Estimation and Bounding of Energy Consumption in Burst-Mode Control Circuits

Peter A. Beerel*
EE-Systems Dept.
USC
Los Angeles, CA 90089

Kenneth Y. Yun
Dept. of ECE
UC San Diego
La Jolla, CA 92093

Steven M. Nowick†
Dept. of CS
Columbia University
New York, NY 10027

Pei-Chuan Yeh
EE-Systems Dept.
USC
Los Angeles, CA 90089

Abstract

This paper describes two techniques to quantify energy consumption of burst-mode asynchronous (clock-less) control circuits. The circuit specifications considered are extended burst-mode specifications, and the implementations are multi-level logic implementations whose outputs are guaranteed to be free of any voltage glitches (hazards). Both techniques use stochastic analysis to combine a small number of simulations in order to quantify average energy per external signal transition. *The first technique uses* N-valued simulation *to derive mathematically tight upper and lower bounds of energy consumption. Using this technique we bound the effect of hazards under all possible operating conditions and environments for a given circuit. Additionally, to drive synthesis tools for low-power, we propose a second technique that uses fixed-delay simulation to derive a realistic estimate of energy consumption within our derived upper and lower bounds. We demonstrate the feasibility of both these techniques on a variety of burst-mode control circuits used in an industrial-quality chip. Our preliminary results indicate that less than 5% of the power of typical multi-level burst-mode circuits can be attributed to hazards.*

1 Introduction

One of the main reasons asynchronous circuits have recently gained interest is their potential for using less energy than their synchronous counterparts. Asynchronous circuits avoid power-expensive global clocks and, if designed well, consume energy only when and where necessary. However, the key words here are *if designed well*. Automated energy consumption estimation has only recently been considered [3, 1], and optimization for energy consumption has been limited to hand-transformations [15]. Hence, the quality of the design is determined purely by the expertise of the designer. Our ultimate goal is to assist less experienced designers with CAD tools for automated energy estimation and low-power synthesis.

In this paper, we restrict our attention to the energy use of asynchronous control circuits. Since these circuits operate independently of a global clock, *energy consumption per clock cycle*, a common means of quantifying energy dissipation in synchronous circuits, can not be used. Because of this, Kudva et al. [3] suggested measuring energy consumption *per output transition*. This was appropriate for their application because control circuits were limited to pre-calculated macro gates whose energy consumption could be pre-measured using SPICE. Beerel *et al.* proposed measuring *energy per external signal transition* for more general circuit structures composed of a netlist of gates, though they only handled speed-independent circuits whose behavior is independent of relative gate delays and in which all circuit nodes are guaranteed to be hazard-free [1].

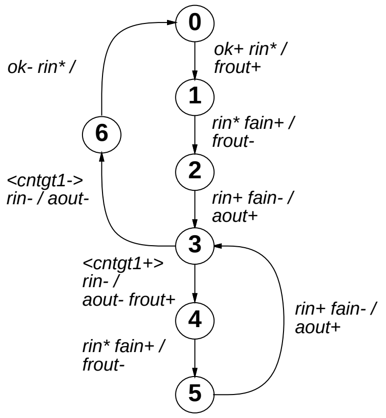

Figure 1: A burst-mode specification of the example *scsi-init-send* .

This paper addresses burst-mode asynchronous circuits which differ from speed-independent circuits in that internal signals may exhibit voltage glitches. In burst-mode circuits only primary outputs are guaranteed to be hazard-free. These circuits are specified with an extended burst-mode (XBM) diagram and are implemented using a modified Huffman architecture. As an example, the extended burst mode specification for *scsi-init-send* is given in Figure 1. This architecture has the advantage of very low latency because the input to output path is purely combinational logic—no explicit storage elements are used. In each state of the machine, the circuit waits for a set of specified input transitions (an input burst), then simultaneously changes a variety of output signals (produces an output burst) and internally sets any number of state signals. It is important to note that from a given state a variety

*This research was funded in part by a Charles Lee Powell Foundation Research Equipment Grant for Young Faculty of the USC School of Engineering and a NSF Career Grant.

†This research was funded in part by the National Science Foundation MIP-9308810 and by a grant from IBM Research.

of different input bursts may occur, driving the circuit into different modes of activity. For example, in the *scsi-init-send* example depicted in Figure 1, there are two branches from state 3. The first, to state 4 represents the sending of another byte, while the second, to state 6 represents the signaling of the end of transmission. Using this design style, SCSI interfaces, cache controllers, an infrared communications controller, as well as a variety of interface specifications have been designed with promising results [9, 12, 17, 16, 18, 7].

This paper provides efficient algorithms for quantifying the energy consumption of burst-mode circuits implemented with two-level or multi-level logic. Since different modes of circuit operation may consume different amounts of energy, we must determine the relative likelihood of executing different modes. To do this, we assume the availability of branch statistics which describe the relative probabilities of different burst-mode branches. In practice, these branch statistics may be given by the user or estimated through behavioral simulation of the circuit in its environment. In the *scsi-init-send*, the branch probabilities depend on the packet sizes of SCSI transfers. For example, a reasonable packet size is 1K bytes, in which case the branch probabilities would be as follows: state 3 to state 4: 0.999 and state 3 to state 6: 0.001.

Our energy estimation techniques follow a two step approach. First, we quantify the energy consumed for each specified burst in the XBM diagram. Second, using the quantified energies per burst, given branch statistics, and Markov chain analysis, we quantify the energy per average external signal transitions.

