

# HotSpot : Visualizing Dynamic Power Consumption in RTL Designs

T. English<sup>\*</sup>, K.L. Man<sup>†</sup>, E. Popovici<sup>\*</sup> and M.P. Schellekens<sup>†</sup>

<sup>\*</sup> Dept of Microelectronic Engineering / <sup>†</sup> Centre for Efficiency-Oriented Languages (CEOL)  
University College Cork (UCC), Cork, Ireland  
tom.english@ue.ucc.ie, klm1@cs.ucc.ie,  
e.popovici@ucc.ie and m.schellekens@cs.ucc.ie

## Abstract

*We propose 'HotSpot', a tool for the visualization of dynamic power consumption in RTL designs. Using an RTL simulation, the amount of switching activity at major circuit nodes is determined. This information is annotated onto a custom graph of the circuit in the form of 'temperature' colour coding. HotSpot highlights areas in which switching activity (and thereby dynamic power consumption) may be reduced. We survey some existing techniques for dynamic power minimization and discuss how HotSpot may identify opportunities for the application of such techniques at RTL. We believe this could lead to reduced power consumption, faster design turnaround and better design practices in future.*

## 1. Introduction

In the last decade, considerable EDA research effort has been expended on Low Power Design Methodologies [1,2]. A wide range of design techniques has appeared in the literature which can dramatically reduce power consumption in many types of digital circuits. Some of these techniques (which will be surveyed later) have already been incorporated into popular commercial EDA tools and have become widely adopted as a result.

The power dissipation of a microelectronic circuit can be analysed at several points during the implementation flow – for example at system, register-transfer, gate or transistor level. With complex modern designs, power analysis accuracy and turnaround time increase dramatically with each subsequent step through the flow. It is therefore beneficial to the designer to be able to identify power-hungry areas early on, using a fast preliminary analysis. Appropriate changes made at RTL yield greater reduction in final power consumption than can be achieved by

optimizations at later stages, where most commercial tools focus.

In this paper, we propose a new tool for power analysis at RTL called 'HotSpot'. The tool establishes a visual connection between a design's structure and its average dynamic power consumption. This will be achieved by annotating switching information from simulation onto a custom graph of the design, extracted from the RTL. The aims of our approach include:

1. to address power and identify possible design flaws at an earlier stage than traditional gate-level or transistor-level analyses
2. to uncover more opportunities for application of existing low-power design techniques
3. to find such opportunities more quickly than in traditional manual/iterative approaches

We hope to implement HotSpot as a stand-alone tool facilitating easy integration into existing industrial design flows (for example those presented in [13]).

Examples of related work in dynamic power visualization include [8, 9, 10]. Shortly before the preparation of this paper, a report appeared concerning a toolset called GRAAL [11]. This facilitates dynamic power visualization in SystemC designs. Future work could include a comparative study between HotSpot and GRAAL.

This paper is organized as follows: Section 2 presents a survey of some existing dynamic power optimization techniques. In Section 3, the motivation for the development of HotSpot is discussed. The tool's usefulness is demonstrated with practical examples. In Section 4, we discuss how the tool may be integrated into existing industrial design flows. Section 5 describes the tool's implementation. Finally, Section 6 identifies areas for further investigation and makes some concluding remarks.

## 2. Survey of Existing Techniques for Dynamic Power Minimization

Power dissipated by CMOS logic circuits can be divided into two categories - *static* and *dynamic*. *Static power dissipation* arises from the quiescent current flow in CMOS circuits due to device-level leakage effects [3]. *Dynamic power dissipation* is made up of *internal power* and *switching power* components. In this paper, we focus on *switching power*, which remains a dominant component of overall circuit power even in deep sub-micron processes.

Switching power is dissipated when load capacitances within the circuit are charged or discharged. For synchronous digital circuits, it is commonly expressed as:

$$P_{sw} = \alpha CV^2f$$

In this expression,  $C$  is the circuit's total switched capacitance (in Farads),  $V$  is the circuit's supply voltage,  $f$  is the switching frequency (in Hz) and  $\alpha$  is the overall switching probability for nodes in the circuit – a real number ranging from 0 to 1.

