

# Formulation and Validation of an Energy Dissipation Model for the Clock Generation Circuitry and Distribution Networks

David Duarte, Vijaykrishnan Narayanan, Mary Jane Irwin and Mahmut Kandemir  
 Department of CSE., The Pennsylvania State University, University Park, PA. 16802.  
 {duarte, vijay, mji, kandemir}@cse.psu.edu

## Abstract

*Proliferation of mobile devices and increasing design complexity have made low power consumption one of the major factors guiding digital design. The clock distribution and generation circuitry forms a critical component of current synchronous digital systems and is known to consume around a quarter of the power budget of current microprocessors.*

*We propose and validate a high level model for evaluating the energy dissipation of the clock distribution and generation circuitry. Our validation results show that our model is fairly accurate and will be suitable for use in architectural level energy simulators. We believe access to this model can precipitate further research at high-level design stages in optimizing the system clock power.*

## 1. Introduction

In today's digital design, high performance and scalability have been joined by low power consumption as the directives that are to be followed if a commercial microprocessor is to be widely accepted. In particular, the design of the clock distribution sub-system remains as one of the main challenges, if power savings are desired.

Previous studies [1], [2] have shown that the clock circuitry is responsible for up to 30% of the power consumption of a microprocessor. Many researchers [3], [4] have highlighted the importance of the clock power in the total power budget but have focused only in circuit-level methods to obtain clock power savings. Other authors have proposed clock distribution energy models [5], [6]. These, however, do not model the clock generation circuitry with enough detail, as it is done here. Such a model is of extreme importance in completing a framework [7], [8] for architectural-level decisions on total power budget. Since the details of the model have already been discussed in detail [15], the focus of this document is to provide a complete and careful validation of the model and to present some of the insights that can be obtained from it.

## 2. A clock power model

We assume a microprocessor implementation where all components are built from static CMOS gates and a single phase clocking strategy is provided for all sequential logic. Then, the capacitance driven by the clock ( $C_{\text{clock}}$ ) can be estimated as the sum of the following components (a clock gating factor per component is required to appropriately model the effective load):

$$C_{\text{clock}} = \left[ \begin{array}{l} C_{I\text{-cache}} + C_{D\text{-cache}} + C_{I\text{-TLB}} + C_{D\text{-TLB}} + C_{RgFile} \\ + C_{PipeRegisters} + C_{PLL} + C_{Drivers} + C_{Wire} + C_{Control} \end{array} \right]$$

### 2.1. Memory Structures

Much effort has been devoted to the study of memory structures and, in particular, to the development of cache [9], register file [10] and translation lookaside buffer (TLB) [11] power models since, as it happens with the clock, they constitute also a big portion of the total microprocessor power. Here, we assume a very general model, which will help to determine the precharge circuitry capacitive load on the clock network.

Let us suppose an address with  $t_c$  bits of tag,  $s_c$  bits for the set index and  $b_c$  bits for word index. So, there are  $N_{ic} = 2^{s_c}$  rows and  $n_{ic} = [(2^{b_c} w_c) + t_c]$  columns. We assume the classic 6 transistors cell and a single NMOS transistor per bit line to provide the precharge voltage. We assume also that the gate capacitance of the precharge transistor to be proportional to the bit line capacitance but scaled down by a factor beta. The bit line capacitance per way is then:

$$C_{bl/way} = N_{ic} [C_{wire} H_{cell} + C_{drain} W_{tr}]$$

Where the first term represents the interconnect capacitance ( $H_{\text{cell}}$  = cell height) and the second represents the capacitance due to the pass transistor. Then, the total precharge capacitance per way:

$$C_{prech/way} = n_{ic} \left[ C_{wire} W_{cell} + 2 \left( \frac{C_{bl/way}}{\beta} \right) W_{pt} \right]$$

The first part represents interconnect capacitance ( $W_{\text{cell}}$  = cell width) and the second estimates the gate capacitance of the precharge transistors (one per bit line, for a total of 2). So, for the whole array we have:

$$C_{\text{prech}} = \left\{ \begin{array}{l} m[C_{\text{prech/way}}](1+k) + C_{\text{status}}(1+k) + \\ (m-1)C_{\text{wire}}W_{\text{way}} \end{array} \right\}$$

