The Effectiveness of Transition Counting as a Predictor of Relative Energy Consumption

David J. Pursley, Sari L. Coumeri, Donald E. Thomas ([pursley,scoumeri,thomas]@ece.cmu.edu)
Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

We evaluate the effectiveness of transition counting as a predictor of relative energy consumption. We have found that layout information is not necessary for judging the relative energy consumption of datapath logic. Neither accurate capacitance nor timing models are needed for good first order analyses. However, transition counting is not a good predictor of relative energy consumption of random logic (i.e. control logic). The results are drawn from the analyses of dozens of designs, the energy consumption of each design being estimated by a gate level simulator that we have produced. This finding strengthens previous work regarding power optimizations through the reduction of transition counts. It also allows the designer to quantify to quantify the validity of energy estimates without laying-out the design and effectively perform power trade-off analyses at a high level of abstraction.

Introduction

Power consumption of portable digital systems has become a critical design parameter. Whether the designer is concerned with extending the operating time before recharging or replacing a battery, or reducing cooling problems by limiting power dissipation, it is essential for designers to estimate power. In this paper, we will concentrate on estimation at the logic level of abstraction, since logic level estimations support our register-transfer and behavioral design methodology.

Figure 1 shows the logic level design flow and the power estimation that can be performed at each level. Our register-transfer and behavioral level design methodology, which will not be discussed here, produces a gate level design. Before the library is selected, rough relative energy estimation can be done by counting transitions and assuming either a unit- or zero-delay model. Once the design has been tech-mapped to a library, a more accurate library-specific timing model can be used. Normally, this would be a (min/typ/max) delay model, because although more accurate models (such as piecewise linear delay models) are normally included with the library, no capacitance information has been determined. At this level, transition counts with this more accurate delay model can be used to estimate relative energy consumption. Once the design has been placed and routed, accurate capacitance information is available and true energy estimates can be done. The capacitance and delay estimates produced by the place and route tools can be used to back-annotate a gate (cell) level simulator to produce accurate (“true”) energy estimates. This transition count-based model is the most accurate energy model available at the gate level of abstraction (i.e. without any transistor modeling).

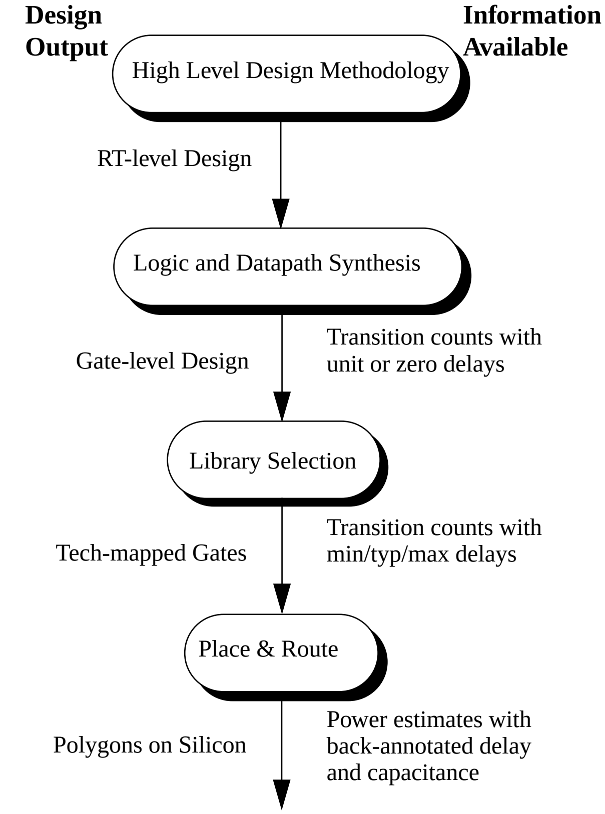

FIGURE 1. Logic level design flow and power estimation that can be done at each step.