Energy consumption is quantified in two ways. First, N-valued simulation is used to provide mathematically tight upper and lower bounds on overall energy consumption assuming the unbounded (time-varying) gate and wire delay model. This bounds the potential effects of hazards on switching activity over all possible operating conditions and process variations. Second, fixed-delay simulation is performed to provide a reasonable estimate of power consumption that may be used to guide low-power synthesis and choices between circuits designed with different asynchronous methodologies.

The organization of the paper is as follows. Section 2 discusses the differences in calculating energy in asynchronous circuits versus synchronous circuits, thereby motivating our two-step approach to energy estimation. Section 3 describes the classes of burst-mode circuits and their specifications. Section 4 presents our techniques to quantify energy consumed during a given simple burst. Section 5 describes our stochastic analysis to combine the measured and estimated energy per bursts into single figures of merit. Section 6 extends the techniques to extended burst-mode circuits. Section 7 relates our work to that in synchronous switching estimation. Section 8 describes some preliminary results and conclusions.

2 Energy estimation in asynchronous versus synchronous circuits

2.1 *Motivation*

The lack of a global clock in asynchronous circuits creates circuit and architectural characteristics which motivate our two-step approach to estimating energy consumption in asynchronous circuits:

- **Control circuits drive registers:** Because asynchronous control circuits are responsible for controlling and driving all register enable lines (in lieu of a global clock), certain outputs will have large capacitive loads, implying that these control circuits may very well consume a larger percentage of total chip energy as compared to synchronous control circuits. Moreover, since the gate capacitances within an asynchronous circuit can vary dramatically, accurate switching estimates for each gate may be more important in asynchronous circuits than in synchronous ones.
- **Distributed control:** Control circuits in asynchronous circuits tend to be more numerous and on average smaller than those in synchronous circuits. In our experience, most practical burst-mode specifications are less than 100 specified bursts and circuit sizes of less than 1K gates. Consequently, it is computationally feasible to use a more complex power-estimation procedure that is more accurate.

Most synchronous switching-estimation procedures (e.g., [5, 14, 8]) assume given switching probabilities on inputs that are symbolically propagated through all circuit nodes to the outputs. This methodology has low computation complexity, which is important since many synchronous circuits have more than 10K gates. However, it is not obvious how to derive signal probabilities and correlations from asynchronous circuit specifications. Moreover, few tools based on this approach can give accurate switching estimates on individual gates; they are much better at providing good total energy estimates. This suggests that for asynchronous circuits, naive probabilistic approaches may have lower accuracy. On the other hand, extensive simulation of an asynchronous circuit in its environment is still computationally very expensive and unnecessary. We propose a two-step approach in which an efficient simulation-based algorithm bounds energy per burst from which bounds on total average energy consumed can be derived using given branch probabilities and stochastic analysis.

2.2 *Average energy per external signal transition*

To define energy per external signal transition for a circuit, the following definitions are needed: a *trace* T of the circuit is a sequence of specified signal transitions $\langle t_1, \ldots, t_n \rangle$; $\mathcal{T}$ is the set of all traces that start at the initial state and are specified to occur; and, $\mathcal{T}^k$ is a set of all traces in $\mathcal{T}$ that have k external signal transitions. For example, in the *scsi-init-send* example, *rin** is a directed don't care signal which may or may not rise in state 0 (described more formally below). As a result, $\mathcal{T}^3 = \{ < ok^+, rin^+, frout^+ >, < rin^+, ok^+, frout^+ >, < ok^+, frout^+, rin^+ >, < ok^+, frout^+, fain^+ > \}$.

For simplicity, we assume energy consumption depends only on the capacitance at the output of the gates that switch during the execution of T. The well-known formula for energy consumption is:

$$En(T) = \sum_{\text{gates}} \frac{1}{2} \cdot C_{\text{gate-load}} \cdot Vdd^2 \cdot (\text{\# of gate switches}). \quad (1)$$

A more accurate model of $En(T)$ would consider the energy consumed through charging/discharging of capacitance that is internal to the gates along with other factors, such as short-circuit current.

Furthermore, we associate a probability for each trace in $T \in \mathcal{T}^k$, denoted $Pr_{XBM}(T)$, subject to the constraint that

$\sum_{T \in \mathcal{T}^k} Pr_{XBM}(T) = 1$ for all k. These probabilities depends on gate delays and the probability of various environmental choices. For example, consider the four traces in $\mathcal{T}^3$. While the sum of their probabilities should equal 1, their exact probabilities, however, depend on the relative delays of the circuit and the environment. If the environment is known to generate new requests (raise *rin*) relatively slowly, a delay analysis might generate 0.7 for the probability of the fourth trace $< ok^+, frout^+, fain^+ >$ and 0.1 for each of the other three traces. These probabilities depend on gate delays and the probabilities ov various environmental choices.

Finally, we present our definition for average energy per external signal transition, denoted $\overline{E}$:

$$\overline{E} = \lim_{k \to \infty} \frac{\sum_{T \in \mathcal{T}^k} En(T) \cdot Pr_{XBM}(T)}{k}. \quad (2)$$

3 Specifications and circuit implementations

In this paper we restrict ourselves to specifications that are extended burst-mode diagrams and hazard-free multi-level logic implementations, as defined in this section.

