

The Challenges of Implementing Fine-Grained Power Gating

Anja Niedermeier^{* †}
A.Niedermeier <at>
ewi.utwente.nl

Kjetil Svarstad
Kjetil.Svarstad <at>
iet.ntnu.no

Norwegian University of Science and Technology (NTNU)
Trondheim, Norway

Frank Bouwens

Jos Hulzink

Jos Huisken

<firstname> . <lastname> <at> imec-nl.nl
Holst Centre / Stichting IMEC-NL
Eindhoven, The Netherlands

ABSTRACT

Power consumption in digital systems, especially in portable devices, is a crucial design factor. Due to downscaling of technology, dynamic switching power is not the only relevant source of power consumption anymore as power dissipation caused by leakage currents increases. Even though power gating is a seemingly simple method for reducing the leakage power, the implications of introducing power gating to a design have to be analyzed in detail. We present an extensive analysis of the impact of fine-grained power gating on the overall power consumption. The presented results are based on the analysis of an actual implementation of power gating in the datapath of a very long instruction word (VLIW) processor. The extracted power consumption values clearly demonstrate that the overhead of power gating is, in contrary to the analysis found in previous publication, not determined by the energy required to switch a power domain on. Rather, it is determined by the energy consumption of additionally required modules. We show that, for the break-even point case, about 2/3 of the energy overhead is caused by the isolation cells, about 1/3 by the control modules, and only roughly 1% by the energy to switch a power domain on.

Categories and Subject Descriptors

B.5.2 [Hardware]: Register-Transfer-Level Implementation—*Design Aids*; C.4 [Computer Systems Organization]: Performance of Systems—*Modeling techniques*

*This work was performed at IMEC-NL

†Is now with the University of Twente, The Netherlands

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GLSVLSI'10, May 16–18, 2010, Providence, Rhode Island, USA.
Copyright 2010 ACM 978-1-4503-0012-4/10/06 ...\$10.00.

General Terms

Design, Estimation, Power, Implementation

Keywords

Power Gating, Leakage power minimization, Power modeling, Register-Transfer-Level, Analysis, Power management

1. INTRODUCTION

In recent years the demand for dedicated hardware targeted for a specific task or application domain has increased. One possibility to implement such dedicated hardware is the use of *application specific instruction set processors* (ASIPs). They include dedicated function units (FUs) and registers designed for a specific purpose. However, in many cases those FUs and the registers are idle for long periods of time. To minimize power consumption during those idle periods, usually clock gating is applied. However, due to downscaling of technology, leakage power consumption is gaining more and more impact on the total power dissipation in both absolute numbers [4] as well as in power consumption per area [6]. One promising method to minimize leakage power is *power gating* [5, 2], a method where idle blocks are disconnected from the power supply, hence minimizing the leakage power. Creating designs with power gating requires a thorough analysis of the system including the introduced overhead.

Some research has been done in the domain of fine grained power gating and its break-even point. In [3], an exploration of the potential of power gating execution units in the datapath is performed. Also, an analytical equation for the break-even point is derived where the authors assume the power consumed by the power switch to be the only source for the energy overhead. The authors of [4] also perform an analysis of the break-even point for power gating. They include besides the power switch also additionally required decoupling capacitor (*decap*) area in their model. In [8], an implementation methodology for power gating and an analysis of the overhead are presented. The authors base their methodology on exploiting existing clock-gating control signals. In their analysis of the overhead, they only consider

the power switch. In [7], a more detailed analysis of power gating than in the previous papers is presented. The authors include in their analysis of the break-even point the leakage power savings, power mode transition energy, sleep transistor size, performance degradation, and power mode transition time.

In the work underlying this paper, we analysed the impact of power gating with respect to the complete system, including the energy overhead due to additionally required modules like isolation cells or a power manager and derived an equation for the break-even point. Our discussion is based on power figures obtained for *typical case* conditions (1.2V, 25°C) after place and route (P&R) for 90nm TSMC technology of a processor with fine-grained power gating implemented in the datapath.