Another way to look at Figure 1 is as levels of abstraction. At the bottom of the figure, we have knowledge about the gate level design, including accurate delay and capacitance data. At the top of the figure we have no information about the delay or capacitance, so we must make abstractions to model the missing information.

Our delay abstraction is to simply assume that all gates either have the same delay (unit-delay model) or no delay (zero-delay model). The difference between these abstraction is that the unit-delay model attempts to capture spurious transitions, while the zero-delay model does not capture spurious transitions.

Our capacitance abstraction is to assume that all gates are driving the same capacitance. We call this a "unit-capacitance" model. Using this model, transition counts are used to estimate relative energy consumption.

For our register-transfer and behavioral level design methodology, we are more concerned with relative energy estimation rather than "true" energy estimation. With relative energy estimation we are comparing register-transfer level designs and predicting how much energy each would use relative to the other. For example, relative energy estimation might predict that Design A will use 50% more energy than Design B, but it will not predict if Design B will use 1 mJ, 10 mJ, or 100mJ under the unit capacitance model.

It is our goal to quantify the effectiveness of transition counting as a predictor of relative energy consumption for these differing types of logic.

Using transition counts to predict relative energy consumption strengthens confidence in high level design methods that focus on reducing the number of transitions. However, it is likely that accuracy of such predictions varies with the type of logic being implemented, for instance datapath or random logic.

In the next section, we discuss power estimation in general and point out some previous work that our findings will serve to strengthen. We will then discuss the methodology of our experiments and the design examples we chose to use. We present the results of our experiments and then close with a discussion of the impact of these findings on previous and future work.

Background

Recently, much work has been done towards reducing power dissipation in digital circuits. But before power optimizations can be done, estimates of power consumption must be made. If we limit ourselves to CMOS circuits and logic styles, the dominant factor in power dissipation is the dynamic power consumed in the charging and discharging of the capacitive loads. This power dissipation can be calculated with the following equation [Wes85]:

$$P_{dyn} = \frac{1}{2} C V_{dd}^2 f_s$$

where C is load capacitance, V_{dd} is the supply voltage and f_s is the switching frequency of the circuit.

High level power estimators generally calculate f_s, and take as inputs the values of V_{dd} and C, which can either be extracted from layout or estimated by other tools, such as [Don79][Feu82][Lan94].

The most intuitive way to calculate f_s is to simulate the circuit at the gate-level and actually count the number of transitions made on each net. This methodology has been adopted by several tools, including [Rag96] and our tool [Pur96]. Statistical methods have also been introduced that work extremely well for some types of circuits [Mar94][Gho92][Cho94][Naj91][Lan95].

There are two issues left unaddressed by transition counting. First, load capacitance is not factored into transition counts. In a sense, using transition counts to model energy consumption assumes that all nets in the design have the same capacitance. The other issue is with the transition counts themselves. Without accurate layout information, the actual delays of the gates are unknown.

In the above equation, we see that transition counting ignores one variable (load capacitance) and may be inaccurate for another (switching frequency). This calls into question the validity of using transition counts to predict energy consumption.

In previous work regarding reducing transitions counts to minimize power, one of two approaches were taken. Either optimizations were performed and a power savings was assumed [Mur95], or the optimizations were made and resulting power savings were reported [Tiw95]. In either case, no relationship was stated regarding when transition counts could or could not be used to determine the amount of power savings.

This work quantifies the validity of transition counting as an energy predictor. Our findings tell the designer when transition counts can and cannot be used to estimate the percent change in energy consumption between designs. Because our findings are not tied to any particular optimization or trade-off, it can be incorporated into any high level design methodology.

Experimental methodology

The goal of our experiments is to quantify the validity of unit-delay transition counts as a predictor of relative energy consumption. When using unit-delay transition counts energy estimates, two abstractions are made: unit-delay and unit-capacitance.

