Spatial and Temporal Granularity Limits of Body Biasing in UTBB-FDSOI

Johannes Maximilian Kühn†, Dustin Peterson‡, Hideharu Amano*, Oliver Bringmann‡, Wolfgang Rosenstiel†

†University of Tübingen
Computer Engineering
Sand 13, 72076 Tübingen
Germany

*Keio University
Department of ICS
3-14-1 Hiyoshi, Kohoku-ku, Yokohama
Japan

‡University of Tübingen
Embedded Systems
Sand 13, 72076 Tübingen
Germany

Email: muccra@am.ics.keio.ac.jp

***Abstract*—Advances in SOI technology such as STMicro's 28nm UTBB-FDSOI enabled a renaissance of body biasing. Body biasing is a fast and efficient technique to change power and performance characteristics. As the electrical task to change the substrate potential is small compared to Dynamic Voltage Scaling, much finer island sizes are conceivable. This however creates new challenges in regard to design partitioning into body bias islands and body bias combinations across such designs. These combinations should be chosen so that energy efficiency improves while maintaining timing constraints. We introduce a combination based analysis tool to find optimized body bias island partitions and body biasing levels. For such partitions, optimized body bias assignments for static, programmable and dynamic body biasing can be computed. The overheads incurred by dynamically switching body biases are estimated to yield actual improvements and to give an upper bound for the power consumption of required additional circuitry. Based on these partitionings and the switching overheads, optimized application specific switching strategies are computed. The effectiveness of this method is demonstrated in a frequency scaling scenario using forward body biasing on a Dynamic Reconfigurable Processor (DRP) design. We show that leakage can be greatly reduced using the proposed methods and that dynamic body biasing can be beneficial even at small time periods.**

I. Introduction

The growing demand in performance with unchanging power budgets keeps on driving technology scaling. With scaling came multiple problems, such as Short Channel Effects (SCEs) and increasing leakage currents. New process technologies like FinFET or FDSOI were developed to mitigate these problems. While SCEs immunity has been improved, the portions on a chip that can be run concurrently still becomes smaller with each generation. Mission profiles on the other hand demand ever greater flexibility in terms of peak performance and minimal power consumption in idle or sleep modes. In advanced SOI processes such as STMicro's 28nm UTBB-FDSOI, an old transistor tuning knob has been reintroduced. Body biasing (BB), i.e. the application of a potential V_{bb} on the transistor substrate, allows to alter transistor threshold voltages. By controlling V_{bb}, this process allows to realize dynamic multi-V_{th} designs. Using this approach, leakage can be reduced in idle paths and in timing uncritical paths. Whenever performance is required, it can be used to raise maximum clock frequencies without increasing supply voltage V_{dd}, omitting the quadratic relationship of V_{dd} with dynamic power consumption. Thus idle power consumption can be greatly reduced while also increasing energy efficiency in high load scenarios. Controlling V_{bb} however comes at certain power and area cost. Especially when changing V_{bb} dynamically, it is critical that the break even point (BEP) has been reached. In this study, we present a method to find the spatial and temporal limits of BB. Three ways to apply this technique are examined. First static BB, i.e. directly connecting substrate pins to the voltage source. The second examination focusses on programmable BB, i.e. V_{bb} set once per application. Finally, dynamic BB, meaning V_{bb} may change during execution. The method is based on building and evaluating combinations of V_{bb} across combinations of partitions with individual V_{bb}, i.e. body bias islands (BBI). It also considers application traces to find programmable and dynamic BB assignments. As a case study, we examined a DRP and show up to which spatial and temporal granularity BB yields benefits in this design.

