A New Methodology for Power-Aware Transistor Sizing: Free Power Recovery (FPR)

Milena Vratonjić[1], Matthew Ziegler[2], George D. Gristede[2], Victor Zyuban[2], Thomas Mitchell[3], Ee Cho[4], Chandu Visweswariah[2], and Vojin G. Oklobdzija[5]

[1] University of California Davis, Davis, CA
milena@ucdavis.edu
[2] IBM T. J. Watson Research Center, Yorktown Heights, NY
{zieglerm,gristede,zyuban,chandu}@us.ibm.com
[3] IBM Electronic Design Automation, Burlington, VT
tmitch@us.ibm.com
[4] IBM Electronic Design Automation, Poughkeepsie, NY
cho@us.ibm.com
[5] University of Texas at Dallas, Dallas, TX
vojin@acsel-lab.com

Abstract. In this paper we present a new transistor sizing methodology called Free Power Recovery (FPR) for low power circuit design. The objective of this methodology is to minimize the total power of a circuit by accounting for node switching activities and leakage duty cycles (LDC). The methodology has been incorporated into the EinsTuner circuit tuning tool. EinsTuner automates the tuning process using state-of-the-art non-linear optimization solvers and fast circuit simulators. Node switching activities and LDC are integrated into the EinsTuner framework as parameter inputs to the FPR tuning mode. In FPR mode, the power is minimized using gate width reduction with respect to power properties of the node. The FPR methodology is evaluated on next generation microprocessor circuit designs. Power reduction results are compared with the results from the existing EinsTuner tuning methodology. The results show improvement in power reduction with the FPR optimization mode.

Keywords: Low-power, Optimization.

1 Introduction

Power dissipation remains one of the critical challenges in microprocessor design. It requires innovation at all design levels to sustain performance scaling [1],[2]. Designers rely on the use of automated circuit design tools to provide low power and high performance circuits. However, existing automated circuit design tools focus solely on minimizing total device width under the constraint of critical path delay [3]. These tools do not consider the impact of circuit properties such as switching activities and leakage duty cycles at each circuit node. Minimization of the total device width, without accounting for these circuit properties

J. Monteiro and R. van Leuken (Eds.): PATMOS 2009, LNCS 5953, pp. 307–316, 2010.

that determine the power consumption, does not guarantee that the optimized (tuned) circuit will operate under minimum power consumption.

EinsTuner can operate under a variety of tuning modes, some of which are: area minimization, Free Area Recovery (FAR), delay minimization, etc. For example, area minimization has the goal of reducing the total device width of the circuit under the timing constraint given by the specified slack[1] threshold value. Another EinsTuner tuning mode known as Free Area Recovery (FAR) [4] attempts to reduce the total device width without degrading the slack of the critical path. FAR is a variation of the area minimization method discussed above and it also requires a slack threshold specification. However, the slack threshold parameter for the area minimization and FAR tuning modes has two very different meanings.

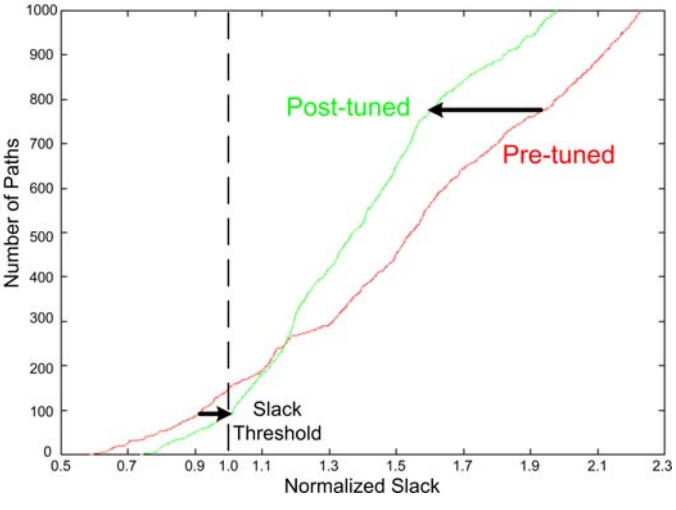

Fig. 1. Example of a slack histogram for which the worst slack is smaller then the slack threshold