Where the k factor is different from zero (and usually a number between 0.2 and 0.3) and the last term are only to be considered if intermediate buffers are included between ways ( $W_{\text{way}}$  is the width of a bank or “way”). The second term is the additional load due to storage required for the status bits that support the replacement policy, if any. Then, the effective capacitance per memory read is:

$$C_{\text{mem/read}} = C_{\text{prech}} + C_{\text{control}}$$

The component due to the control logic may be discarded if the size of the memory is large, which is usually true. During a write, since now there is no need to precharge the bit lines, only the control circuitry must be clocked. The model can be easily tailored to the specific cases of caches, register files and the TLB’s [15].

## 2.2. Clock Generator (PLL)

Clock generators have been widely implemented using Phase Locked Loops [12]. A power consumption analysis, assuming that the PLL is in **locked condition**, was done and the details can be found in [15]. For the phase-frequency detector (PFD) and the frequency divider (FDIV) the effective capacitances can be expressed as:

$$C_{\text{eff-pfd}} = \frac{C_{\text{pfd}}}{m} = \frac{1}{m} [N_{\text{tpfd}} G C_{\text{gate}}]$$

$$C_{\text{eff-div}} = [N_{\text{div}} G C_{\text{gate}}]$$

Where the factors  $N_{\text{tpfd}}$  and  $N_{\text{div}}$  estimate the number of transistors in the design, G represents the activity factor and  $C_{\text{gate}}$  is the gate capacitance of a minimum-size transistor. Similarly, for the VCO in both differential and single-ended configurations, we have:

$$C_{\text{vco-diff}} = 2n^2 k C_{\text{cell}} = 2n^2 k (5C_{\text{gate}}) \quad (1)$$

$$C_{\text{vco-nodiff}} = 2n C_{\text{cell}} = 2n (12C_{\text{gate}}) \quad (2)$$

In the above expressions, n is the number of stages, k is the fraction of  $V_{\text{dd}}$  at which the cell switches and  $C_{\text{cell}}$  is the cell capacitance. We disregard any power consumption caused by the low pass filter and the charge-pump circuit, as long as the PLL is locked.

## 2.3. Pipeline Registers

Assuming  $M_n$  flip-flops for register n and a total of  $N_p$  registers. The effective capacitance per register would be:

$$C_{\text{reg-n}} = M_n [C_{\text{ff}} + C_{\text{wire}} H_{\text{ff}}]$$

Here,  $C_{\text{ff}}$  is the flip-flop clock input capacitance and  $H_{\text{ff}}$  is the average height of the flip-flop (which is used to determine the interconnect capacitance). We assume a buffer per register, which is also in charge of providing gating functionality. Now, for all the registers in the pipeline:

$$C_{\text{pipe-registers}} = \sum_{n=1}^{N_n} C_{\text{reg-n}} G_n (1+k) + C_{\text{wire}} W_{\text{pipe-stage-n}}$$

Where k represents the additional power consumed by the buffers,  $G_n$  is the gating factor (i.e., the fraction of flip-flops being gated, 0 = all, 1 = none) and  $W_{\text{pipe-stage-n}}$  accounts for the width of stage n, so that the interconnect capacitance is also included.

## 2.4. Buffers and Drivers

A way to approximate the driver load is to include a constant k, as it was presented before. This, however, only considers the capacitance of the buffer attached to a unit and not all other buffers in the clock distribution network. An assumption that can be made is that buffers in the terminal points of the distribution network are built as chain of variable size inverters and optimized for speed, as described in [13]. The effective capacitance of the buffer is then given by [15]:

$$C_{\text{driver}} = \frac{C_{\text{gate}}}{2} \sum_{i=1}^{\ln(x)-1} x^i$$

The definition of the clock capacitance due to wiring follows the directives presented in [5] and [6].