We use several examples to explore the validity of these abstractions. We use 16 versions of the discrete cosine transform (DCT), an algorithm used in image compression, including the JPEG codec. The 16 versions vary in the number and types of multipliers, amount of parallelism and pipelining, as well as controller state encoding. Because this design includes 16 implementations, this will be our main example.

We also present results from a 4-tap finite impulse response (FIR) ASIC and two versions of a microprocessor core running a programmed version of the same algorithm. The cores implement a subset of the Motorola DSP 56000 instruction set [Mot90]. The two versions of the core differ only in their ALU's: one uses guard latches and the other does not. From here on, "Core 1" will refer to the version with guard latches, and "Core 2" will refer to the version without guard latches.

When looking at the register-transfer level descriptions of these designs, we can break our designs into three major portions: controller, datapath elements, and memory. A breakdown of the designs into these three components is shown in Table 1. We will not consider memory further here, since transistor level information is needed to capture energy dissipation in memories.

The DCT's were described at the register-transfer level, synthesized through Synopsys' Design Compiler and placed and routed in Cadence's Cell Ensemble. Back-annotated delay and capacitance estimates were obtained from the Cadence environment. The designs range from 12,000 to 21,000 gates in size, and we simulated each design with 25 8x8 blocks of data from a real JPEG example.

The FIR ASIC was described at the behavioral level and synthesized to the register-transfer level through Dasys' RapidPath. The control logic was then synthesized through Synopsys, while the datapath was synthesized through Cascade Design Automations' Epoch, which also placed and routed the design. The FIR ASIC is 12,000 gates in size and was simulated with 400 test vectors.

The core was described at the register-transfer level and also synthesized through Synopsys and Cascade. These designs are roughly 28,500 and 29,300 gates and were simulated for 1000 clock periods.

All energy estimates were performed by our gate-level simulator which used Verilog dump files and the Verilog programming language interface to count transitions. This estimator is tied into the Verilog-XL simulator, and can be back-annotated with delay and capacitance values extracted from layout [Pur96]. We found that the energy estimates produced by our tool are consistently 30% lower than the estimates produced by running Anagram's ADM in ACS mode. Our tool does not estimate energy consumed by memories. Therefore, we will not discuss memory power consumption further here.

We simulated each design under zero-delay, unit-delay, and back-annotated delay timing models, counting gate level transitions for each functional unit. We compared the zero-delay and unit-delay transition counts against the transition counts produced by simulating the designs with back-annotated delays to judge the accuracy of the zero- and unit-delay timing abstractions. Transition counting with delay information back-annotated from layout is the most accurate transition count estimate available at this level.

We also estimated energy for each design by back-annotating capacitance and delay information from the layout of each design. Comparing the zero- and unit-delay transition counts with the energy numbers for each functional block allowed us to assess the validity of using transition counts as an energy predictor for each functional block of each design. We also compared back-annotated transition counts to back-annotated energy estimates to give us an upper bound on how well transition counts could predict energy given "perfect" transition counts.

Designs	Datapath Elements	Random Logic	Memory	Gate Count
DCT (16 designs)	yes	yes	yes	12,000 to 21,000
FIR ASIC	yes	very little	yes	12,600
DSP core (2 designs)	yes	yes	yes	28,500 and 29,300

TABLE 1. Breakdown of analyzed designs into their major components.

Results

We performed the analyses described above on each design. We first present data quantifying the validity of unit- and zero-delay models as a predictor of "true" transition counts. We will then quantify the validity of the unit-capacitance abstraction, where transition counts are used as a predictor of actual energy consumption.

Estimating transition counts

In order to judge the accuracy of the unit- and zero-delay models in estimating transition counts, we simulated each of the 16 DCT designs using zero-delay, unit-delay, and back-annotated delay models. The results of those simulations are shown in Figure 2.