In the area minimization tuning mode, the slack threshold is a constraint and has to be satisfied or the optimization problem is infeasible and the tuned circuit is suboptimal. In the FAR mode, the slack threshold is used to specify the criticality boundary. When tuning in the FAR mode, any slack that is smaller (worse than) the specified slack threshold is guaranteed to either improve or remain constant after tuning (it will not get any worse). However, this does not imply that the devices along the paths that have slacks below the threshold slack are excluded from the tunable pool of devices. All devices are subjected to optimization in the mathematical tuning problem formulation. This means that

[1] Slack is a timing characteristic associated with each timing point of the design and represents the difference in arrival time versus required arrival time.

all device widths can be modified, provided that the modification of the widths of the devices on the critical path does not cause the slack to fall below the threshold or initial slack value, depending on which value is smaller. An example is a circuit in which the critical path slack is smaller than the specified slack threshold, i.e. the worst case slack is considered to be negative. The representative slack histogram of a circuit with such timing characteristics is shown in Figure 1. We cannot tune such a circuit in the area minimization mode since the slack threshold constraint is not satisfied, but we can perform the circuit optimization using the FAR mode. Even though the worst case slack of the circuit is negative, the circuit contains many other non-critical paths that satisfy timing requirements and have positive slacks. Note that the worst case slack can also improve after tuning since off-path capacitive loading is reduced, as shown in this example for the post-tuned timing histogram in Figure 1. Unlike in the area minimization tuning mode, the slack threshold value specified in the FAR tuning mode will not cause infeasible problems. The FAR tuning mode can therefore be applied to any circuit with either positive or negative critical (worst) slack.

The ideal case is to have the circuit that is delay optimized such that further delay improvement cannot be obtained from delay-based tuning (delay minimization mode in EinsTuner) for a given circuit topology. Usually, such a circuit is said to be "fast" but it is not necessarily power optimized. To minimize the power of the circuit without degrading the critical path delay, we need a follow-up optimization step for circuit power reduction. In this paper, we propose a new tuning step for power reduction, Free Power Recovery (FPR), which will provide power reduction while maintaining circuit delay performance. The result of FPR as the second optimization step is a fast and minimum-power circuit.

2 Free Power Recovery

FPR (Free Power Recovery) is a variation of the FAR (Free Area Recovery) tuning mode. The optimization objective of the FPR tuning mode is to minimize the total power of a circuit instead of the circuit area, as done in Free Area Recovery (FAR) and in area minimization modes. The area of a circuit is represented as the total device width (1).

$$Area = \sum_{i=1}^{N} W_i \tag{1}$$

The FPR mode takes the switching activity and leakage information to formulate the total power as an objective of the optimization problem. The optimization problem is then presented to the non-linear solver under the constraint that circuit timing cannot degrade on the paths which have slack below the threshold.

The total power P of a circuit is the sum of dynamic power ($P_{dynamic}$) and leakage power ($P_{leakage}$) components:

$$P = P_{dynamic} + P_{leakage} \tag{2}$$

Dynamic power of a circuit can be represented as the sum of dynamic power contributions from each of the N devices as given by

$$P_{dynamic} = \frac{1}{2} f V_{dd}^2 c_g \sum_{i=1}^{N} \alpha_i W_i \tag{3}$$

where c_g represents the gate capacitance per unit gate width and α_i is the device switching factor. We only model dynamic power component associated with device switching which is directly affected by transistor sizing; therefore, dynamic power due to switching wire capacitance is not included in the model but it is accounted for in the total power reported in the results.

Leakage power can be expressed as

$$P_{leakage} = \sum_{i=1}^{N} p_i^{leakage} \delta_i W_i \tag{4}$$

where δ_i stands for *Leakage Duty Cycle (LDC)* of the i^{th} device and $p_i^{leakage}$ is the leakage power per device width $[\mu W/\mu m]$ which depends on the operating voltage supply V_{dd}, polarity of a device (*nFET, pFET*) and the device threshold voltage V_t (low LV_t, regular RV_t, high HV_t, and super-high SV_t threshold voltage).