## 3. Model Validation

### 3.1. Clock Generator (Phase-Locked Loop)

We implemented multiple versions of differential and single-ended VCOs, with different number of cells using Berkeley’s MAGIC CAD Tool in a 0.35um technology with a 3.3V supply voltage. Figures 1 and 2 show the resulting data of the estimated (using equations (1) and (2) in section 2.2.) and measured (as inferred from HSPICE power measurements) effective capacitances for the single-ended and differential VCOs, respectively. The number of stages is kept below 10, as power consumption is to be maintained as low as possible. Note that it was necessary to guarantee that during simulation, each VCO was running at the same frequency, so that the capacitance value can be directly extracted from the power readings. In particular, all non-differential VCO implementations were running at 353.53MHz while all differential VCO implementations were running at 500MHz.

It was observed that the estimated values had an error margin within 13% for non-differential VCOs and within

20% for differential VCOs, with respect to the actual capacitance value. This error drops to 1% for the former and to 10% for the latter, if only the first three measurements are considered (for differential VCOs, the measurements were normalized with respect to the second one). We believe that, as the slower devices (large  $n$ ) were pushed to run at the same frequency at which the faster devices (small  $n$ ) were running, excessive current was forced to flow on the device and non-linear transient effects that are difficult to model were responsible for the error margin. For differential VCOs, due to the analog nature of each cell (which can be regarded as a differential amplifier), this effect is even more accentuated.

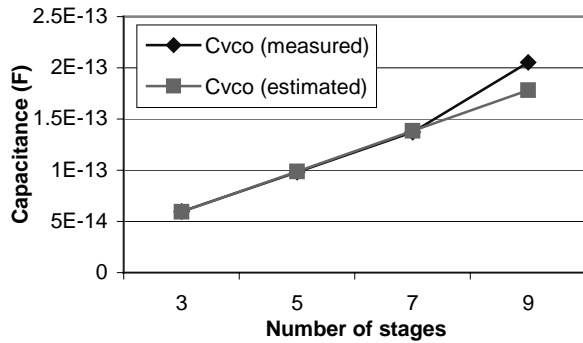


Figure 1. Capacitance data for single VCOs.

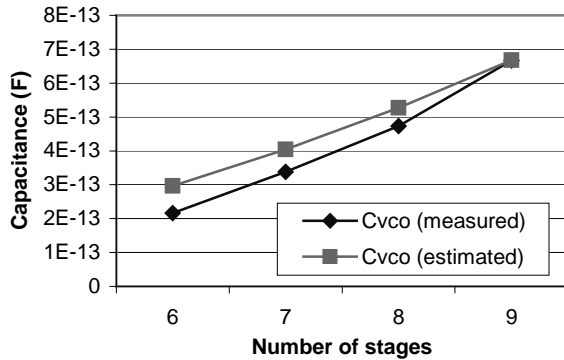


Figure 2. Capacitance data for differential VCOs.

The other remaining factor that influences the total effective capacitance is the value of  $C_{gate}$ . Table 1 shows the technology parameters required for this evaluation [14]. Since the VCO standby frequency changes as technology scales, the PFD and the FDIV layouts were used for this experiment as their operating frequency is determined only by the input signals, so it can then kept constant for all the cases. The supply voltage was again 3.3V. The equations on section 2.2., were used and the values of  $N_{pfd}$  (136) and  $N_{div}$  (72) were directly estimated by the layouts while the choice of the  $G$  (0.65) factor comes from the fact that not all the transistors switch simultaneously. Figure 3 shows the results.

Technology (um)	Oxide Thickness ( $t_{ox}$ ) (nm)	$C_{gate}$ (F)
0.35	7.70	8.24043E-16
0.5	9.70	1.33497E-15
0.8	17.0	1.95E-15

Table 1. Parameters of the technologies used.

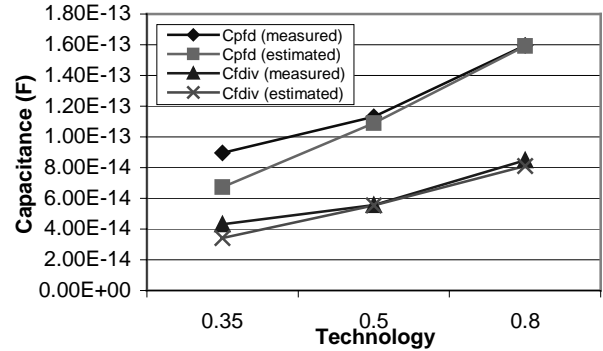


Figure 3. Effective capacitance data for the PFD.