2. POWER GATING

A general overview of a power gated system is illustrated in Figure 1. In the system, two power domains (*PD*) are present, where a *PD* is defined as a part of the system which is connected to a common power supply. A *PD* which can be switched off is represented by *PD_switchable*. It has outputs to another *PD* (*PD_always_on*) which is connected to an always-on voltage supply in this example. The outputs of *PD_switchable* are isolated to prevent unknown signals propagating through the design when *PD_switchable* is switched off. A power switch is inserted between the voltage supply *VDD* and *PD_switchable* in order to allow *PD_switchable* to be disconnected from *VDD*. Furthermore, a power manager is integrated into the system to control the power gating procedure.

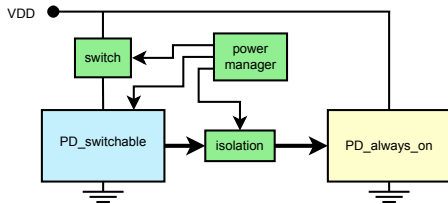


Figure 1: General power gating scheme

3. BREAK-EVEN POINT

In Figure 2, the power dissipation over time of a power domain is depicted. The total energy consumption is composed of the energy which is consumed by the modules within a power domain and the energy which is consumed by the additional power gating related modules. The modules which are constantly active, like a power manager, are consuming energy during the complete runtime ($E_{add.modules}$). This energy is determined by the power consumption of the modules ($P_{add.modules}$) multiplied by the total run-time (t_{total}). The energy consumption of the remaining components depends on the state of the power domain which is described in more detail in the following.

During t_{active} , the system is in its active state. The energy in this state is consumed by the modules within a power domain ($E_{mod,active}$), the isolation cells ($E_{iso,active}$) and the SR registers which consume more energy than regular registers ($\Delta E_{SR,active}$).

At t_{idle} , the power domain has finished the active state and goes immediately to the idle state by switching off the clock. At the same time, the power domain is powered off. The powering off process takes place until t_{off} . During that time, the power domain still leaks but the leakage energy ($E_{mod,leak,on}$) is converging to the off-level.

Then, the power domain remains switched off during t_{down} . The energy which is consumed depends on the leakage of the power switch(es) ($E_{switch,leak}$), the leakage of the isolation cells ($E_{iso,leak}$) and the leakage of the SR registers ($E_{SR,leak}$).

At t_{sleep} , the power domain is switched on again. The switching-on process takes until t_{on} . The energy which is consumed during that period is $E_{mod,leak,on}$ and the additional energy required to switch on the power domain ($E_{poweron}$). Afterwards, the power domain is fully functional.

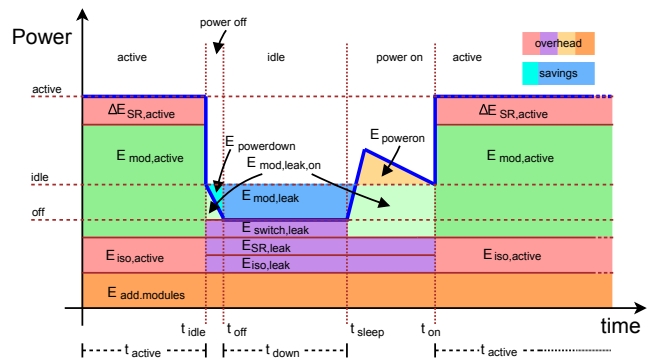


Figure 2: Power over time

The energy savings of a power domain are determined by the leakage power which the modules within a power domain would consume if not switched off ($E_{mod,leak} = P_{mod,leak} \cdot t_{down}$ where $P_{mod,leak}$ is the leakage power of the modules within a power domain) and $E_{powerdown}$, which is the difference between the leakage energy the power domain would normally consume between t_{idle} and t_{off} and the energy which is still consumed ($E_{mod,leak,on}$). The total energy savings are defined as follows:

$$E_{savings} = P_{mod,leak} \cdot t_{down} + N \cdot E_{powerdown} \quad (1)$$

with N defining the number of transitions from *on* to *off*.