II. State of the Art and Contributions

A technology case-study important for this work is [1]. The authors implemented a dual-core ARM Cortex A9 based SoC in STMicro's 28nm UTBB-FDSOI process. The study shows that by employing forward body biasing (FBB), it is possible to raise maximum clock frequency by almost 2x from $550MHz$ to $1050MHz$ while staying at $0.61V$ supply voltage. In [2], the authors describe a dynamic BB approach with an optimized piece-wise convex set based Dynamic Voltage and Frequency Scaling (DVFS) strategy. The examined design is a Digital Signal Processor consisting of several Processing Elements (PEs). Each PE constitutes a BBI. Results were obtained using a test chip manufactured in 28nm UTBB-FDSOI process. The proposed approach yields a total power saving ranging from 8.18% to 17.31%. On temporal and spatial granularity limits of power domains, [3] examines technological limits of DVFS. The authors specify a mathematical framework to analyze technologically driven limits as well as the performance of DVFS strategies. [4] present a formal method to partition a design into several power domains for power gating. Thereby incurred overheads are not considered. In regard to DRPs, [5] describes Dynamic Voltage Switching on PE granularity. While the reduction of power consumption did not outweigh the switching cost with unchanged application mappings, the authors showed that power consumption can be reduced up to 13% when adapting the application. This optimization aims to reduce the switching frequency. Using a similar semiconductor technology called Silicon on Thin BOX, [6] demonstrates significant clock frequency boosting at ultra low supply voltages. The evaluated design consists of a

978-3-9815370-4-8/DATE15/©2015 EDAA

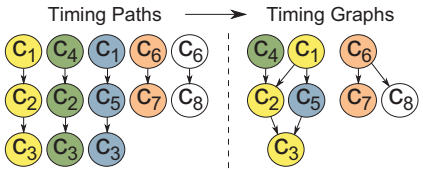

Fig. 1. Example for the generation of timing graphs from timing paths

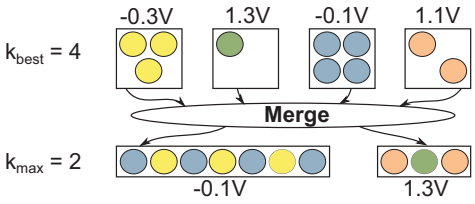

Fig. 2. Merging k_{best} islands into k_{max} islands

Coarse-Grained Reconfigurable Architecture and a small micro controller. Through the application of FBB, a clock frequency of $89MHz$ can be reached at a supply voltage of $0.4V$.

Our work contributes i) a method to evaluate fine-granular BB which is capable to find ii) optimized partitionings of designs to minimize leakage and iii) to find optimized BB switching strategies for such partitionings. We do this while taking the effort to switch the BB into account, thus iv) stating upper bounds of power consumption for the additional circuitry up to which energy efficiency benefits from these efforts.

III. Methodology and Implementation

In this section we focus on the formulation of BB optimization problems, the implementation of algorithms for solving these problems and determining appropriate BBIs for a given hardware design and an application running on this architecture.

A. Modeling Aspects

Given: A hardware design $D = (C, TP, V_d, V_b)$ with a set $C = \{c_1, ..., c_N\}$ of N components, a set $TP = \{tp_1, ..., tp_M\}$ of M component-level timing paths ($\forall tp_i | tp_i \subseteq C$), a set $V_d = \{V_{dd_1}, ..., V_{dd_Q}\}$ of Q available supply voltages (in Volt) and a set $V_b = \{V_{bb_1}, ..., V_{bb_R}\}$ of R available body bias voltages (in Volt).

Given: The leakage currents (in Ampere) $CL(x, y, z) = \{cl_{1,1,1}, ..., cl_{N,R,Q}\}$ and delays (in seconds) $CD(C(x), V_d(y), V_b(z)) = \{cd_{x,y,z}, ..., cd_{N,R,Q}\}$ of each component c_x, supply voltage $V_{dd_y} \in V_d$ and body bias voltage $V_{bb_z} \in V_b$.

Given: The substrate charges (in Coulomb) $SC(x) = \{sc_1, ..., sc_N\}$) of each component c_x.

Given: An application $A = a_1, ..., a_Z$ with Z operations where each $a_i \subseteq TP$ specifies the relevant timing paths for the given operation.

BB Layout: A BB Layout consists of disjoint component subsets (islands) and correlated sets of V_{bb}.