3.1 Extended Burst-Mode Specifications

An extended burst-mode asynchronous finite state machine [18] is specified by a state diagram which consists of a finite number of states, a set of labeled state transitions connecting pairs of states, and a start state. (These specifications extend the original burst-mode format introduced in [11].)

Figure 1 shows an example of the extended burst-mode specification. Signals not enclosed in angle brackets and ending with $+$ or $-$ are *terminating edge signals*. The signals enclosed in angle brackets are *conditionals*, which are level signals whose values are sampled when all of the terminating edges associated with them have occurred. A conditional $\langle a^+ \rangle$ denotes "if a is high" and $\langle a^- \rangle$ denotes "if a is low." A state transition occurs only if all of the conditions are met and all the terminating edges have appeared.

A signal ending with an asterisk is a *directed don't care.* If a is a directed don't care, there must be a sequence of state transitions in the machine labeled with a^*. If a state transition is labeled with a^*, the following state transitions in the machine must be labeled with a^* or with a^+ or a^- (the terminating edge for the directed don't care). Figure 1 describes a state machine having a conditional input *cntgt1*, 3 edge inputs (*ok*, *rin*, *fain*), and 2 outputs (*aout*, *frout*). Consider the state transitions out of state 3. The behavior of the machine at this point is: "if *cntgt1* is low when *rin* falls, change the current state from 3 to 6 and lower the output *aout*; if *cntgt1* is high when *rin* falls, change the current state from 3 to 4 and lower *aout* and raise *frout*."

A directed don't care may change at most once during a sequence of state transitions which it labels. That is, directed don't cares are *monotonic* signals, and, if one does not change during this sequence, it must change in the state transition labeled by the terminating edge. In figure 1, rin is low when the specification is in state 4. It can rise at any point as the machine moves through state 5 but it must have risen by the time the machine moves to state 3 because the terminating edge rin^+ appears between states 5 and 3.

The input signals are globally partitioned into level signals (conditionals), which can never be used as edge signals, and edge signals (terminating or directed don't care), which can never be used as level signals. If a level signal is not mentioned on a particular state transition, it may change freely. If an edge signal is not mentioned, it is not allowed to change.

3.2 3-D Machines

In this paper, we assume that XBM specifications are implemented in *3-D* design style [17] (for an alternative burst-mode style, see [9]). A 3-D machine is formally represented as a 4-tuple (X, Y, Z, δ), where X is a set of primary input symbols, Y a set of primary output symbols, Z a possibly empty set of state variable symbols, and $\delta : X \times Y \times Z \to Y \times Z$ is a *next-state function*. The hardware implementation of a 3-D machine is a combinational network, which implements the next-state function, with the outputs of the network fed back as inputs to the network. There are no explicit storage elements such as latches, flip-flops or C-elements in a 3-D machine.

3.3 Multi-level implementations

Multi-level implementations of the combinational network implementing the next-state functions can be derived from a BDD-based synthesis [18], or from hazard-free two-level circuits [12] using hazard-non-increasing transformations [4]. In both cases, the combinational circuits resulting from cutting feedback paths are guaranteed to be hazard-free *at the outputs* under arbitrary wire delay assumption.

Multi-level logic may amplify the potential effect of hazards on energy consumption. In the presence of widely varying path lengths through the combinational logic an internal signal can glitch multiple times, consuming a significant fraction of energy. This is a well-known problem within many implementation of combinational multipliers [2], suggesting that the analysis of hazards in multi-level burst-mode circuits is worthwhile.

4 Quantifying energy per burst

This first half of our energy estimation procedures analyzes the energy consumption of each input/output burst in isolation. We first present an algorithm to find mathematically tight bounds of energy consumed and then a methodology to provide an estimate on energy consumed. The estimate indicates where between the upper and lower bounds the actual energy consumption will be.

4.1 Bounding energy per burst

To mathematically bound the switching activity of a burst, an underlying delay model much be chosen. This delay model should be pessimistic which can take into consideration all possible changes in operating and process conditions. Note that a fixed-delay model would not be a good choice, since small variations in fixed delays could lead to significant differences in switching activity. We propose, instead, the unbounded gate and wire delay model in which wires can have arbitrary (but finite) and time-varying delays. Although this model may be unrealistically pessimistic, our experimental results suggest that the difference in upper and lower bounds is often a very small percentage of the total energy consumed. Hence, using this model we can provide solid evidence that hazards do not significantly contribute to the energy consumption of burst-mode circuits.

To bound the energy per burst under the unbounded gate and wire delay model, we introduce a new method called *N-valued*

Algorithm 4.1 (N-valued simulation)
/* $G = (V, E)$ *is the graph of the circuit*
T = *topological sort of gates from inputs to outputs*
i = *a given burst (e.g,* $i = a^+ b^+ c^+$*)*
n^{max} = *a vector containing all upper bounds*
$n^{max}[u]$ = *upper bound on # times signal u switches*
n^{min} = *a vector containing all lower bounds*
$n^{min}[u]$ = *lower bound on # times signal u switches*
v^{new} = *a vector for each new value*
$v^{new}[u]$ = *value of signal* u *after burst occurs and circuit settles*
v^{old} = *a vector of all old values*
$v^{old}[u]$ = *value of signal* u *before burst occurs* */

```
N-value-simulation(T, i) {
Set v^new[u] and v^old[u] for all signals u in burst i
For each gate u in topological order T {
  n^max[u] =
    ub_#_gate_switches(n^max, v^old, v^new, u)
  n^min[u] =
    lb_#_gate_switches(n^min, n^max, v^old, v^new, u) }}
```