The energy overhead ($E_{overhead}$) is determined by the additional energy consumption during the different states, as explained above. Summarizing, the energy overhead can be written as follows:

$$\begin{aligned} E_{overhead} = & t_{down} \cdot (P_{switch,leak} + P_{iso,leak} + P_{SR,leak}) \\ & + t_{active} \cdot (P_{iso,active} + \Delta P_{SR,active}) \\ & + t_{total} \cdot P_{add.modules} \\ & + N \cdot E_{poweron} \end{aligned} \quad (2)$$

where $P_{switch,leak}$, $P_{iso,leak}$ and $P_{SR,leak}$ represent the leakage power consumption of the switches, the isolation and the SR registers. $P_{iso,active}$ is the power consumption of the isolation cells during active mode and $\Delta P_{SR,active}$ represents the additional power consumed by the SR registers compared to what normal registers would consume during

active mode. N defines the number of transitions from *on* to *off*.

Merging the individual factors leads to the following equation:

$$E_{overhead} = t_{down} \cdot \beta + t_{active} \cdot \gamma + t_{total} \cdot \delta + \epsilon \quad (3)$$

with $\beta = P_{switch,leak} + P_{iso,leak} + P_{SR,leak}$, $\gamma = P_{iso,active} + \Delta P_{SR,active}$, $\delta = P_{add.modules}$ and $\epsilon = N \cdot E_{poweron}$

Building on this analysis, a formula can be derived for the minimum fraction of time that the power domain has to be switched off in order to gain energy savings, i.e. the break-even point at which the energy savings are bigger than the energy overhead:

$$E_{savings} > E_{overhead} \quad (4)$$

By using the above definitions for the savings and the overhead, and expressing t_{active} with $t_{total} - t_{down}$, a condition for t_{down}/t_{total} can be found. Also, some of the factors of the above analysis can be omitted because they are negligible, namely $P_{switch,leak}$, $P_{iso,leak}$, $E_{poweron}$ and $E_{powerdown}$. This will be shown in Figure 5 in Section 5 where the results are presented. Summarizing, the condition for the minimum down time in relation to the total time is:

$$\frac{t_{down}}{t_{total}} > \frac{\gamma + \delta}{\gamma + \alpha - \beta'} \quad (5)$$

where $\alpha = P_{mod,leak}$, $\beta' = P_{SR,leak}$, $\gamma = P_{iso,active} + \Delta P_{SR,active}$ and $\delta = P_{add.modules}$

4. IMPLEMENTATION

The design which was used in this work is an improved version of an VLIW (very long instruction word) processor which was designed for ultra wide band (UWB) purposes and presented in [1]. It consists of one scalar issue slot, one issue slot for both scalar and vector operations, and one issue slot for vector operations only.

For this work, the following applications were used: first, the UWB receiver application as described in [1] is executed, consisting of a *Synchronization / Timing Acquisition Phase* with a subsequent *Payload demodulation phase*. Afterwards, a data decompression algorithm based on the discrete wavelet transform (DWT) algorithm is executed.

The analysis of the power consumption of the modules of the processor and their utilisation during the application led to a partitioning into three power domains: PD_{vec} which includes a vector adder, PD_{mul} which includes a scalar multiplier and PD_{VIS} which includes the complete vector issue slots and the vector registers. PD_{vec} has 96 output signals and consists of 900 gates, PD_{mul} has 32 output signals and consists of 1200 gates, and PD_{VIS} is the largest power domain with 144 output signals and 13700 gates. The proposed power-off scheme is as follows: At startup, i.e. before the UWB receiver application is executed, PD_{mul} and PD_{VIS} are switched on and PD_{mul} is switched off as it is not needed. After the *Synchronization / Timing Acquisition Phase*, PD_{vec} can be switched off. After executing the *Payload demodulation phase*, the UWB application is finished and the DWT application is executed. For that, PD_{VIS} is also switched off and PD_{mul} is switched on.