B. Data Extraction

In order to obtain the data required for the above described modeling, different tools and flows were employed. To extract the design structure, each Verilog module was assigned a component $c_i \in C$. If c_i contained implementations for different and independently executed functionality, e.g. different arithmetic functions, c_i has been split up into c_{ij}, storing c_{ij} instead of c_i. Then for each $c_i \in C$, a layout is synthesized. Using a layout extracted netlist, leakage CL for every V_{bb_r} and substrate charge SC is determined using SPICE simulations. Delays CD have been obtained through static timing analysis of the top module. For each $c_i \in C$, the delay of its path is stored. All technology dependent values have been obtained at a temperature of $125°C$. The set of timing paths TP is determined through the design hierarchy.

C. Static Body Biasing

First, we are focusing on Static Body Biasing. Here, a BB layout uses the same mode for all applications.

Optimization Problem 1: Partition C into k_{max} disjoint subsets (islands) $P = p_1, ..., p_{k_{max}}$ with $p_i \subseteq C$. An optimum BB voltage $V_{bb} = V_{bb_1}, ..., V_{bb_{k_{max}}}$ with $V_{bb_i} \in V_b$ is selected for each p_i such that the leakage power is minimal. Leakage Power P_L is defined as follows:

$$P_L = V_{dd_x} \cdot \sum_{i=1}^{k_{max}} \left(\sum_{c_j \in p_i} CL(c_j, V_{dd_x}, V_{bb_i}) \right).$$

Furthermore a maximum path delay d_{cycle}, which is equal to the clock cycle period, has to be observed on all paths:

$$\forall_{tp_i \in TP} : \sum_{p_j \in P} \left(\sum_{c_k \in p_j \wedge c_k \in tp_i} CD(c_k, V_{dd_x}, V_{bb_j}) \right) \leq d_{cycle}$$

The given problem cannot be efficiently solved by a combination-based approach. The number of possible combinations for a layout with maximum K BB Islands, N components and Q BB voltages is $K^N \cdot Q^K$, which is hard to compute – e.g. for a small number of $N = 17$ components, $Q = 17$ different BB Voltages ($-0.3V$ to $1.3V$) and $K = 4$ islands we would need to evaluate 17 billion combinations.

Therefore we propose a two-step approach also allowing the computation for larger problem sizes. First, we need to determine the minimum-leakage BB layout, disregarding the maximum number of islands. Thus, we convert the set of timing paths TP into a set of timing graphs TG, which are isolated sets of components that do not have components in common with any other timing graph. This approach allows to locally optimize graphs leading to a global design optimum. The conversion step is illustrated in figure 1. The absolute minimum leakage layout with k_{best} is determined as follows:

```
1:  for all G in TG do
2:      minLeak ← inf, best ← null
3:      maxTP ← longest timing path in G (V_bb = 0.0V)
4:      for all V_bb combinations in maxTP do
5:          Calculate min. V_bb for each comp. in G\maxTP
6:          to fulfill timing requirements
7:          leak ← Leakage of solution
8:          if leak < minLeak then
9:              minLeak ← leak, best ← solution
10:         end if
11:     end for
```

12: **end for**

Subsequently, k_{best} islands are merged into k_{max} new islands iteratively, setting the V_{bb} of the new partition to the highest value (to ensure timing correctness) of the corresponding merged islands, which is illustrated in figure 2. Each merge combination is evaluated and the lowest leakage solution is stored as the final BB layout with k_{max} islands.

D. Programmable Body Biasing

Programmable Body Biasing extends Static BB with switching capabilities across different applications. An application A is specified as a list of operation cycles. Each operation cycle collects information about the relevant and irrelevant paths in a design – e.g., for an addition, the paths for multiplication are irrelevant while those for addition are relevant. This information is used for the application-specific Programmable BB. We will cut paths that are irrelevant for an application, which allows to lower V_{bb}.