Figure 2: N-valued simulation to bound switching activity during a given burst

simulation, which is very similar to 9-valued simulation described by Kung [4]. Like 9-valued simulation, our N-valued simulation traverses the circuit topologically and propagates the number of transitions possible at each gate. However, unlike 9-valued simulation, N-valued simulation can keep track of an arbitrary number of glitches at each node. Moreover, instead of managing memory-intensive look up tables, we implement each step in the N-valued simulation using a simple decision procedure based on algebraic equations. Finally, we have adapted our N-valued simulation to keep both upper and lower bounds of switching activity at each node. The high-level description of the algorithm is given in Figure 2. There are two key functions in the algorithm which return lower and upper bounds on the switching activity of a gate given bounds on the switching of the gates inputs. These functions are described in more detail below.

Note that for simplicity we assume that each burst consist of transitions of external signals that either remain constant or switch monotonically. We will extend its application to DDC bursts and conditional signals in Section 6.

4.2 Lower bounding gate switches

A *lower bound* on the number of times a gate switches is any number which is less than or equal to the minimum number of times the gate can switch under the unbounded gate and wire delay model. A lower bound on the number of switches for a gate u that should execute a 0-0 or 1-1 transition (i.e., $v^{old}[u] = v^{new}[u]$) is zero. Moreover, a lower bound on a gate that should execute a 0-1 or 1-0 transition (i.e., $v^{old}[u] \neq v^{new}[u]$) is one. Since we presume the gate is initially at $v^{old}[u]$ and we simulate the burst until the internal signals in the circuit stabilizes, the gate has to switch at least once.

We now prove that there always exists a set of wire and gate delays that demonstrates this scenario. In other words, we prove that this lower bound is also a *tight lower bound*.

Theorem 1 *Under the unbounded gate and wire delay model, a tight lower bound for the switching activity of a signal* u *in a burst b, with initial value* $v^{old}[u]$ *and new value* $v^{new}[u]$ *is given as follows:*

$$lb\_\#\_gate\_switches(n^{min}, n^{max}, v^{old}, v^{new}, u) = \mid v^{new}[u] - v^{old}[u] \mid . \quad (3)$$

Proof: We use induction on the gate level, to show that delays can always be selected to meet the lower bound.

Base case (level = 0): It is assumed that primary inputs change monotonically as specified during the burst.

Inductive hypothesis (level = n): Presume all inputs to all gates at level n or below change only when they are supposed to and, if so, monotonically.

Inductive step (level = $n + 1$*):* Without loss of generality, consider an AND gate which has level equal to n+1. All inputs to this gate must have level n or below. We will show by proper setting of the input wire delays to this gate the given lower bound can be met. The analysis for other basic gates is similar.

Case 1: 1-1 transition: By the inductive hypothesis, all inputs must be at a 1 and must not change. Hence, the AND gate will stay at a 1.

Case 2: 0-1 transition: By the inductive hypothesis, either an input must be at a 1 without changing, or it must monotonically change to 1. The result is a monotonic change of the AND gate.

Case 3: 0-0 transition: By the inductive hypothesis, either one input is at a 0 and should remain at 0, or some input must monotonically change to 0. In the first case, we are assured the AND gate will remain at 0. In the second case, we can set the input wire delays of the gate such that the falling input is the first input to switch, ensuring that the AND gate remains at 0.

Case 4: 1-0 transition: By the inductive hypothesis, some input must monotonically change to 0. We set the input wire delays of the gate such that this input is the first to switch, ensuring that the AND gate can monotonically change to 0.

Since each gate has an independent set of input wires whose delays can be arbitrarily set, all gates at level $n + 1$ can simultaneously meet their given lower bound on switching activity, completing the proof. ∎

This theorem states that the calculation of the lower bound on switching activity can be derived by a simple static analysis of the circuit in which initial and final values of u are compared.

Notice that the order in which transitions in the burst arrive is not specified and is not assumed during this analysis. Indeed, due to the arbitrary wire delays at the inputs of gates this lower bound can be achieved for any transition arrival schedule.

4.3 Upper bounding gate switches

To explain the upper bound for switching activity, we describe the case of an AND gate. The analysis for other basic gates is very similar.

Let an AND gate u have non-monotonic inputs $v \in FI(u)$, where each input can switch $n^{max}[v]$ times. To obtain the tight upper bound on number of gate switches we count the possible 1-0-1 glitches and then add any possible leading 0-1 or trailing

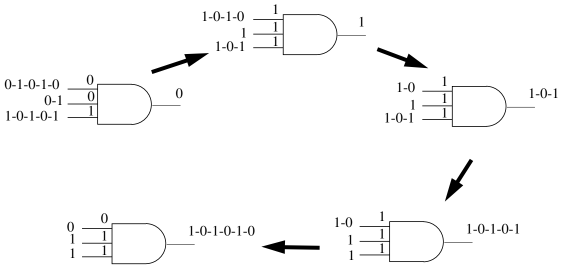

Figure 3: Illustration of worst-case scenario of glitch propagation in an AND gate.

1-0 transition. This method of analysis leads to a simple closed formula.