The unit-delay transition counts correlate with back-annotated transitions with ρ=0.94. In fact, the unit-transitions nearly coincide with the back-annotated delay transition line, the line a perfect predictor would produce. This suggests that unit-delay transition counts are a good estimate of "true" (back-annotated) transition counts.

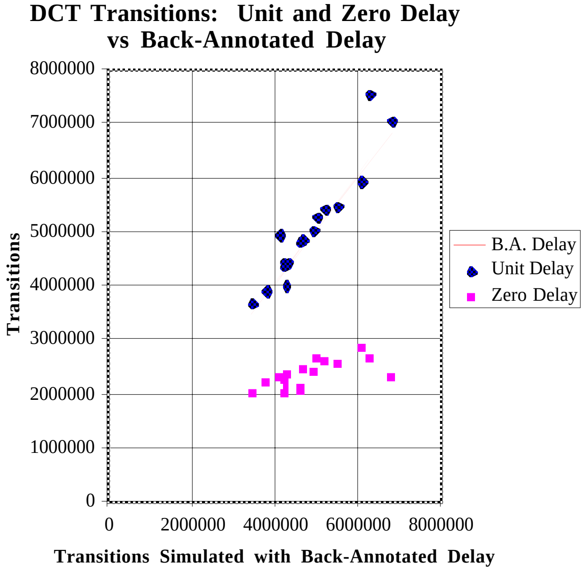

FIGURE 2. Accuracy of unit and zero delay models for transition counting.

Zero-delay transition counts correlate with back-annotated transitions with ρ=0.68. It is not surprising that zero delay transition counts do not correlate as well as unit delay transition counts since spurious transitions are not accounted for in the zero-delay model. For the DCT designs, 43-66% of the "true" transitions are spurious.

The other designs produce similar results, as shown in Table 2. For these designs, unit-delay transition counts correlate with back-annotated transition counts with ρ>0.98. Zero-delay transition counts have a lower ρ value than the corresponding unit-delay transition counts for each design.

Note that the correlation values for the FIR ASIC and core implementations were also derived by creating a scatterplot of unit- and zero-delay transitions vs. back-annotated transitions with each data point corresponding to a functional unit in the design. The FIR ASIC was broken into 20 functional units, while the cores were broken into 58 functional units.

Although the zero-delay transition counts are accurate for some examples, the unit-delay transition counts are always more accurate. When using a simulator, either delay model can be chosen. Because it is more accurate, the unit-delay model is the better choice. Therefore, from here on, zero-delay transition counts will be presented in the data sets, but they will not be discussed in detail. We will focus our discussions on unit-delay transitions.

Estimating energy consumption

Design	Unit-delay correlation coefficient	Zero-delay correlation coefficient
DCT's	0.94	0.68
FIR ASIC	0.996	0.992
Core 1	0.987	0.964
Core 2	0.987	0.964

TABLE 2. Correlation of unit-delay and zero-delay transitions with back-annotated delay transitions for each design.

In the previous section, we showed that counting transitions under a unit-delay model is a fairly accurately predictor of "true" transition counts. In this section, we evaluate the effectiveness of unit-delay transition counts as a predictor of relative energy consumption.

Figure 3 compares transition counts to the energy estimates obtained by back-annotating capacitance and delay information gained from layout for the DCT designs. The lines are zero-delay, unit-delay, and back-annotated delay transition counts. Note that these data points are the same as in Figure 2, but divided by 25 to obtain transitions per block of input data. Transition counts are on the right axis. The dark bars are energy estimates, and their axis is the left axis.

Visually, transition counts are at best weakly correlated with energy estimates when looking at the DCT designs. Statistical analysis confirms this observation; the correlation coefficients are ρ=0.24 for zero-delay transitions, ρ=0.25 for unit-delay transition, and only ρ=0.37 for back-annotated delay transition counts. Again, back-annotated transition counts represent "true" transition counts, the transition counts a perfect transition predictor would provide. As such, it provides an upper bound for the accuracy of transition counting as an energy predictor. Note that these values of ρ were obtained from a scatterplot of zero-, unit-, and back-annotated delay transitions vs. energy.