Optimization Problem 2: Partition C into k_{max} disjoint subsets $P = p_1, ..., p_{k_{max}}$ with $p_i \subseteq C$. An optimum BB Voltage $V_{bb} = V_{bb_1}, ..., V_{bb_{k_{max}}}$ with $V_{bb_i} \in V_b$ is selected for each p_i such that the leakage power is minimal for a given application A and V_{dd_x}. Furthermore a maximum path delay d_{cycle} has to be observed on the relevant timing paths:

$$\forall_{a_g \in A} \forall_{tp_i \in a_g} : d \leq \sum_{p_j \in P} \left(\sum_{c_k \in p_j \wedge c_k \in tp_i} CD(p_k, V_{dd_x}, V_{bb_j}) \right)$$

For solving problem 2, we use the BB Islands identified for Static BB, which allows the minimization of leakage even if all operations are used in an application run. For the determination of the optimum BB mode for the given islands, we first execute the Static BB algorithm with one exception: The maximum timing path for each graph G in TG has to be a relevant timing path in application A (so $maxTP \leftarrow$ longest timing path in $G \cap A$). Finally, the optimum for each component is used to generate the Programmable BB mode for application A.

1: **for all** p in P **do** ▷ Determine island layout
2: $V_{bb}[p] \leftarrow -infinity$
3: **for all** $comp$ in P **do**
4: $V_{bb}[p] \leftarrow max(V_{bb}[p], best[comp])$
5: **end for**
6: **end for**

E. Dynamic Body Biasing

Dynamic Body Biasing extends Programmable BB by further switching capabilities among application cycles. In contrast to Programmable BB, we need to take the mode switching costs into account here. Therefore we optimize towards energy while the previous ones are power optimizations.

Optimization Problem 3: Partition C into k_{max} disjoint subsets $P = p_1, ..., p_{k_{max}}$ with $p_i \subseteq C$. For a given application A and supply voltage V_{dd_x}, each application cycle a_y selects an optimum BB Voltage $V_{bb} = V_{bb_{y,1}}, ..., V_{bb_{y,k_{max}}}$ with $V_{bb_{x,i}} \in V_b$ for each p_i such that the energy E (disregarding dynamic energy) is minimal for the application. Furthermore the maximum path delay d_{cycle}, which is equal to the clock cycle period, has to be observed on the relevant timing paths. The energy E is defined as follows:

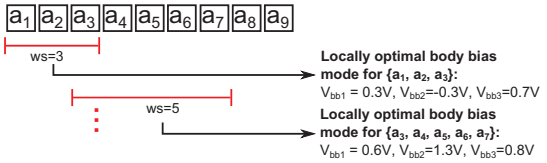

Fig. 3. Example for interval-based local optimum search

$$E = E_{leak} + E_{bbSwitch}$$
$$E_{leak} = d_{cycle} \cdot V_{dd_x} \cdot \sum_{a_y \in A} \sum_{p_i \in P} \sum_{c_j \in p_i} CL(c_j, V_{dd_x}, V_{bb_{y,i}})$$
$$E_{bbSwitch} = \sum_{y=1}^{Z-1} \sum_{p_i \in P} \sum_{c_j \in p_i} |V_{bb_{y,i}} - V_{bb_{y+1,i}}| \cdot SC(c_j)$$

For reducing E, we also use the resulting BBIs of Static BB. In order to determine an optimum switching strategy, a brute force algorithm would be appropriate, but leads to the same issues we had before: The number of switching possibilities $R^{k_{max} \cdot Z}$ increase exponentially.

Therefore, we propose two adjustments to the brute force algorithm to reduce the number of switching possibilities. The first adjustment is an obvious one: Given an application A, we shrink consecutive cycles a_i to a_n to a compressed cycle a_j, if $a_i = a_{i+1} = ... = a_n$, while storing the number of compressions $n - i + 1$. This lossless compression may reduce the number of possibilities a lot. Our second adjustment reduces the number of different BB modes: We create multiple windows with size 1 to Z and shift these windows from the first application cycle to the right, collecting all operations in each window and computing the optimum BB mode for this window. We add the local optimum modes to each operation of the window. This step is is illustrated schematically in figure 3. Additionally, we add intermediate modes, having BB modes in between the modes of neighboring windows of size 1 in order to have little bit leakier modes with lower switching costs to neighbored modes. We build all combinations of all modes for all cycles, evaluate the energy E and select the sequence of BB modes with the minimum energy as optimum BB strategy for the given application A.