The maximum number of 1-0-1 glitches feasible at the output of an AND gate is the sum of possible 1-0-1 glitches at the inputs to the gate. This is depicted in the worst-case scenario shown in Figure 3. To obtain a 1-0-1 output glitch of the AND gate, all its inputs must be 1 or switched to 1 using a feasible transition. If an input is not 1 and cannot be driven to 1, then no 1-0-1 glitches are feasible.

Then, the input transitions are skewed such that each input executes all its feasible 1-0-1 glitches one at a time. Since input wire delays are arbitrary and time-varying, by setting all other input wire delays to be much longer, this is a feasible scenario under the unbounded gate and wire delay model. Executing all 1-0-1 glitches on the input enables the output to execute matching 1-0-1 glitches. This process is repeated until all inputs exhaust their 1-0-1 glitches.

Because only input wire delays need to be set appropriately to ensure this feasibility of this scenario, all gates in the circuit can simultaneously exhibit this scenario independent of the arrival order of the burst transitions.

We now quantify this analysis. The maximum number of 1-0-1 glitches on a particular fanin $w \in FI(u)$ is given by the following formula:

$$\text{num\_1-0-1\_glitches}(n^{max}[w],\, v^{new}[w], v^{old}[w]) = \text{num\_ones}(n^{max}[w],\, v^{new}[w], v^{old}[w]) - 1 \qquad (4)$$

$$\text{num\_ones}(n^{max}[w],\, v^{new}[w], v^{old}[w]) = \frac{v^{old}[w] + n^{max}[w] + v^{new}[w]}{2}. \qquad (5)$$

Function num_ones($n^{max}[w]$, $v^{new}[w]$,$v^{old}[w]$) gives the maximum number of ones in the string of transitions at w can exhibit. For example, if w can exhibit a 0-1-0-1-0-1 glitch, then the number of ones in the glitch is 3. This number can be computed by examining the initial and final value of u in conjunction with the number of transition u can exhibit. There are four cases to consider depending on the initial and the final value of u: (i) 0-0 transition: n even, contains $n/2$ ones; (ii) 0-1 transition: n odd, contains $(n+1)/2$ ones; (iii) 1-0 transition: n odd, contains $(n+1)/2$ ones; and (iv) 1-1 transition: n even, contains $(n+2)/2$ ones.

It can be easily verified that in each of the four cases the function num_ones($n^{max}[w]$, $v^{new}[w]$,$v^{old}[w]$) returns the desired result. The function num_1-0-1_glitches($n^{max}[w]$, $v^{new}[w]$,$v^{old}[w]$) returns one less than num_ones($n^{max}[w]$, $v^{new}[w]$,$v^{old}[w]$) since the number of 1-0-1 glitches on u is exactly one less than the number of 1's in u. For example, the glitch 0-1-0-1-0-1 contains two 1-0-1 glitches which share the middle 1.

Combining the 1-0-1 glitches of all inputs to gate u, we get a total number of 1-0-1 glitches that the gate u can execute

$$\text{tot\_1-0-1\_glitches(u)} = \sum_{w \in FI(u)} \text{num\_1-0-1\_glitches}(n^{max}[w],\, v^{new}[w], v^{old}[w]). \qquad (6)$$

Each 1-0-1 glitch of a fanin w of u represents two feasible output transitions on u. The remaining feasible transitions on u include a 0-1 transition in the case that the gate was initially 0 and a 1-0 trailing transition in the case u is supposed to change to 0. Summing up, we can obtain an equation for the maximum number of output switches:

$$\text{ub\_\#\_gate\_switches}(n^{max}, v^{old}, v^{new}, \text{u}) = \begin{cases} 0, \text{ if some input stays } 0 \\ \overline{v^{old}[u]} + 2 \cdot \text{tot\_1-0-1\_glitches(u)} + \overline{v^{new}[u]}, \text{ otherwise} \end{cases} \qquad (7)$$

Notice how the logical inverses of $v^{old}[u]$ and $v^{new}[u]$ quantify the existence of leading and trailing transitions.

The above analysis demonstrates that the number of glitches given by Equation 7 is feasible under the unbounded gate and wire delay assumption. It is easy to see that Equation 7 also provides an upper bound on the number of possible transitions. Hence, we have the following theorem:

Theorem 2 *Under the unbounded gate and wire delay assumption, the value given by Equation 7 is a tight upper bound on the number of switches of gate u during a given burst.*

The complexity of N-valued simulation is linear with respect to the size of the circuit. Because such simulation is performed only once per burst, the computational cost is still quite manageable for most circuits of interest.

4.4 Estimating switching activity

Although the above bounds are tight in a mathematical sense, there is no guarantee they will be close to measured results. This seemingly contradictory statement is due to the fact that the underlying unbounded gate and wire delay model is overly conservative. While it is important to analytically bound the effects of hazards, it is equally important to provide a realistic estimate of the effect of hazards.

Therefore, we propose to use *fixed-delay simulation* with measured or estimated gate and wire delays to estimate the actual switching activity. Of course, to perform such a simulation it is necessary to assume a schedule of arrival of transitions within a burst. When such information is available and used, the estimate will obviously be more accurate. Unfortunately, without analyzing the environment of the circuit, the arrival schedule of burst transitions may be difficult to estimate. Hence, in such cases, we propose to arbitrarily pick an arrival schedule. Notice that the complexity of fixed-delay simulation is linear with respect to the size of the circuit.