More insight can be gained into the validity of transition counting as an energy estimate by looking at individual portions of the designs. When considering only the multipliers in the DCT, transitions correlate with energy much better, as shown in Figure 4.

Note that for this graph and the other graphs presented in this section, the X-axis is the energy estimate provided by back-annotating capacitance and delay estimates from layout, and the Y-axis is the transition counts under zero-, unit-, and back-annotated delay models. Regression lines are draw for each delay model to help visually display the strength of correlation.

The unit-delay transition counts correlate with back-annotated energy estimates for the DCT multipliers with ρ=0.85. Note that if we omit the one design that

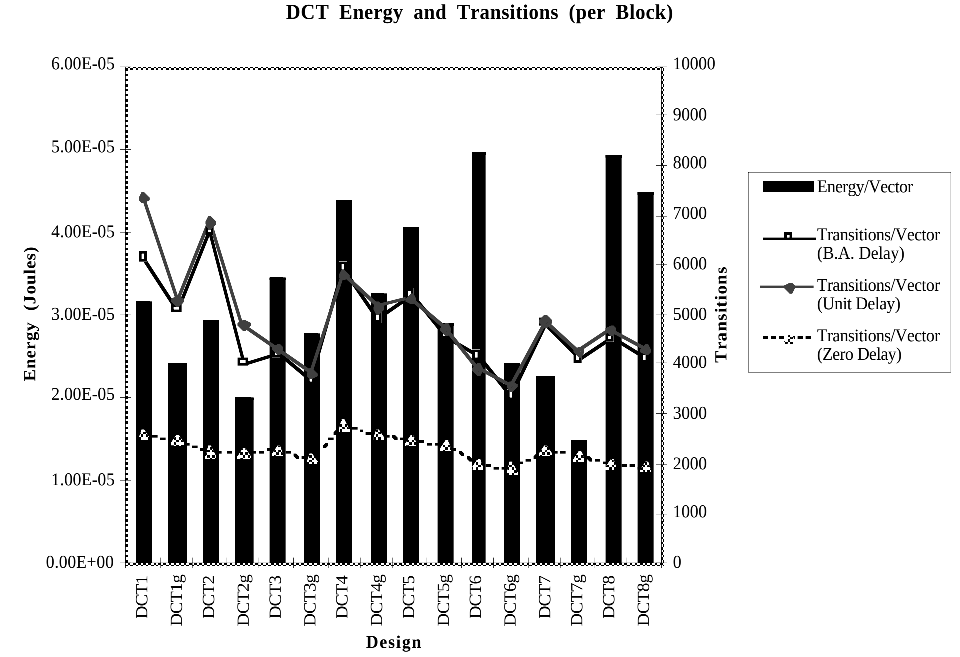

FIGURE 3. Transitions and Energy estimates for the DCT designs.

Cadence placed and routed quite differently than the others, ρ rises above 0.90.

This suggests that unit-delay transition counting is a good predictor of relative energy for the multiplier portions of this design.

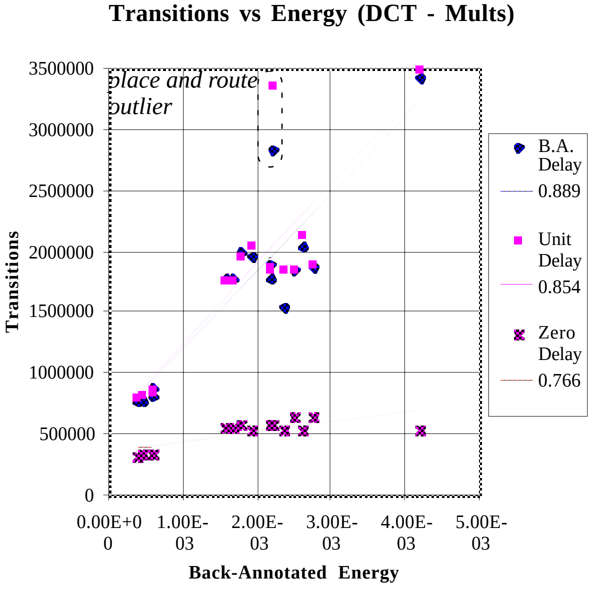

FIGURE 4. Transitions vs. Energy for multipliers inside the DCT designs.