Leakage Duty Cycle accounts for the probability of the input pattern for which the device is in the leaking state. It also evaluates the reduction of leakage current in transistor stacks with one or more *off* devices. State dependent calculation of leakage duty cycles for a two-input NAND gate example is summarized in Table 1. The transistor stacking effect (DIBL) [5] of two devices that are *off* is incorporated in the calculation and reduces the leakage current when compared to a single *off* device by a factor of 10, depending on the technology [6]. The source-follower effect is also taken into account with 30% leakage reduction for the nFET device at the bottom of the stack, when the top device in the stack is *on*.

Table 1. State dependent leakage LDC calculation for the two-input NAND gate example with transistor B at the bottom of the stack

a	b	State Probability	N_a	N_b	P_a	P_b
0	0	p_{00}	0.1	0.1	0	0
0	1	p_{01}	1	0	0	0
1	0	p_{10}	0	0.7	0	0
1	1	p_{11}	0	0	1	1

LDC for each device in the two-input NAND gate is calculated as follows:

$$\delta_{N_a} = p_{00}(0.1) + p_{01}(1) + p_{10}(0) + p_{11}(0) \tag{5}$$
$$\delta_{N_b} = p_{00}(0.1) + p_{01}(0) + p_{10}(0.7) + p_{11}(0) \tag{6}$$
$$\delta_{P_a} = p_{00}(0) + p_{01}(0) + p_{10}(0) + p_{11}(1) \tag{7}$$
$$\delta_{P_b} = p_{00}(0) + p_{01}(0) + p_{10}(0) + p_{11}(1) \tag{8}$$

Assuming that all states for the two-input NAND gate example are equally probable, we obtain the following values of *Leakage Duty Cycles* for each of four devices:

$$\delta_{N_a} = 0.275, \delta_{N_b} = 0.2, \delta_{P_a} = \delta_{P_b} = 0.25 \tag{9}$$

Taking equations for dynamic power (3) and leakage power (4) components, the total power (2) of a circuit can be rewritten as:

$$P = \sum_{i=1}^{N} t_i W_i \tag{10}$$

where the *weight factors* for device width in the power equation (10) are computed as:

$$t_i = \frac{1}{2} f V_{dd}^2 c_g \alpha_i + p_i^{leakage} \delta_i \tag{11}$$

Previously, we had the area of a circuit as the sum of equally weighted device widths (1). A comparison of the equations for total area (1) and total power (10) shows that the total power is proportional to the weighted sum of the gate widths, where the weight factors correspond to the t_i factors as defined in equation (11). Therefore, minimizing the total area of a circuit is not equivalent to minimizing its total power.

The FPR tuning mode properly accounts for the dynamic power of a circuit because it considers only those devices that are switching (see Figure 2). On the other hand, the area minimization and FAR modes consider all devices equally in the optimization, irrespective of switching activity.

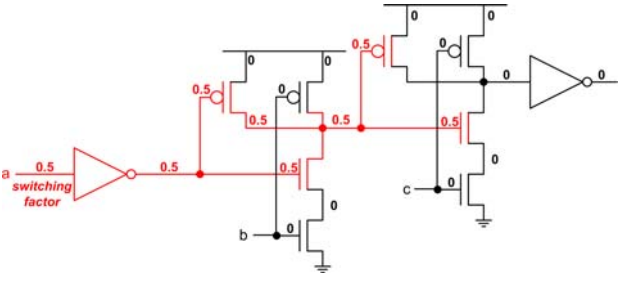

Fig. 2. Example of propagation of switching factors in a circuit

3 FPR vs. FAR

Figure 3 depicts the results we expect from tuning a circuit in FAR and FPR modes. The left-hand side of the figure shows power and timing characteristics of a pre-tuned and post-tuned circuit in both modes. The right-hand side shows area and timing characteristics of these circuits. The pre-tuned design point is delay optimized. Starting from this point and applying both FAR and FPR tuning, we expect FAR mode to achieve minimum area, and FPR mode to achieve minimum power.