5 QUANTIFYING AVERAGE ENERGY

This section describes the Markov chain analysis used to derive a small set of equations which bounds average energy per burst. These equations are based on given branch statistics and calculated energies per burst.

To model the execution of the circuit, we define a stochastic process $\{B_n, n = 0, 1, 2, \ldots\}$. We say $B_n = i$ if the n^{th} transition of a trace T is a part of burst i. Any trace T defines a sequence of n states in which the n^{th} state is i if the n^{th} transition of is part of the i^{th} burst. For example, if from the initial state of the circuit, the input burst $i = a^+b^+c^+$ must occur followed by the output burst $j = x^+$, then the sequence of process states would be $< i, i, i, j, \ldots >$.

The *long term proportion* of a burst i, denoted Π_i, is the long term proportion of states that the stochastic process is in state i:

$$\Pi_i = \lim_{k \to \infty} \frac{\sum_{\mathrm{T} \in \mathcal{T}^k} (\text{\# times trace T visits state i}) \cdot Pr_{XBM}(T)}{\mathrm{k}}. \tag{8}$$

Then, taking into consideration upper and lower bounds for energy per burst and the number of signal transitions per burst, it is easy to see that the overall lower, estimated, and upper bounds for the average energy consumption can be calculated as follows:

$$\overline{E}_{min} = \sum_{i \in bursts} \frac{\Pi_i \cdot En^{min}(i)}{|b_i|}, \tag{9}$$

$$\overline{E}_{est} = \sum_{i \in bursts} \frac{\Pi_i \cdot En^{est}(i)}{|b_i|}, \text{ and} \tag{10}$$

$$\overline{E}_{max} = \sum_{i \in bursts} \frac{\Pi_i \cdot En^{max}(i)}{|b_i|}, \tag{11}$$

where $|b_i|$ denotes the number of signal transitions in the i^{th} burst. It is possible to show that under the unbounded gate and wire delay model, these upper and lower bounds are mathematically tight, since the wire delays can be time-varying. The rationale behind this result is that, at the beginning of each burst, the delays can feasibly switch to whatever values necessary to ensure that the upper and lower energy bounds per burst are met.

Markov chain theory can be used to obtain the long-term proportions of bursts. We show how the stochastic process B_n can be modified to form a Markov chain from which we can derive long-term proportion of bursts.

First, we review the definition of a Markov chain. Consider a stochastic process $\{X_n, n = 0, 1, 2, \ldots\}$. If $X_n = i$ then we say the process is in state i at time n. If the conditional distribution of any future state X_{n+1} is independent of the past states and depends only on the present state, then the process is a Markov chain.

Consider the initial finite-state stochastic process $\{B_n, n = 0, 1, 2, \ldots\}$ that models the sequence of bursts visited during a trace of the circuit. Each process state B_n can take on a *finite* set of values, the set of bursts. If $B_n = i$, then the n^{th} process state is the burst i. Unfortunately, the conditional distribution of the future state depends on past states. For example, assume the process is in state $B_2 = i$ in which 2 of 3 transitions of burst $i = a^+b^+c^+$ have fired. Then, the next state of process, B_3, must also equal state i. After entering B_3, all 3 transitions of burst i must have fired. Hence, the next process state, B_4, must have a different value j associated to one of possibly many next bursts specified to occur. Hence, the conditional probabilities of the next state depend on a number of past states.

To overcome this problem, we use a well-known technique to form a new stochastic process B'_n in which we use a different (larger) set of states [13]. For each burst i, B'_n contains multiple states with the following semantics: exactly one signal transition of burst i has fired (denoted state i_1); exactly two signal transitions of burst i have fired (denoted state i_2); and so on.

For each burst i, a state i_j is created for the accumulative number of transitions j that have fired in the burst. For example, the burst $i = a^+b^+c^+$ has three associated states. The first state i_1 represents the state in which exactly one of a^+, b^+, or c^+ has fired. The second state i_2 represents the state in which two of three transitions have fired. This third state i_3 represents the state in which all three transitions have fired. Assuming that there is no correlation between branch decisions, the conditional distribution of any future state B'_{n+1} now depends only on the present state. For example, for the burst $a^+b^+c^+$, the probability of going from state i_1 to i_2, denoted $P_{i_1 i_2}$, is 1, while the probability of going from state i_1 to i_3, denoted $P_{i_1 i_3}$, is 0. For any pair of states i_j and k_l the conditional probability is

- one if $k = i$ and $l = j + 1$,
- the probability of a state transition from burst i to burst j, if $j = |b_i|$ and $l = 1$,
- zero otherwise.

Notice that the only non-binary conditional probabilities can occur between output bursts and input bursts at branch points in the XBM diagram. In conclusion, the stochastic process B'_n is a finite-state Markov chain. In Section 7, we suggest that for burst-mode circuits, correlations among branch decisions do not affect the energy consumption of the circuit.

Before deriving the final equations, however, we must show that the Markov chain B'_n satisfies a few properties. A Markov chain is *irreducible* if every state can be reached from every other state. Since, the XBM is assumed to be strongly-connected, from every state there exists a sequence of bursts which enables every burst, hence B'_n is irreducible.