We can see how the predicted values underestimate the measured values (obtained from HSPICE power readings done at a frequency of 100MHz), with a margin error below 25% for the PFD (for 0.5 and 0.8um technologies, the error was below 3%) and below 20% for the FDIV (below 4% for 0.5 and 0.8um technologies). Note that the error increases as technology scales down, but this, however, was mostly related to the actual design style, which, of course, is very difficult to model. It was found that signal glitching that was not present in 0.8um, was deeply accentuated in 0.35um, as the devices became faster and signals propagated also faster. Also, since process tolerances (related to the process parameter  $\lambda$ ) do not scale down at the same rate that the minimum feature does, the effective gate capacitance is somewhat larger than the predicted value, especially for technologies below 0.5um. This effect plus the influence of leakage are enhancements to be added to the model.

### 3.2. Memory Structures

For the modeling of the memory structures, technology also affects in the form of the diffusion capacitance,  $C_{drain}$ , which is taken to be:

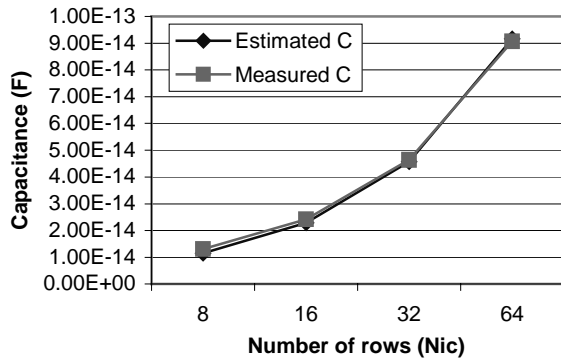
$$C_{drain} = C_{diff} = C_j L_s W + C_{jsw} [2L_s + W]$$

Where  $C_j$  is the junction capacitance,  $C_{jsw}$  is the sidewall junction capacitance,  $L_s$  is the diffusion zone length (which is usually equal to process parameter) and  $W$  is the transistor width. As it can be observed,  $C_{drain}$  is basically a function of the process being used, as shown in Table 2 for a minimum-size transistor ( $W = 3 * \lambda$ ).

Tech. (um)	$C_L$ (F/m <sup>2</sup> )	$C_{jsw}$ (F/m)	$C_{drain}$ (F)
0.25	1.74645E-3	3.6135E-10	3.44E-16
0.35	1.08158E-3	3.4800E-10	3.54E-16
0.5	5.09724E-4	4.2897E-10	7.81E-16

**Table 2.  $C_{drain}$  value as a function of technology.**

From these results is clear that as technology scales down,  $C_{drain}$  also decreases, but for technologies below 0.35um this reduction is not as significant as one would expect, which correlates to what was mentioned in the previous section. Figure 4 shows both the estimated and the measured bitline capacitance ( $C_{bl/way}$ ), for the indicated number of rows. The estimated values were calculated from the equation given in section 2.1. for  $C_{bl/way}$ , taking  $W_r$  equal to 1 and all values corresponding to a 0.35um process. The measured values were obtained from the HSPICE-generated netlist supplied after analyzing the layout. The maximum deviation is 12% with respect to the measured capacitance and the average error was calculated to be 5%. The error tends to zero as the number of rows increases (1.31% for 32 rows, 0.96% for the 64 rows case) which suggests that the model would work very accurately for the situations of interest, where the number of rows is usually large



**Figure 4. Bitline capacitance validation.**

Now, for the other dimension of the memory array, a similar equation was obtained (see section 2.1.) and is given as the precharge capacitance per way ( $C_{prech/way}$ ). Due to the similarity between this expression and the one just validated for the bitline capacitance, we omit the results here and accept an accuracy similar to the one found above.