To evaluate the difference between a hardware (HW) based and a software (SW) based power manager, both methods were implemented. In the implementation using the HW based power manager, one power manager is instantiated

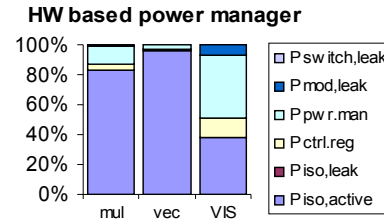


Figure 3: Results for the HW based power manager

per power domain. The power managers are controlled by a control register which was added to the processor. It contains one bit per power domain, where a '1' indicates that the power domain is shut off, otherwise it is on. To access single bits of the register, an additional function unit was implemented. For the SW based power manager, the control signals to the power gating related cells are determined directly from the control register. The control register contains one bit for *poweroff* and one bit for *isolation* for each power domain. In contrast to the first implementation, for this approach, the control register is not implemented bitwise but can only be accessed completely. This was done so that clock gating can reasonably be applied to the control register.

The resulting design was synthesized and placed and routed with 90nm TSMC LP (low power) library for 100 MHz with the Cadence design tools. The power domains were defined with the *common power format* (CPF) during the design flow.

5. RESULTS AND DISCUSSION

The hardware based power manager.

In Figure 3, the results for the power consumption distribution of the power gating related components (introduced as α , β , γ and δ in Section 3) of the hardware based power manager implementation are depicted. The denotation corresponds to the definitions used in Section 3. The terms $P_{pwr.man}$ and $P_{ctrl.reg}$ form $P_{add.modules}$ (δ). They represent the power consumed by the power manager and the control register, respectively.

It can be seen, that for the power domains PD_{mul} and PD_{vec} the overhead is dominated by the power consumed by the isolation cells during active mode ($P_{iso,active}$). The overhead for the power domain PD_{VIS} is caused by the isolation cells during active mode ($P_{iso,active}$) and the power manager ($P_{pwr.man}$) to approximately equal parts, a small part is due to the control register ($P_{ctrl.reg}$). Also it is noticeable that PD_{VIS} is the only power domain with a leakage power consumption ($P_{mod,leak}$) which is significant enough to be visible in the graph.

The minimum t_{down}/t_{total} , calculated using Equation 5, are 1.19 for PD_{mul} , 1.04 for PD_{vec} and 2.04 for PD_{VIS} . That implies, that in theory the respective power domain had to be switched off for more than 100 % of the total time in order to save energy, which in practice is not possible. That means that a benefit from power gating cannot be reached. The results demonstrate clearly, that power gating would cause extra energy consumption in the system as the introduced energy overhead will always exceed the energy savings.

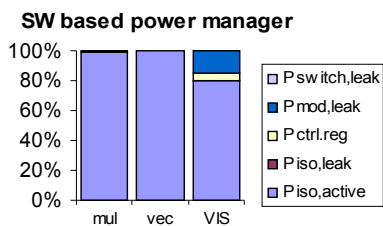


Figure 4: Results for the SW based power manager

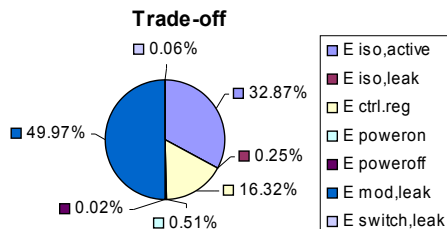


Figure 5: Energy distribution for PD_VIS for the software based power manager

The software based power manager.

As the power manager has shown to be a significant contributor to the overhead, the system was implemented using a software based power manager. The power breakdown of α , β , γ , and δ is depicted in Figure 4. The results demonstrate that the overhead is mainly caused by the isolation cells during active mode ($P_{iso,active}$). For PD_VIS , additionally the power consumed by the control register ($P_{ctrl.reg}$) has a small influence.

The minimum t_{down}/t_{total} are 1.00 for PD_mul and PD_vec . That means, that for those cases energy savings can never be obtained as the overhead will always be at least as big as the savings. However, for the power domain PD_VIS it is 0.89, which means that benefits could be gained when PD_VIS is switched off for more than 89 % of the time. To analyze the distribution of the energy overhead and savings for the case that overhead and savings outweigh each other exactly, i.e. PD_VIS is switched off for 89 % of the time, the contributors are depicted in Figure 5.