State t in a Markov chain is *periodic* with *period* d if the probability of being in state t after n transitions starting in state t is 0 whenever n is not divisible by d and d is the largest integer with this property [13]. In other words, starting in state t, it may be only possible to enter state t at times 2, 4, 6, 8, etc, in which case state t will have period 2. State t is called *aperiodic* if d equals 1. Since a XBM transition t cannot fire twice without firing a complete cycle of bursts in the XBM diagram, B'_n is periodic.

Because B'_n is a finite-state Markov chain that is both irredundant and periodic, it can be shown that the Π_{i_j} are the unique non-negative solutions to the following equations [13]:

$$\Pi_{i_j} = \sum_{k_l \in V'} \Pi_{k_l} \cdot P_{k_l i_j}, \text{ for all } i_j \in V' \tag{12}$$

$$\sum_{i_j \in V'} \Pi_{i_j} = 1 \tag{13}$$

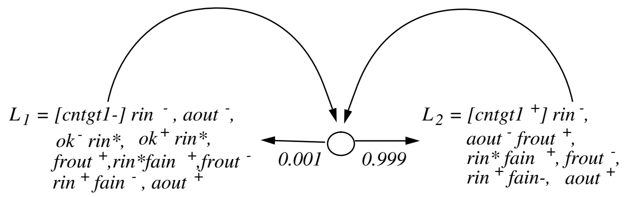

Figure 4: The collapsed XBM diagram for the circuit *scsi-init-send*.

The long term proportion of burst i is then simply the sum of the long term proportion of its sub-states:

$$\Pi_i = \sum_j \Pi_{i_j} \tag{14}$$

Because most the states have a single predecessor, many of the above equations are trivial. First, all intra-burst process states $i_j, j > 1$ have only one predecessor. Second, many inter-burst process states also have only one predecessor. Hence, many states will have the same long term proportion and can be combined into a *burst list* L_m. The different lists β form a partition of the bursts. For the XBM diagram depicted in The *scsi-init-send* XBM diagram depicted Figure 1, the bursts collapse into two burst lists as illustrated in Figure 4.

The new system of linear equations based on this partition of bursts are as follows:

$$\Pi_{L_m} = \sum_{L_l \in \beta} \Pi_{L_l} \cdot P_{L_l L_m}, \quad L_m \in \beta, \tag{15}$$

$$\sum_{L_m \in \beta} (\text{number of transitions in } L_m) \cdot \Pi_{L_m} = 1. \tag{16}$$

In the *scsi-init-send* example, this reduces the number of linear equations from 16 to 3.

The bounds and estimates on average energy can then be calculated in terms of these burst lists as follows:

$$\begin{aligned} \overline{E}_{max} &= \sum_{L_m \in \beta} \Pi_{L_m} \cdot En^{max}(L_m), & (17)\\ \overline{E}_{min} &= \sum_{L_m \in \beta} \Pi_{L_m} \cdot En^{min}(L_m), & (18)\\ \overline{E}_{est} &= \sum_{L_m \in \beta} \Pi_{L_m} \cdot En^{est}(L_m), & (19) \end{aligned}$$

Here, $En^{max}(L_m)$, $En^{min}(L_m)$, and $En^{est}(L_m)$ are the maximum, minimum, and estimated energy consumed in the burst list L_m. They are respectively equal to the sum of the maximum, minimum, and estimated energy consumed in the composite bursts. $|L_m|$ is the number of signal transitions in burst list L_m.

Finally, we can bound the actual average energy per signal transition as follows:

$$\overline{E}_{min} \geq \overline{E} \geq \overline{E}_{max}. \tag{20}$$

6 Extended burst-mode circuits

This section extends our energy analysis to the extended burst-mode features of directed don't cares (DDCs) and conditional signals.

A signal specified as a DDC changes at most once during a sequence of state transitions it labels. This sequence of state transitions may span more than one burst list. However, the signal can change in only one of the burst lists (for this sequence of state transitions). We calculate the minimum/maximum energy of each burst list for each possible location of a DDC signal change (including the case that it does not need to change). Using the minimum (maximum) of these minimums (maximum) yields a minimum (maximum) bound of energy consumption. While these bounds are not mathematically tight, our results suggest that the difference is not significant in practice.

Handling conditional signals is a bit more involved. Because conditional signals are excluded from the set of specified signal transitions, their transitions are not modeled in the Markov process. Consequently, the energy consumed as a result of switching conditional signals is reflected as an increase in the average energy per signal transition. If we assumed that the number of times a conditional signal can switch in every burst-mode state is infinite, the tight upper bound on energy consumption would most likely be infinity. In practice, however, information concerning the behavior of these signals is known. Hence, it is reasonable to assume the circuit designer can provide bounds on their switching in each burst-mode state. For example, in the *scsi-init-send* circuit, the *cntgt1* signal is an output of a counter that counts the number of bytes sent. Hence, it is reasonable to assume that the *cntgt1* conditional signal can only change a small number of times (e.g., four) and only in burst mode state 3, the state in which it is set up.

Given minimum, maximum and estimated bounds on conditional signals, the techniques to bound average energy can trivially be extended to handle transitions on conditional signals.

7 Relationship to other work

The goal of current research is to obtain the most accurate estimate of switching activity while maintaining low complexity. Current research in the synchronous energy estimation targets incorporating different correlations among different circuits signals into the energy estimation procedure. Correlations among signals can be related temporally (on the same signal at different times) [2], spatially (between different signals at the same time), and spatio-temporal correlations (different signals at different times) [5, 14, 8, 6].