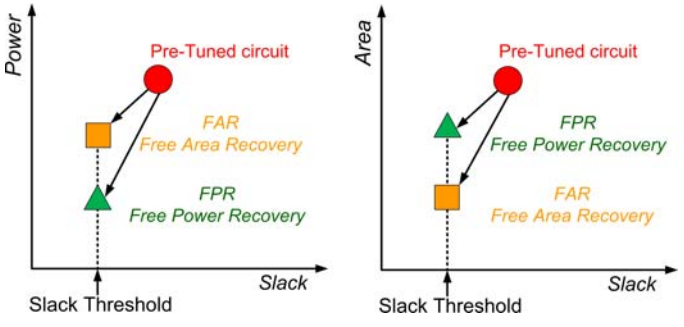

Fig. 3. Expected results for FAR and FPR optimizations

As an example, we have applied both our new FPR tuning methodology and the existing FAR tuning mode on the tiny test circuit chosen for illustration purposes only and shown in Figure 4. In particular, we compare the results of power reduction obtained with the FPR with the results of power reduction obtained with the FAR tuning mode.

We divide the test circuit into two parts: one part that is switching all the time ($\alpha = 1$) and the other part that is not switching at all ($\alpha = 0$). The main purpose of this example is to exaggerate the effects of input pattern dependency on power-aware tuning and to identify which devices are targeted for optimization when using different optimization modes. We expect that FPR mode will focus only on the part of the circuit that is switching, whereas the FAR mode will target every device equally whether it is switching or not.

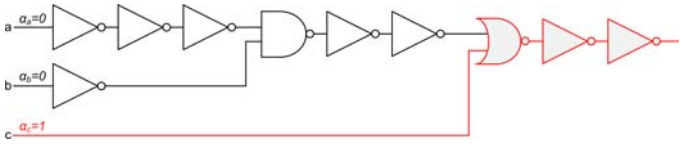

Fig. 4. Test circuit for comparison of FAR and FPR optimization modes

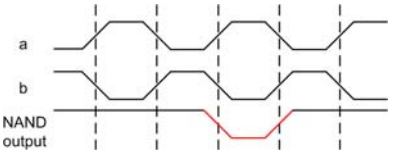

Fig. 5. Glitch-aware tuning

In this example, the results from the test circuit match our expectations. Specifically, Figure 6 shows the timing, power and area comparison of the initial pre-tuned test circuit and both FAR and FPR post-tuned results. Results from both the new and existing tuning methodologies were obtained and verified using existing timing and power tools. Figure 6 shows that the FPR mode was more successful than the FAR mode at minimizing the circuit power and the FAR mode achieved the minimum area as expected (see Figure 3).

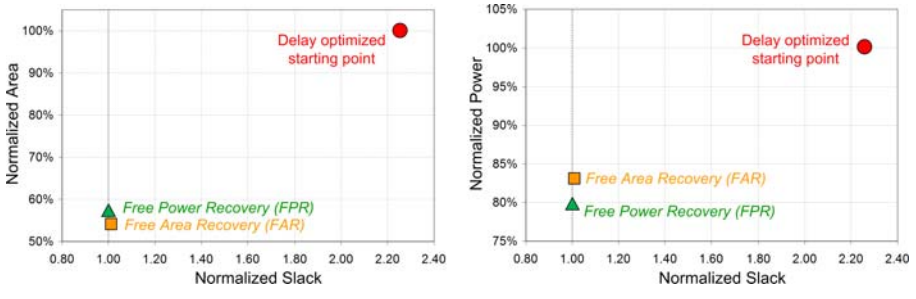

Fig. 6. Power recovery improvement and area comparison of pre-tuned and post-tuned results with FPR over FAR shown for the test circuit from Figure 4

To analyze how the devices were tuned differently in the FAR and FPR tuning modes, we plotted the area comparison of the circuit blocks in Figure 7. A comparison of total device width is plotted for the part of the circuit that is not switching versus the part that is fully-switching. When tuning in the FAR mode, the ratio of total device width of the circuit block that is switching versus the one that is not switching stayed constant as compared to the ratio of respective circuit blocks in the pre-tuned (initial) circuit. As seen from the comparison, the FPR tuning mode properly accounts for the circuit power because it considers only those devices that are switching. The area of the circuit that is not switching ($\alpha = 0$) is allowed to increase and the area of the circuit that is switching ($\alpha = 1$) is reduced in order to simultaneously minimize the power and satisfy the timing requirements. Of course, other constraints keep this from shrinking to zero.