Figure 5 shows similar results for the adders and subtractors in the DCT design. Although unit-delay transition counts do not correlate as well for these functional units as they did for the multipliers, they still correlate moderately well with energy (ρ=0.78). We expect this correlation would be somewhat higher if a datapath layout tool were used to place and route the design, as occasionally these layouts (admittedly using Cadence's default settings) produced unexpected results, such as splitting adders in half.

Still, the ρ=0.78 correlation suggests that unit-delay transitions are an acceptable predictor of relative energy consumption for the adders and subtractors in the DCT.

The remaining portion of the DCT is the random logic. Random logic is defined as units laid out in an irregular "sea of gates" fashion. In the case of the DCT's, this includes not only the control logic, which was quite significant in some of the more complicated pipelined designs, but also the multiplexors since they were laid out in this fashion as well (again using Cadence's default settings).

As shown in Figure 6, the correlation coefficient of the unit delay transition counts is ρ=-0.10. The back-annotated transition counts, which are the most accurate transition counts possible at the logic level have a correlation coefficient of only ρ=0.14.

It is clear that transition counts cannot be used to predict energy for random logic in the DCT. The unit-capacitance model is not a good abstraction due to the wider variation in gate load capacitances for random logic.

The other designs produced similar results. Unit-

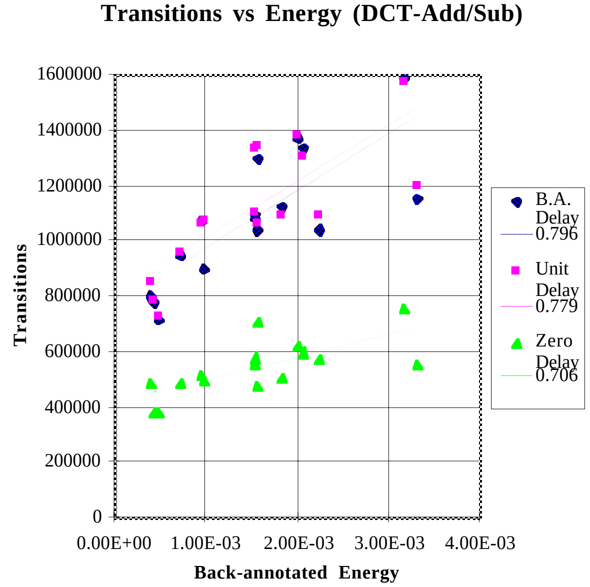

FIGURE 5. Transitions vs. Energy for adders and subtractors inside the DCT designs.

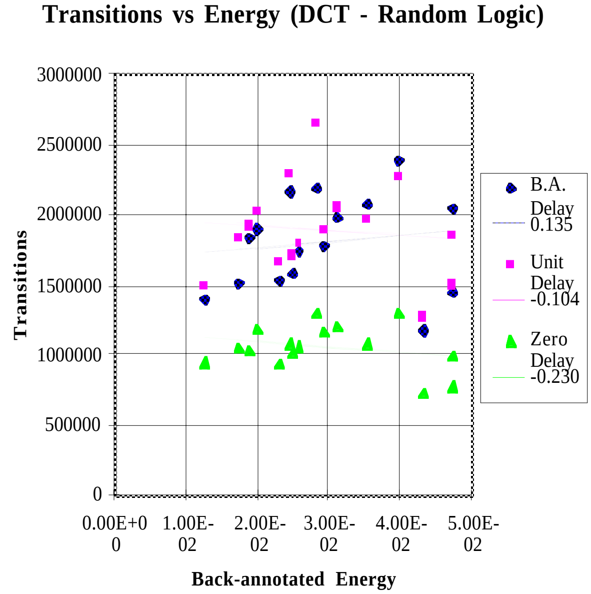

FIGURE 6. Transitions vs. Energy for random logic inside the DCT designs.