Based on the expression for  $P_{sw}$ , there are several approaches to dynamic power minimization. Reducing the circuit's total switched capacitance, clock frequency or switching probability each cause linear reductions in dynamic power. Reducing the supply voltage yields a quadratic reduction. Using these approaches, several techniques have been developed to reduce dynamic power, including:

**Operand Isolation/Guarded Evaluation** – it is common in combinatorial datapath circuits that input signals act as operands for several arithmetic or boolean operators. The operator results are calculated in parallel, but not all of them are actually selected for output. Significant power is therefore wasted in the calculation of 'unused results'. We can reduce this by 'isolating' logic whenever its results go unused, preventing unnecessary switching [4, 5]. This reduction in switching probability ( $\alpha$ ) reduces dynamic power considerably in many datapath circuits.

**Pre-computation** – Alidina et al. [6] describe a technique for sequential circuits which exploits redundancy in multi-bit comparisons and bitwise reductions. Such operations can be split into two parts, enabling a small but critical part to be 'pre-computed' one clock cycle early. The result determines whether the circuit needs to perform the remainder of the calculation. If so, data is registered as normal and the calculation is completed; otherwise the circuit uses

only the pre-computed result. Again, the reduction in switching probability leads to reduced dynamic power.

**Clock Gating** – much of the dynamic power consumed by sequential circuits is dissipated in the clock tree (where there is a large switched capacitance) and in the flipflops themselves (on each clock edge). Not all flipflops may need to be updated on every clock edge. This redundancy is exploited by gating the clock to the flipflops with a suitable enable. In some cases, this reduction of the switched capacitance in the clock tree and wasted power in flipflops can result in to dramatic reductions in circuit dynamic power [1].

**Bus Encoding** – many SoC designs have shared buses for address and data traffic, often implemented physically as groups of long wires. Switching these buses consumes considerable power due to the heavy RC load they present to bus driver logic. In [1, 7], we find a summary of some common bus encoding techniques which aim to reduce the switching probability of bus lines (at the expense of some additional codec logic). This can help to reduce dynamic power dissipated in on-chip interconnect.

**Memory Partitioning for Low Power** – examining SoC memory access patterns may identify address ranges accessed frequently. Macii et al. [1] summarise techniques for physical partitioning of memories in such systems yielding reduced dynamic power consumption. The most frequently-accessed addresses reside in a small memory with minimal address decode and read/write logic overhead. Remaining addresses are stored in larger memories which are accessed less frequently (and which are de-selected when not addressed). This reduces average switched capacitance per memory access, saving dynamic power.

**Voltage and Frequency Scaling** – modern SoCs have many modes of operation in which workloads vary and different functional units become idle or busy. In units where workload is light, clock frequency ( $f$ ) can be reduced accordingly. Using a control algorithm, the supply voltage ( $V$ ) can be reduced to a level just sufficient to sustain operation at the lower frequency, thereby obtaining a dramatic reduction in dynamic power consumption [1, 3].

## 3. Motivational Examples

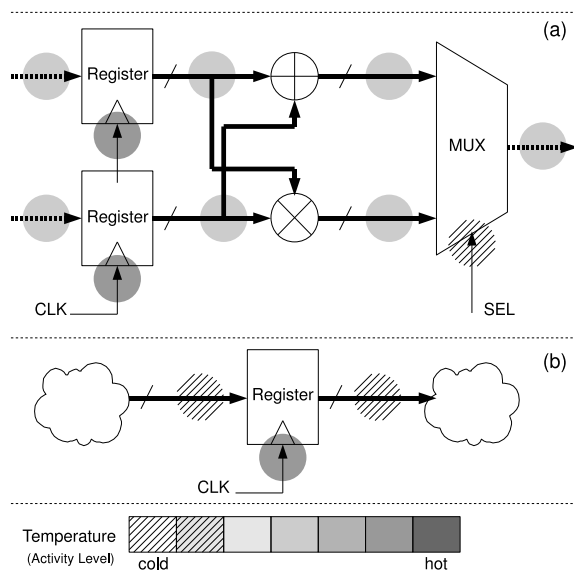
We now illustrate simple applications of HotSpot in detecting operand isolation and clock gating opportunities.

Figure 1(a) shows a primitive ALU with combinatorial addition and multiplication performed in parallel.