Figure 5 shows that glitch events must also be considered as part of node switching activity in order to correctly account for associated power consumption. In the FPR framework, switching activity accounts for both true and spurious (glitch) transitions, thereby allowing for both power- and glitch-aware circuit tuning.

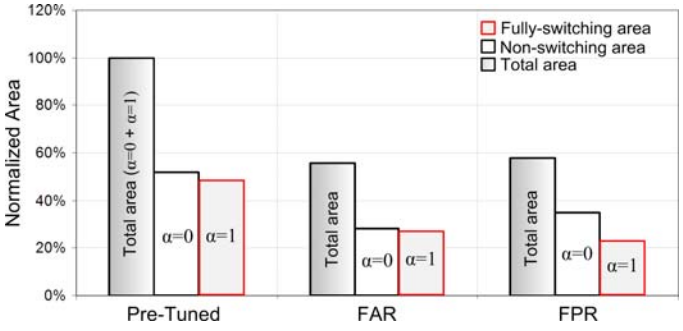

Fig. 7. Comparison of total area and areas of non-switching and fully-switching circuit blocks (marked with $\alpha = 0$ and $\alpha = 1$, respectively) for pre-tuned and post-tuned test circuit from Figure 4 using FAR and FPR optimization modes

4 Implementation Flow

The EinsTuner tool [3] automates the tuning process using state-of-the-art non-linear optimization solvers [7] and fast circuit simulators. We have incorporated the FPR tuning mode in the EinsTuner framework by providing switching activities for each circuit node and their corresponding Leakage Duty Cycle values as inputs to EinsTuner as shown in Figure 8.

We obtain timing information of the pre-tuned circuit using the EinsTLT transistor level timing engine [8]. Our framework includes a power profiler that generates customized input patterns for each circuit. In the analysis and optimization we use several input patterns with different switching activities. Formation of customized input patterns is a complex and effort-intensive process. Good coverage for circuit devices that are switching is also addressed in the pattern formation process. For each input pattern, we use logic simulator and leakage analysis tools [9] [10] to obtain switching activities and LDC information for every device in the circuit.

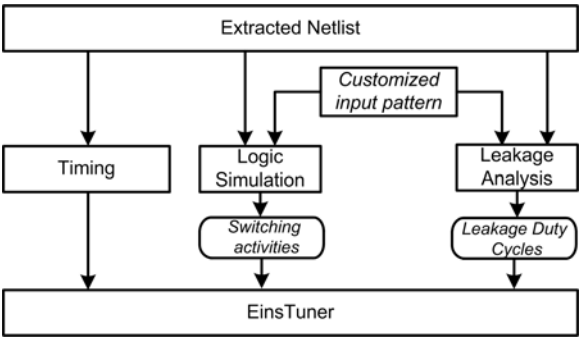

Fig. 8. Tuning implementation flow

5 Results

Our new power-aware tuning methodology (FPR) was applied to the post-layout optimization of an arithmetic block. Maintaining the constant number of fingers was the constraint in our optimization. Even though our analysis was performed to the post-layout optimization, we were able to achieve more than 10% power reduction. Input data patterns with switching activity of 50% were applied. We performed both FAR and FPR tuning optimizations on the test circuit and we compared the optimization results in terms of achieved power reduction. For a given delay target, we performed FAR optimization first and then we performed FPR power-aware tuning. We also analyzed how optimization results varied with and without clock-gating. Note that latches are non-tunable objects and the relative improvement in power reduction is higher in clock-gated runs. The results of FPR vs. FAR tuning for the adder with an input data pattern with 50% switching activity are shown in Figure 9. In both free and clock-gated optimization runs, FPR achieved lowest power consumption at the same delay target. In terms of dynamic (AC) power, FPR was able to achieve 2% more in power reduction when comparing to FAR optimization. In terms of leakage (DC) power, FPR tuning achieved 4% more in power reduction.