In our circuits, the only correlations that are not taken into consideration by our Markov model are correlations between different branch decisions. Correlations among the branch decisions affect the probability of each burst list. We believe, however, that they do not affect the expected number of occurrences of a particular burst list in long sequence of burst lists. Since the energy consumed in a circuit depends only on the long term proportion of each burst list, we conjecture that correlations do not affect average energy.

8 Results and Conclusions

Circuit	# nodes	# signal transitions	# burst lists	$\overline{E}_{min}$	$\overline{E}_{max}$
				(pJ.)	
pipeline ctrl	3	9	1	2.81	2.89
scsi-init-send	7	22	2	1.95	2.03
binary-counter	32	80	1	6.25	6.25
buf-ctrl	3	10	1	2.13	2.19
cache-ctrl	38	359	18	13.44	13.58
dramc	12	53	3	3.07	3.07
pe-send-ifc	11	52	8	4.15	4.16
q42	4	14	1	2.77	2.77

Table 1: Calculated average energy bounds.

We have automated our upper/lower bounding algorithms, tested them on a number of circuit examples taken from academia and industry, and present the results in Table 1. The circuits were generated using the multi-level logic synthesis technique described in [18]. We assumed that each minimum sized fanout of a signal contributes 25fF to the output load of a gate and counted fanouts; we also assumed a 5 volt power supply. Primary outputs were assumed to drive four additional minimum sized gates.

The largest circuit we tested is a second-level cache-controller [10] which consists of over 400 gates and 20 burst lists. We assumed a 80% hit rate for both read and write cycles, that 90% of the cycles were read accesses, 8% write accesses, 2% snoop accesses.

The table shows that hazards for these circuits contribute less than 5% of the total energy consumed per signal transition. While the results are not conclusive due to the relatively small sample size, they suggest that unlike some multi-level combinational circuits, burst-mode multi-level circuits have very few glitches.

References

[1] P. A. Beerel, C.-T. Hsieh, and S. Wadekar. Estimation of energy consumption in speed-independent control circuits. In *International Symposium on Low Power Design*, 1995.

[2] A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of average switching activity in combinational and sequential circuits. In *Proc. ACM/IEEE Design Automation Conference*, pages 253–259, 1992.

[3] P. Kudva and V. Akella. A technique for estimating power in asynchronous circuits. In *International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, pages 166–175, 1994.

[4] D. S. Kung. Hazard-Non-Increasing Gate-Level Optimization Algorithms. In *IEEE ICCAD Digest of Technical Papers*, pages 631–634, 1992.

[5] R. Marcalescu, D. Marcalescu, and M. Pedram. Switching activiy analysis considering spatiotermporal correlations. In *Proc. International Conf. Computer-Aided Design (ICCAD)*, pages 294–299, 1994.

[6] R. Marcalescu, D. Marcalescu, and M. Pedram. Efficient poewr estimation of highly correlated input streams. In *International Symposium on Low Power Design*, 1995.

[7] A. Marshall, B. Coates, and P. Siegel. Designing an asynchronous communications chip. *IEEE Design & Test of Computers*, 11(2):8–21, 1994.

[8] J. Monteiro, S. Devadas, and B. Lin. A methodology for efficient estimation of switching activity in sequential circuits. In *Proc. ACM/IEEE Design Automation Conference*, pages 12–17, 1994.

[9] S.M. Nowick and B. Coates. UCLOCK: automated design of high-performance unclocked state machines. In *Proc. International Conf. Computer Design (ICCD)*, October 1994.

[10] S.M. Nowick, M.E. Dean, D.L. Dill, and M. Horowitz. The design of a high-performance cache controller: a case study in asynchronous synthesis. *INTEGRATION, the VLSI journal*, 15(3):241–262, October 1993.

[11] S.M. Nowick and D.L. Dill. Synthesis of asynchronous state machines using a local clock. In *Proc. International Conf. Computer Design (ICCD)*. IEEE Computer Society Press, 1991.

[12] S.M. Nowick and D.L. Dill. Exact two-level minimization of hazard-free logic with multiple-input changes. *IEEE Transactions on Computer-Aided Design*, 14(8):986–997, August 1995.

[13] S. Ross. *Introduction to Probability Models*. Academic Press, 1985.

[14] C.-Y. Tsui, M. Pedram, and A. Despain. Exact and approximate methods for calculating signal and transition probabilities in FSMs. In *Proc. ACM/IEEE Design Automation Conference*, pages 18–24, 1994.

[15] K. van Berkel, R. Burgess, J. Kessels, M. Roncken, F. Saeijs, and A. Peeters. Asynchronous circuits for low power: A DCC error corrector. *IEEE Design & Test of Computers*, pages 22–32, Summer 1994.

[16] K. Y. Yun and D. L. Dill. Unifying synchronous/asynchronous state machine synthesis. In *IEEE 1993 ICCAD* Digest of Papers. IEEE Computer Society Press, 1993.

[17] K. Y. Yun, D. L. Dill, and S. M. Nowick. Synthesis of 3D asynchronous state machines. In *International Conference on Computer Design, ICCD-1992*. IEEE Computer Society Press, 1992.

[18] K. Y. Yun, B. Lin, D. L. Dill, and S. Devadas. Performance-driven synthesis of asynchronous controllers. In *IEEE 1994 ICCAD* Digest of Papers. IEEE Computer Society Press, 1994.