Switching information for the circuit can be recorded during simulation as SAIF (Switching Activity Interchange Format [12]). Several key statistics are reported for each node, including:

- T0 – total time spent at logic zero
- T1 – total time spent at logic one
- TC – number of times the node toggled

HotSpot reads the RTL and the SAIF, generating the graph in Figure 1(a). The SAIF toggle count (TC) field is used to assign nodes a 'temperature' corresponding to their relative activity levels – 'colder' for nodes with lower toggle counts, 'hotter' for nodes with greater activity. Cold signals receive a hatched spot. As signal activity increases, the hatching gives way to darker greys. The most intense grey is annotated to the most active node(s) in the circuit – usually the clock network. (In this paper, we employ a greyscale temperature scheme in place of blue-to-red gradations to improve readability).



**Figure 1 : Annotated Example Circuits**

The approach is illustrated in Figure 1(a). Combinatorial logic performs addition and multiplication, producing 'hot' data nodes, marked with medium-grey spots. The ALU selects one of the calculated results for output, producing a 'hot' output node. The MUX select line is 'cold', marked with a hatched spot. It has switched rarely during simulation.

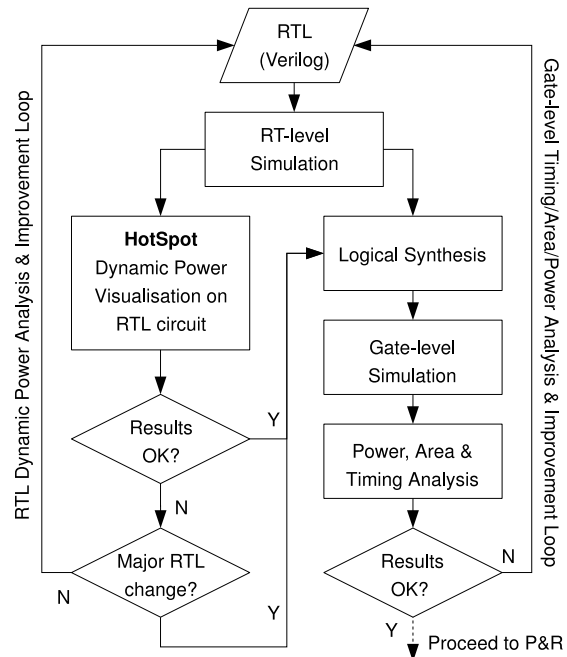
High activity on the datapath suggests that significant dynamic power is being wasted in the calculation of unused results. Introducing SEL-controlled isolation muxes or guard latches after the flipflops will reduce unwanted switching. Since this is a

busy datapath and the MUX select switches rarely, we expect power savings to justify the inevitable area and timing penalties of the additional logic.

Figure 1(b) shows a section of datapath logic. Here, the temperature spots are 'cold' on the input and output of the data register. The register is clocked with the 'hot' clock signal. The register may be oversampling the data and wasting dynamic power. If this were not the designer's intention, it might be possible to derive a clock gating enable from upstream logic which enables the register only when needed, saving dynamic power.

#### 4. Power Analysis Flow with HotSpot

HotSpot creates a quick, iterative RTL power analysis and improvement loop, illustrated in Figure 2. It helps to identify potential problems and opportunities for the application of power-saving techniques early on. No lengthy synthesis or gate-level simulation steps are needed. HotSpot does not perform power estimation and is not a substitute for traditional gate-level or physical power analyses.



**Figure 2 : Proposed HotSpot Flow**

Since changes made to the RTL can have an unexpected impact after synthesis or layout, it is still necessary to run synthesis, place and route and power analysis steps after major changes are made. The designer must use judgement and experience to determine which changes (if any) will benefit dynamic power with justifiable cost.

## 5. Implementation

HotSpot is to be composed of:

- an RTL parser (Icarus Verilog [14])
- a Switching Activity Analyser (in C/C++)
- a graph plotter (GraphViz [15])

As shown in Figure 3, a wrapper script combines the components to create the HotSpot command-line tool. After the RTL is parsed, the Switching Activity Analyser reads simulation SAIF and annotates it onto the design. It generates a switching activity summary report along with a hierarchy of the design in HTML. Drawing commands are written for interpretation by GraphViz which then generates the final bitmaps.