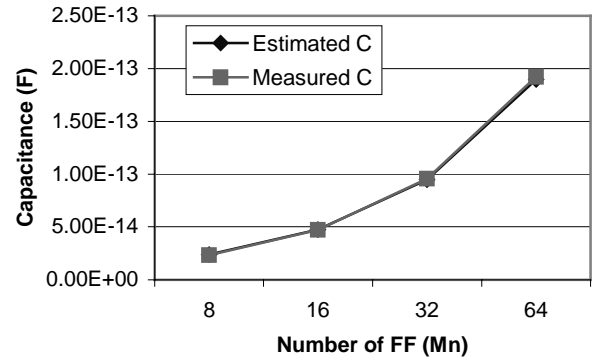
### 3.3. Pipeline Registers

In section 2.3, an expression was developed to estimate the effective capacitance of a pipeline register, with and without gating.

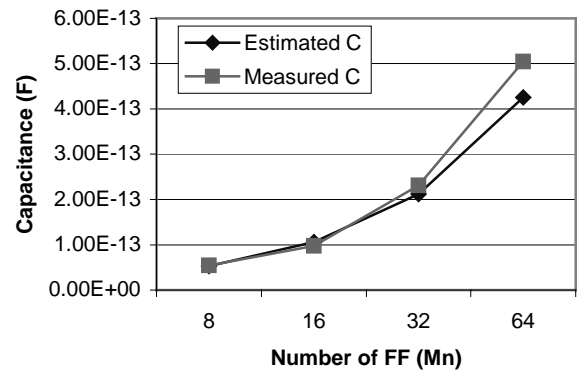
The expression is reproduced here for the sake of convenience, when  $N_p = 1$ , which represents the case evaluated.

$$C_{pipe-registers} = M_n [C_{ff} + C_{wire} H_{ff}] (G_n + k)$$

The estimated effective capacitance (as predicted by the expression above) and the measured effective capacitance (as inferred from HSPICE power readings) are shown in Figures 5 and 6, for non-buffered and buffered implementations, respectively. The variable is the size of the register ( $M_n$ ), with  $C_{ff}$  estimated as 2 times the gate capacitance and the height of the flip-flop taken from the layout. The estimation for the buffered implementation, assumes a value of  $k = 0.12$ , while for the non-buffered this factor is clearly zero. Obviously, if gating is considered ( $G_n$  is equal to 0, for total clock gating), the power dissipation is reduced (and is very close to zero) when the clock is being suppressed. Here, we intend to assess the effect of the inclusion of drivers, which are in general required, for registers with a large number of flip-flops.



**Figure 5. Pipeline register validation (no drivers).**



**Figure 6. Pipeline register validation (drivers).**

In the case where no drivers were considered, the maximum deviation from the measured values was 2.21% with an average error of 1.21%. In general, this is

reasonable due to regular structure of the register and the model prediction is remarkably accurate. On the other hand, where drivers are considered, the maximum error jumped to 15.8%, with an average of 8.92%. This relatively large error, as compared with the first case, is caused by the following reason. The drivers for the registers in the experiment were designed using the rules presented in section 2.4, which are optimized for speed and required an exponential growth in the size of the consecutive inverters in the chain that constitutes the driver. But, in the equation presented above, the overhead is estimated as a linear increase in the total effective capacitance (the k factor), which clearly will grow slower for larger values of  $M_n$ , as compared to an exponential increase.

### 3.4. Buffers and Drivers

Following what was presented in section 2.4, the expression used for capacitance estimation is:

$$C_{driver} = C_{inv} u_{ave} = 4 * C_{gate} \sum_{i=1}^{N-1} u^i$$

For the validation of the model presented for the buffers, the only variable that is changed is the number of stages (N) of the driver. The remaining factors are basically function of technology, effect that was previously examined in section 3.1. For the simulations, u was taken to be 3 (since the optimum value of u, which is e, is impossible to implement) and the results are presented in Figure 7.

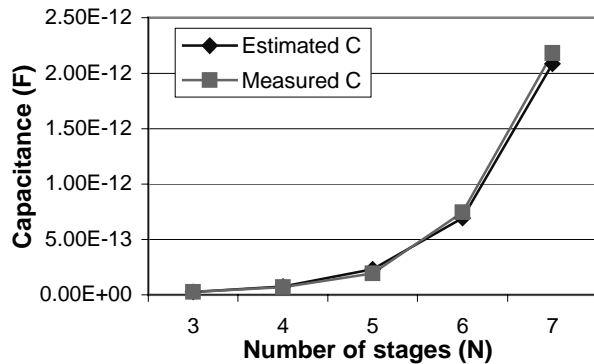


Figure 7. Buffers and drivers validation.