delay transitions can be used to predict relative energy consumption of datapath elements but not random logic. Table 3 shows the correlation results for all of the designs. Note that the correlation coefficient column corresponds to ρ in Figures 4 through 6.

The datapath portions of these other designs were synthesized and laid-out as datapaths using Cascade's Epoch, a datapath layout tool, which explains why their correlations (ρ>0.95) are better than even the multipliers from the DCT's.

As before, unit-delay transition counts for the random logic portions of the designs do not correlate as well with energy (ρ=0.695 for Core 1's ALU control logic and ρ=0.0418 for both cores' bus switch) as the datapath portions of these designs.

An unexpected result is that unit-delay transitions for Core 2's ALU control logic correlate with energy estimates (ρ=0.902). This high correlation value is due to a surprisingly regular layout of the control logic. From time to time, transitions counts will actually correlate to energy estimates as they did in this example. However, such a correlation cannot be expected and, in fact, would not be known until after the design has been laid-out, which negates the usefulness of using unit-delay transition counts to predict relative energy consumption.

Design	Correlation coefficient
Datapath Elements:	
Core 1 and 2 AGU	0.990
FIR ASIC Datapath	0.986
Core 1 and 2 PCU	0.983
Core 1 ALU Datapath	0.956
Core 2 ALU Datapath	0.952
DCT Mults	0.854
DCT Adders	0.779
Mixed Datapath and Random Logic:	
Core 2 ALU (Entire)	0.926
Core 1 ALU (Entire)	0.833
DCT (Entire)	0.250
Random Logic:	
Core 2 ALU Control	0.902
Core 1 ALU control	0.695
Core 1 and 2 Bus Switch	0.0418
DCT Random Logic	-0.104

TABLE 3. Correlation of unit-delay transition counts with energy for components of all designs.

As expected, the correlations for designs containing both datapath elements and random logic elements fall somewhere between the correlations for the datapath elements and the random logic of the design.

Note that for the designs others than the DCT, the ρ values were calculated by creating a scatterplot of unit-and transitions vs. energy with each data point corresponding to a functional unit in the design. The designs were broken down as in the previous section.

Discussion

Using unit-delay transition counts to predict energy abstracts away delay and capacitance information. We have already shown that the unit delay model is a valid abstraction; thus, the majority of the error between the back-annotated transition counts and back-annotated energy estimated is introduced by an error in the unit capacitance model.

Table 4 supports this claim. The gate load capacitances of the adders and subtractors have a standard deviation 2.4 times larger than the multipliers, which explains why its transitions and energy correlation coefficient (ρ=0.78) was less than the correlation coefficient for the multipliers (ρ=0.85). The capacitances of the random logic portions of the design have a standard deviation of more than 21 times the standard deviation of the multipliers. This explains the lack of correlation between transition counts and energy for the random logic.

We could not perform a similar evaluation with the other designs since only one or two versions of each design were implemented.

Design Portion	ρ	Normalized standard deviation of load capacitances
Multipliers	0.85	1.0
Adders & Subtractors	0.78	2.4
Random Logic	-0.102	21

TABLE 4. Capacitance distribution of portions of one DCT design.

Conclusions and future work

We have determined that transition counts with a unit-delay model are a fairly good estimate of both “true” transition counts and relative energy consumption for datapath elements. Unit-delay transition counts correlate very highly (ρ>0.95) with energy estimates for datapath elements that have been synthesized and laid-out with a datapath synthesis tool. Therefore, low level capacitance and timing information is not needed to accurately predict relative energy consumption in datapath circuits. However, transition counting is not suitable for energy estimation of random logic.