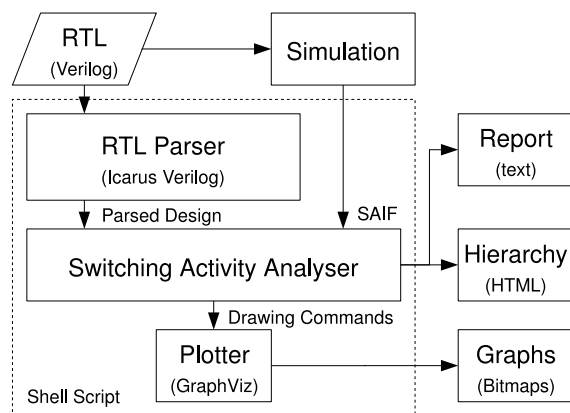


Figure 3 : HotSpot components

After each run, a web browser can be used to navigate the circuit's hierarchy and view the annotated graphs.

## 6. Conclusion and Further Investigation

We have proposed HotSpot, a tool which creates a visual link between circuits and the dynamic power dissipated in them. We believe it will make a relevant contribution to EDA research by guiding the designer to reduce waste. However, a significant amount of work remains to be done before the tool is ready.

Areas for further investigation will include:

- Testing using standard circuit benchmark sets (e.g. ISCAS, IWLS, MCNC and Texas97)
- Examination of the switching profiles of SoC address and data buses; this may help to identify appropriate power-efficient bus encodings and memory partitioning schemes
- Identification of idle circuit regions to guide clock gating, power gating or voltage/frequency scaling

- Identification of potential RTL design errors

## 7. Acknowledgement

We gratefully acknowledge the support of IDA Ireland and Synopsys in funding our research as indicated in this paper.

## 8. References

- [1] E. Macii and R. Zafalon, "Low-Power EDA Technologies: State-of-the-Art and Beyond", Book Chapter, Advanced Signal Processing, Circuits, and System Design Techniques for Communications, 2006, pp. 3-43
- [2] L. Benini, G. De Micheli and E. Macii, "Designing Low-Power Circuits: Practical Recipes", *IEEE Circuits and Systems Magazine*, 2001, pp. 6-25
- [3] M. Keating, D. Flynn, R. Aitken, A. Gibbons and K. Shi, "Low-Power Methodology Manual", Springer, 2007
- [4] A. Correale, "Overview of the power minimization techniques employed in the IBM PowerPC 4xx embedded controllers", *Proceedings of the International Symposium on Low Power Design*, 1995, pp. 75-80
- [5] V. Tiwari, S. Malik and P. Ashar, "Guarded evaluation: Pushing power management to logic synthesis/design", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v 17, n 10, Oct 1998, pp. 1051-1060
- [6] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh and M. Papaefthymiou, "Precomputation-based sequential logic optimization for low power", *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers*, 1994, pp. 74-81
- [7] E. Macii, M. Pedram, and F. Somenzi, "High-Level Power Modeling, Estimation, and Optimization", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v 17, n 11, Nov 1998, pp. 1061-1079
- [8] PowerArtist - <http://www.scdsource.com/>
- [9] PCE - <http://www.bulldast.com/>
- [10] M. French, P. Graham, M. Wirthlin, G. Larchev, P. Bellows and B. Schott, "Design Tools for Reconfigurable Hardware in Orbit (RHinO)", in *Proceedings of the 4th annual Earth Science Technology Conference (ESTC)*, 2004
- [11] B. Juurlink, I. Antochi, D. Crisu, S. Cotofana and S. Vassiliadis, "GRAAL: A Framework for Low-Power 3D Graphics Accelerators", *IEEE Computer Graphics and Applications*, v 28, n 4, Jul-Aug 2008, pp. 63-73
- [12] Synopsys SAIF (Switching Activity Interchange Format) - <http://www.synopsys.com/partners/tapin/saif.html>
- [13] R. Mehrotra, T. English, K.L. Man, E. Popovici and M.P. Schellekens, "An Efficient Power Estimation and Optimisation Design Flow of Digital Circuits" (Draft Paper, Aug 2008)
- [14] S. Williams, Icarus Verilog - <http://www.icarus.com/eda/verilog/>
- [15] E. R. Gansner, S. C. North, "An open graph visualization system and its applications to software engineering", *Software: Practice and Experience*, v 30 n 11, pp. 1203-1233, J. Wiley & Sons 2000