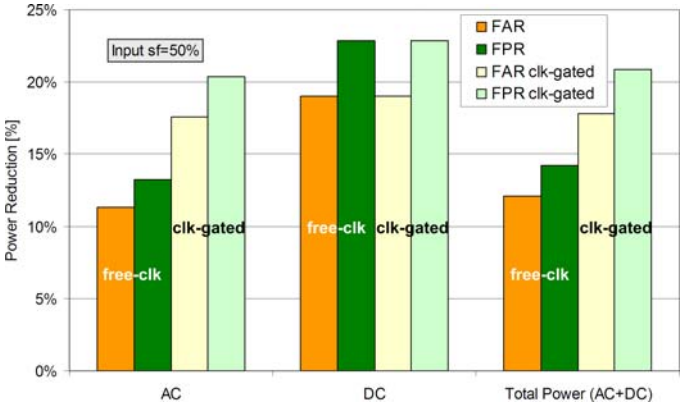

Fig. 9. FAR vs. FPR power reduction comparison on the adder example for an input data pattern with 50% switching activity

6 Conclusion

A new power-aware tuning methodology, Free Power Recovery (FPR), has been introduced. Its application and demonstrated power reduction potential has been evaluated on both a simple example and complex microprocessor circuits. The methodology takes into account switching activities and state-dependent leakage duty cycle properties to tune a design. The goal is to minimize the total power

consumption of a circuit without degrading its delay performance. The FPR optimization mode has been incorporated into the EinsTuner circuit tuning tool. Results from both the new and existing tuning methodologies were obtained and verified using existing timing and power tools. The FPR optimization mode was found to be better at reducing power as compared with the existing optimization modes (such as FAR). The FPR mode appears to be particularly effective for non-symmetrical circuits. Power and timing results of pre-tuned and post-tuned designs were used as comparison criteria.

References

1. Horowitz, M., Stark, D., Alon, E.: Digital Circuit Design Trends. IEEE Journal of Solid-State Circuits 43(4), 757–761 (2008)
2. Oklobdzija, V.G., Krishnamurthy, R.K.: High-Performance Energy-Efficient Microprocessor Design. Series on Integrated Circuits and Systems. Springer-Verlag New York, Inc., Secaucus (2006)
3. Conn, A.R., Elfadel, I.M., Molzen, W.W., O'Brien, P.R., Strenski, P.N., Visweswariah, C., Whan, C.B.: Gradient-Based Optimization of Custom Circuits Using a Static-Timing Formulation. In: DAC, pp. 452–459 (1999)
4. Berridge, R., et al.: IBM POWER6 Microprocessor Physical Design and Design Methodology. IBM Journal of Research and Development 51(6), 685–714 (2007)
5. Narendra, S., De, V., Antoniadis, D., Chandrakasan, A., Borkar, S.: Scaling of Stack Effect and its Application for Leakage Reduction. In: ISLPED 2001: Proceedings of the 2001 International Symposium on Low Power Electronics and Design, pp. 195–200. ACM, New York (2001)
6. BSIM 4.2.1 MOSFET Model: User's Manual, Dept. of EECS, University of California, Berkeley, CA, USA (2002)
7. Wächter, A., Visweswariah, C., Conn, A.R.: Large-Scale Nonlinear Optimization in Circuit Tuning. Future Generation Computer Syst. 21(8), 1251–1262 (2005)
8. Rao, V.B., Soreff, J.P., Brodnax, T.B., Mains, R.E.: EinsTLT: Transistor-Level Timing with EinsTimer. In: Proceedings of the ACM/IEEE 1999 International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (Tau 1999), pp. 1–6 (1999)
9. Bard, K., et al.: Transistor-Level Tools for High-End Processor Custom Circuit Design at IBM. Proceedings of the IEEE, invited paper (March 2007)
10. Neely, J.S., et al.: CPAM: A Common Power Analysis Methodology for High-Performance VLSI Design. In: IEEE Conf. Electrical Performance of Electronic Packaging, pp. 303–306. IEEE Press, Los Alamitos (2000)