The graph shows that the savings (on the left side of the graph) are dominated by the leakage power consumed by the modules within the power domain. The energy which is consumed during powering down (introduced as $E_{powerdown}$ in Section 3) is negligible. The overhead (right side of the graph) is caused mainly by the energy consumption of the isolation cells during active mode ($E_{iso,active}$) and the additional control modules ($E_{add.modules}$), in this case the power gating control register ($E_{ctrl.reg}$). The leakage of the switch ($E_{switch,leak}$) and the isolation cells ($E_{iso,leak}$) are marginal. Also the energy to switch a power domain on ($E_{poweron}$) is negligible which was surprising considering previously published studies.

A surprising observation for both implementations is the large difference in power consumption of the isolation cells between the power domains. For PD_vec it is a factor of almost 40 compared to PD_VIS , which has 1.5 times as many output signals. This is caused by the fact that the isolation block of PD_vec are in a critical path in the design, therefore extra buffers were inserted to meet timing constraints. Con-

sequently, the additional buffers also increased the power consumption.

Summarizing, the obtained results show the following: The power domains need a low duty cycle, otherwise the energy overhead will exceed the savings. The size of the power domain is of importance as it has a direct impact on the leakage power consumption which dictates the savings. The number of outputs is relevant as it determines the number of isolation cells. Finally, the power management is a significant contributor to the energy overhead.

6. CONCLUSIONS

In this paper, a detailed analysis of the break-even point for power gating was presented. Furthermore, two different implementations of fine-grained power gating in the datapath of a VLIW processor were shown. Both implementations demonstrated that the introduced energy overhead is significant.

Surprisingly, the results demonstrated that the overhead due to the energy required to switch a power domain on, which has been stated as main contributor in the literature, is very small compared to the overhead is caused by additional modules, primarily the isolation cells at the boundaries of the power domains and additional control modules like a dedicated power manager or a control register. As the analyzed power domains had a typical utilization profile and the isolation cells are a mandatory part of power gating, these can be considered as new power gating constraint.

Even though the energy overhead could be reduced significantly after omitting the need for a hardware-based power manager by switching to a software based solution, the obtained results showed, that fine grained power gating in the datapath of a processor hardly can gain benefits, as the leakage energy which could be saved during idle is too low compared to the introduced energy overhead during active mode.

7. REFERENCES

- [1] C. Bachmann, A. Genser, J. Huzink, M. Berekovic, and C. Steger. A Low-Power ASIP for IEEE 802.15.4a Ultra-Wideband. In *Design, Automation and Test in Europe (DATE) 2009 Proceedings*, 2009.
- [2] J. Frenkil and S. Venkatraman. Power Gating Design Automation. chapter in *Closing the Power Gap between ASIC and Custom*, Springer, 2007.
- [3] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose. Microarchitectural techniques for power gating of execution units. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 32–37, 2004.
- [4] H. Jiang, M. Marek-Sadowska, and S. Nassif. Benefits and costs of power-gating technique. In *2005 IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings*, pages 559–566, 2005.
- [5] M. Keating. *Low Power Methodology Manual for System-on-chip Design*. Springer, 2007.
- [6] G. Panic, Z. Stamenkovic, and R. Kraemer. Power gating in wireless sensor networks. In *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, pages 499–503, 2008.
- [7] A. Sathanur, A. Calimera, A. Pullini, L. Benini, A. Macii, E. Macii, and M. Poncino. On quantifying the figures of merit of power-gating for leakage power minimization in nanometer CMOS circuits. In *IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008*, pages 2761–2764, 2008.
- [8] K. Usami and N. Ohkubo. A design approach for fine-grained run-time power gating using locally extracted sleep signals. In *Computer Design, 2006. ICCD 2006. International Conference on*, pages 155–161, 2007.