributions to the tree: the power due to the gate clock input, $P(U_T(i), g)$, the power due to the clock gate output $P(U_T(j), g)$ and the power due to the control logic of the gate. Define the function $t(U(i))$, which measures the changes in the activity pattern, and is used to measure the power consumed by the control signals of the clock gates. The total power of the clock tree is obtained from the sum of the power contributions defined above.

Activity-Driven Clock Tree Construction Problem (ADCTC): Let the activity pattern of a clock tree node be obtained by ORing the patterns of its sinks. Construct a tree $\mathcal{T}(V, E)$ on a set of sinks N such that the weighted sum of nodes activities $\mathcal{A}(\mathcal{T}) = \sum_{v_i \in V} \left[c_i t(U_T(i)) + \sum_{j=1}^{u} k_{ij} a_{ij} \right]$ in the resulting tree is minimized.
The weights k_{ij} and c_i can be derived from the power contributions defined previously. Hence the quantity $\mathcal{A}(\mathcal{T})$ can be tuned to measure or estimate as needed the power consumed by the system. We propose a constant approximation algorithm to solve the ADCTC problem. We use a binary tree for the clock tree due to clock skew concerns. The algorithm, is based on recursive weighted matching, where the matching weight is the value of the objective function of the resulting sub-tree.

Activity-Driven Minimum Power Gate Insertion Problem (ADMPGI): Let a gate g added to an edge e_{ij} of clock tree $\mathcal{T}(V, E)$, induce an activity pattern defined by the ORing of the activity patterns of the sub-tree sinks. Given that each gate added to the clock tree has a power contribution, find the combination of gates which minimizes the total power of the tree.

We propose two algorithms that solve the ADMPGI problem. The algorithms insert gates into the clock tree such that that the total power of the system is minimized. The first algorithm, based on a bottom-up traversal of the clock tree, uses a dynamic programming paradigm to enumerate the least power sub-trees with and wthout clock gates. This algorithm produces solutions of arbitrary skew. The second algorithm, uses the solution to the first algorithm to produce solutions with zero-skew. The idea is to have the same number of buffer/gates in all source to sink paths. The locations of buffer/gates are restricted to a subset of the tree levels. This procedure is repeated for all n combinations of tree levels. Details of these algorithms can be found in [6].

4 Experiments and Conclusions

Due to the novelty of our approach no benchmarks exist that can be used in experiments. For this reason we conducted experiments on activity patterns generated by several methods. Primarily, two techniques were used: for each module randomly set to active k out of a maximum of u time periods, and simulate similar patterns by duplicating d times a given set of activity patterns.

We have observed that the quality of our results changes with the types of patterns and with the percentage of active time periods. The results from Figure 4 show that our choice of matching function is a good simplified clock tree construction objective. The results shown in Figure 4 also indicate that the proposed tree construction and gate insertion algorithms are effective in reducing the dynamic power consumption.

Engineering issues exist with clock trees which must be resolved for practical applications. The addition of gates in the clock paths introduce the possibility of glitches in the clock signal. Also the propagation delays of gates are harder to control, thus introducing unwanted skew.

Several open problems related to activity-driven clock trees also remain: scheduling and allocation that considers activity-driven clocks seems possible, the generation of activity patterns is an open question, specially

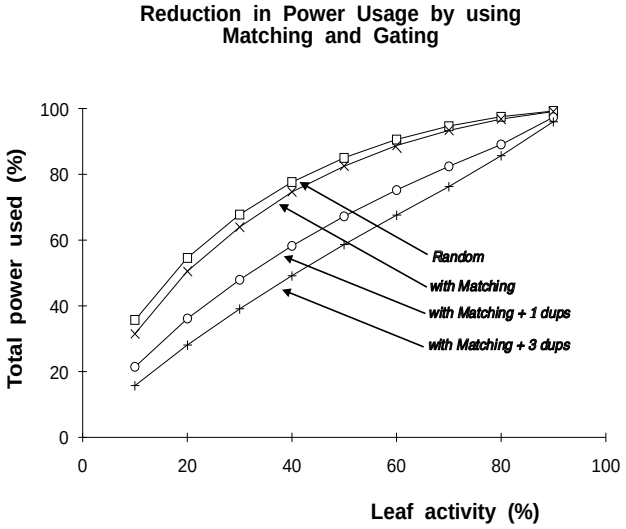

Figure 4: The graph shows the percentage of ungated clock tree power consumed by gated clock trees.

for circuits where the high level behavior is data dependent, finally detail physical design approaches that handle these clock trees is also open.

References

[1] In *International Workshop on Low Power Design*, April 1994.

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

[3] A. H. Farrahi, G. Téllez, and M. Sarrafzadeh. "Memory Segmentation to Exploit Sleep Mode Operation". In *Design Automation Conference*. ACM/IEEE, June 1995.

[4] E. G. Friedman. "Clock Distribution Design in VLSI Circuits - an Overview". In *International Symposium on Circuits and Systems*, pages 1475–1478, May 1993.

[5] F. Najm. "Transition Density: A New Measure of Activity in Digital Circuits". *IEEE Transactions on Computer Aided Design*, 12(2):310–323, 1992.

[6] G. Téllez, A. H. Farrahi, and M. Sarrafzadeh. "Activity-Driven Clock Design for Low Power Circuits". Technical report, Northwestern University, EECS Department, Evanston, IL, November 1994.