F. Implementation

We implemented the proposed methods in a Java- and SQL-based toolset. The model, including hardware design, leakage data and component delays, is stored in a relational database using MySQL. The shell interpreter allows dynamic loading of designs, specifying several optimization criteria (target V_{dd}, maximum path delay d_{cycle}, maximum number of partitions k_{max}) and executing the proposed optimization algorithms. It includes the output of several results, such as the partition layout itself, leakage per mode for a given partition layout and the BB strategy for a given application. The results in the following sections are generated using this tool.

IV. Results

In our case study we evaluated BB partitioning in a frequency scaling scenario. The target design is a performance centric DRP [7]. The examined application is a 8-Tap Finite Impulse Response Filter. Supply voltage and clock is shared across the design. By choosing the maximum clock frequency

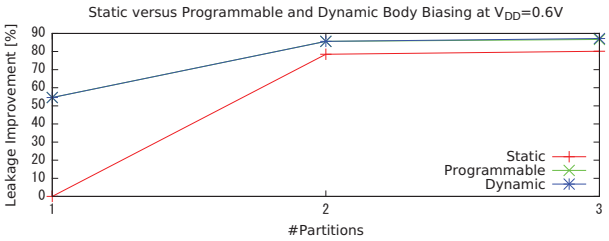

Fig. 4. Benefit over static single BBI BB at $V_{dd} = 0.6V$

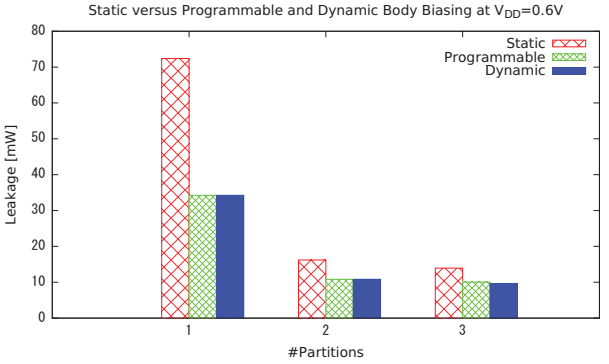

Fig. 5. Leakage per method and number of partition for $V_{dd} = 0.6V$

as $F_{max,BB} = F_{max} \cdot ND(max_{fbb}, V_{dd})$, with ND, the normalized delay at a given V_{bb} at unchanged supply voltage V_{dd}, clock frequency is scaled up to the maximum frequency where the critical path can be accommodated using maximum FBB. Static BB assigns each Processing Element (PE) the same hard wired V_{bb}, where programmable BB allows a per PE setting of V_{bb} per application. Dynamic BB is based on programmable BB, but allows each PE to change its V_{bb} during execution. Fig. 4 and 5 indicates that introducing a single programmable BBI already reduces leakage by over 50%. This is because in the static case, BB is always set to the maximum FBB to be able to meet every timing requirement. Thus the static case with one partition is equal to whole chip BB. Furthermore Tab. I reveals that for 1 and 2 partitions, dynamic BB is equal to programmable BB. This is because changing the substrate charge would require more power than could be saved in this brief period. At 3 partitions however, the effort to change the BB has become small enough to reach the BEP as the partition size decreased. BB approaches further improve when increasing the number of partitions to two. In this case, even the static approach reduces leakage by over 70%. This is because only a small part of a PE is so timing critical to require full FBB. For the remaining components a BB tradeoff is sought, maintaining timing constraints with only moderate FBB. Programmable and dynamic BB can improve upon this by setting V_{bb} for each PE individually, assigning FBB to only those PE partitions that need it. Additional partitions achieve only little benefit in this design. As stated in Tab. I, even in such a small design dynamic BB may yield about $5mW$ less leakage than programmable BB.