Characterizing when transition counts can and cannot be used as a predictor of relative energy gives high level design tools a way to estimate confidence in transition counts power estimates. For example, a high level tool may be able to simply use transition counts when evaluating possible datapath implementations, but it will probably need layout information for control logic or other logic that will be synthesized and laid out in a “sea of gates” manner. Thus, selection of layout style (datapath vs. random logic/standard cell) has an impact on energy predictability.

Counting zero-delay transitions misses all glitching activity and is a less accurate predictor of “true” transition counts. This makes zero-delay transition counts unacceptable as an energy predictor.

These results should give high level designers confidence in the validity of transition counts as an accurate predictor of relative energy consumption for datapath elements. Designers can quickly evaluate the effects of high level design decisions by synthesizing them only to the gate level. This finding also strengthens confidence in using high level design methods that focus on reducing the number of transitions. Eventually, we plan to add high level memory energy estimation to our estimation tool and tie the estimator more strongly to behavioral and higher level synthesis environments.

Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency under Order No. A564, the National Science Foundation under Grant No. MIP9408457, and the Semiconductor Research Corporation under Task ID No. 068-064. The United Stated government has certain rights to this material.

References

[Cho94] T. Chou, K. Roy, S. Prasad, “Estimation of circuit switching activity considering signal correlations and simultaneous switching,” *Proceedings of ICCAD 94*, pp. 300-303, Nov. 1994.

[Don79] W. Donath, “Placement and average interconnection lengths of computer logic,” IEEE Transactions on Circuits and Systems, pp. 272-277, April 1979.

[Feu82] M. Feuer, “Connectivity of random logic,” *IEEE Transactions on Computers*, pp. 29-33, Jan. 1982.

[Gho92] A. Ghosh, S. Devadas, K. Keutzer, and J. White, “Estimation of average switching activity in combinational and sequential circuits,” *Proceedings of DAC 92*, pp. 253-259, 1992.

[Lan94] P. E. Landman, “Low-power architectural design methodologies,” Electronics Research Laboratory, College of Engineering, University of California, Berkeley (UCB/ERL M94/62), 1994.

[Lan95] P. E. Landman and J. M. Rabaey, “Architectural power analysis: the dual bit type method," *IEEE Transactions on VLSI Systems*, pp. 173-187, June 1995.

[Mar94] R. Marculescu, D Marculescu, and M. Pedram, "Switching activity analysis considering spatiotemporal correlations," *Proceedings of ICCAD 94*, pp. 294-299, Nov. 1994.

[Mot90] Motorola, Inc. *DSP56000/DSP56001 Digital Signal Processor User's Manual*, 1990.

[Mur95] R. Murgai, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Decomposition of logic functions for minimum transition activity," *Proceeding of European Design and Test Conference 1995*, pp. 404-410, March 1995.

[Naj91] F. Najm, "Transition density, a stochastic measure of activity in digital circuits," *Proceedings of DAC 91*, pp. 644-649, June 1991.

[Pur96] D. J. Pursley, "A gate level simulator for power consumption analysis," M.S. Thesis, Carnegie Mellon University, May 1996.

[Rag96] A. Raghunathan, S. Dey, and N. K. Jha, "Register-transfer level estimation techniques for switching activity and power consumption," *Proceedings of ICCAD 96*, pp. 158-165,Nov 1996.

[Tiw95] V. Tiwari, S. Malik, and P. Ashar, "Guarded evaluation: pushing power management to logic synthesis/design," *Proceedings of 1995 International Symposium on Low Power Design*, pp. 221-226, April 1995.

[Wes85] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Reading, MA: Addison-Wesley Publishing Company, pp. 147-149, 1985.