It was found that the average error was 9.7%, which is comparable to what has been found previously in other validations. The error increases for larger N, which makes sense as the effect of parasitics like  $C_{drain-source}$  and  $C_{drain-bulk}$  were not considered in the initial model and their effect is not negligible for large transistors. This, of course, is to be studied and included in the second revision of the model.

## 4. Insights provided by the model

Once that it has been shown that the model is reasonable accurate, it can be used to evaluate the influence of the different loads present in the clock network as they are changed. In other words, since the model is parameterizable, almost infinite configurations can be studied and assessed. Also, the model can be a very useful tool in showing behaviors that might not be, in general, easy to detect. To illustrate this, the following figures present the clock power breakdown as a function of the different loads previously analyzed, for two microprocessor configurations, whose details are presented in Table 3.

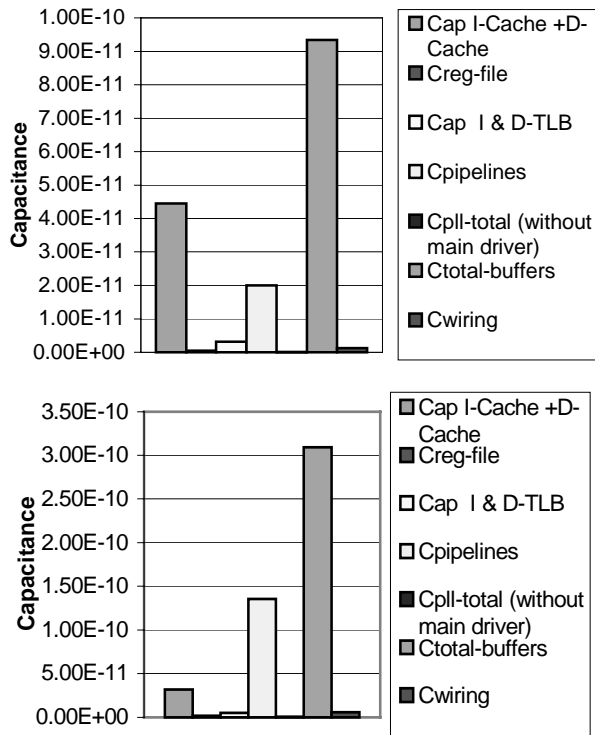
Feature	Proc. 1	Proc. 2
Address Bus Width	32	40
Data Bus Width	64	128
General Purpose Registers	32	32
Floating Point Registers	32	32
Number of Pipelines	5	5
- Register Width	64	96
- Stages in pipeline	5	9
L1 D-Cache (and I-Cache)	32 Kb	8 Kb
- Mapping	8-way	Direct
- Cache block size	32 bytes	32 bytes
D-TLB (and I-TLB)	128	64
- Mapping	2-way	Full
- Virtual address bits	52	43
Technology	0.25um	0.35um
Die Size	40 mm <sup>2</sup>	298 mm <sup>2</sup>

Table 3. Two microprocessor configurations.

There are many interesting insights given by this very simple experiment. The first thing to notice is that in both cases, even though the configurations and technologies were different, the main contributor to the clock power budget is the buffer power dissipation. Then, we have the caches and pipeline registers coming in a second place. The configuration of each processor has a strong influence, but it makes sense that they are also major players due to the generally large size of the caches (processor 1) and also to the super-pipelined and multiple-pipe configuration of modern microprocessors (most clearly shown by processor 2). All other loads, are almost negligible with respect to these main contributors, but it is important to keep in mind that this kind of behavior might be very different for other types of systems, like embedded processors, specific-purpose processors, etc.

So, it is clear that even simple comparisons like the one presented above, are very useful in providing details about

how the clock power is distributed across the different loads. This would help to focus on the loads that are causing the largest power consumption, so that strategies and optimizations can be specifically developed for these power-hungry loads, which in return will provide more significant savings.