V. Conclusion and Outlook

Using BB in a beneficial way is complex but desirable. This study showed that the additional complexity can be eliminated by automation. Even in small designs as has been evaluated in this study, dynamic BB has been shown to be profitable. This hints that for bigger, more heterogenous designs where FBB on big portions would be otherwise prohibitive, much greater improvements are to be expected. For future work, such aforementioned designs shall be addressed. Furthermore, low power modes as e.g. simulated power gating using reverse body bias shall be included in our tool.

1 Partitions

V_{dd}	Leakage [mW]		
	Static	**Progr.**	**Dyn.**
$0.6V$	75.402	34.231	34.231
$0.8V$	146.983	73.260	73.260
$1.0V$	263.103	131.469	131.469
$1.2V$	475.283	244.51	244.51

2 Partitions

Leakage [mW]		
Static	**Progr.**	**Dyn.**
16.200	10.824	10.824
33.679	25.170	25.170
61.581	49.468	49.478
127.037	102.662	102.662

3 Partitions

V_{dd}	Leakage [mW]		
	Static	**Progr.**	**Dyn.**
$0.6V$	14.944	10.0503	9.639
$0.8V$	29.818	22.6656	21.049
$1.0V$	55.019	44.7449	42.412
$1.2V$	108.917	95.033	89.162

TABLE I. Leakage evaluation for optimized 1, 2 and 3 BBI partitionings, 8-Tap FIR filter

Acknowledgment

This work is supported by VDEC Tokyo with Cadence Design Systems and Synopsys, Japan. The authors also express their gratitude to STARC, CMP and STMicro for their cooperation. Furthermore, this work was partially funded by the State of Baden-Württemberg, Germany, Ministry of Science, Research and Arts within the scope of Cooperative Research Training Group EAES and the Things2DO project BMBF 16ES0247.

References

[1] D. Jacquet, F. Hasbani, P. Flatresse, R. Wilson, F. Arnaud, G. Cesana, T. Di Gilio, C. Lecocq, T. Roy, A. Chhabra *et al.*, "A 3 ghz dual core processor arm cortex (tm)-a9 in 28 nm utbb fd-soi cmos with ultra-wide voltage range and energy efficiency optimization," *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, vol. 49, no. 4, pp. 812–826, 2014.

[2] Y. Akgul, D. Puschini, S. Lesecq, E. Beigné, I. Miro-Panades, P. Benoit, and L. Torres, "Power management through dvfs and dynamic body biasing in fd-soi circuits," in *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*. ACM, 2014, pp. 1–6.

[3] S. Garg, D. Marculescu, R. Marculescu, and U. Ogras, "Technology-driven limits on dvfs controllability of multiple voltage-frequency island designs: a system-level perspective," in *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*. IEEE, 2009, pp. 818–821.

[4] A. Dobriyal, R. Gonnabattula, P. Dasgupta, and C. R. Mandal, "Workload driven power domain partitioning," in *Proceedings of the 16th International Conference on Progress in VLSI Design and Test*, ser. VDAT'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 147–155.

[5] T. Yamamoto, K. Hironaka, Y. Hayakawa, M. Kimura, H. Amano, and K. Usami, "Dynamic vdd switching technique and mapping optimization in dynamically reconfigurable processor for efficient energy reduction," in *Reconfigurable Computing: Architectures, Tools and Applications*. Springer, 2011, pp. 230–241.

[6] H. A. Honlian Su, Yu Fujita, "Body bias control for a coarse grained reconfigurable accelerator implemented with silicon on thin box technology," in *Proceedings of the 24th International Conference on Field Programmable Logic and Applications*. IEEE, 2014.

[7] T. Katagiri and H. Amano, "A high speed design and implementation of dynamically reconfigurable processor using 28nm soi technology," in *Proceedings of the 24th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2014.