**Figure 8. Clock capacitive load breakdown for processor 1 (top) and processor 2 (bottom).**

## 5. Concluding Remarks

The design of the clock distribution sub-system remains as one of the main challenges, if power savings are to be achieved. In this work, we have presented a complete model for a single-issue microprocessor, which also can serve as the starting point for the definition of models for embedded processors, multiple-issue machines, other specific-purpose processors and also Systems-on-a-Chip designs. The model includes both the clock distribution and clock generation circuitry. It includes parameterizable loads for the clock distribution network by providing the capability to vary the cache configurations, register file and TLB parameters and the number/size of pipeline registers. The validation results for the presented high-level models for the different clock circuitry components indicate that the model is fairly accurate. Hence, such a model will be suitable for use in high-level architectural energy simulators. We believe access to this model can precipitate further research at

high-level design stages in optimizing the system clock power. Future enhancements to the model will include the formulation of a leakage impact factor and a tolerance effect factor, which will be key for evaluating deep sub-micron technologies. Also, as there is growing interest in building caches with DRAM cells, this new influence in clock power will also be studied. Finally, the model can be enhanced to evaluate multiple architectural level modifications to the cache/register file subsystem.

## 6. References

- [1] Gronowski, P.E., et al., "High-Performance Microprocessor Design", IEEE Journal of Solid State Circuits, Vol. 33, No. 5, pp. 676-686, May 1998.
- [2] Tiwari, V., et al., "Reducing Power in High-Performance Microprocessors", Proceedings of the 35<sup>th</sup> DAC, 1998.
- [3] Friedman, G., "Clock Distribution Design in VLSI Circuits: an Overview", Proceedings of the IEEE ISCAS, May 1994, pp. 1475-1478.
- [4] Tellez, E., Farrah, A. and Sarrafzadeh, M., "Activity-driven Clock Design for Low Power Circuits", Proceedings of the IEEE ICCAD, Nov. 1995, pp. 62-65.
- [5] Liu D. and Svensson C., "Power Consumption in CMOS VLSI Chips", IEEE Journal of Solid State Circuits, Vol. 29, No. 6, pp. 663-670, June 1994.
- [6] Chen, R.Y., Vijaykrishnan, N. and Irwin, M.J., "Clock Power Issues in SOC Designs", Proceedings of the IEEE Computer Society Workshop on VLSI, April 1999, pp. 48-53.
- [7] Ye, W., Vijaykrishnan, N., Kandemir, M., and Irwin, M. J., "The Design and Use of SimplePower: A Cycle-Accurate Energy Estimation Tool", Proceedings of the 37<sup>th</sup> DAC, 2000.
- [8] Brooks, D., Tiwari, V., and Martonosi, M., "Wattch: A Framework for Architectural Level Power Analysis and Optimization", in Proceedings of the IEEE ISCAS, 2000.
- [9] Kamble, M., and Ghose, K., "Energy-Efficiency of VLSI Caches: A Comparative Study", Proceedings of the 10<sup>th</sup> International Conference on VLSI Design, 1997, pp. 261-267.
- [10] Zyuban, V., and Kogge, P., "The Energy Complexity of Register Files", Proceedings of ISLPED 98.
- [11] Juan, T., Lang, T., and Navarro, J., "Reducing the TLB Power Requirements", Proceedings of the 1997 International Symposium on Low Power Electronics and Design, August 1997, pp.196-201.
- [12] Tourette, B., "A 200 MHz, 600uW CMOS PLL for Mobile Communication Asics", Proceedings of the 12<sup>th</sup> Annual IEEE International ASIC/SOC Conference, Sept. 1999, pp. 366-369.
- [13] Rabaey, J., "Digital Integrated Circuits: A Design Perspective", Prentice-Hall International, NJ, 1996, pp.446-452.
- [14] Weste, N., and Eshraghian, K., "Principles of CMOS VLSI Design", Addison-Wesley Publishing Company, MA, 1994, pp. 180-191.
- [15] Duarte D., Irwin, M.J. and Narayanan, V., "Modeling Energy of the Clock Generation and Distribution Circuitry", Proceedings of the 13<sup>th</sup> Annual IEEE International ASIC/SOC Conference, September 2000, pp.261-265.