BODY BIAS GRAIN SIZE EXPLORATION FOR A COARSE GRAINED RECONFIGURABLE ACCELERATOR

Yusuke Matsushita, Hayate Okuhara, Koichiro Masuyama, Yu Fujita, Ryuta Kawano, Hideharu Amano

Dept. of ICS, Keio University, Yokohama Japan
email: leap@am.ics.keio.ac.jp

ABSTRACT

This paper explores the grain of domain size of an energy efficient coarse grained reconfigurable array called CMA (Cool Mega Array). By using Genetic Algorithm based body bias assignment method, the leakage reduction of various grain size was evaluated. As a result, a domain with 2x1 PEs achieved about 40% power reduction with a 6% area overhead.

1. INTRODUCTION

Recent IoTs (Internet of Things) and wearable computing require accelerators which achieve a certain performance with extremely small power budget. Coarse Grained Reconfigurable Arrays (CGRAs), which use a large array of processing elements (PEs), can reduce the power consumption while keeping the total performance. However, the large PE array in CGRAs often requires an enormous amount of leakage power which diminishes their benefits.

Silicon-on-Thin BOX (SOTB) CMOS technology, developed by LEAP [1], allows transistors operation with much lower power supply voltage than that for conventional bulk CMOS transistors by reducing the variation of the threshold level. By using body-biasing, the leakage current and operational delay can be widely controlled. A previously proposed CGRA, called Cool Mega Array (CMA) [2], was developed using SOTB technology, and it is called CC-SOTB (CMA Cube-SOTB). In CC-SOTB, the large PE array is consisting of combinational logic, and the data flow of the target application is mapped directly. The micro-controller manages the data reading and writing between the input/output of the PE array and data memory modules. CC-SOTB has independent body-bias supply for the PE array and micro-controller to make the balance between the performance and leakage power according to the arithmetic intensity of the target application. For a computation intensive application, zero-bias or forward-bias is given to the PE array to enhance the performance while reverse-bias is given to the micro-controller and data memory. If the target application bottlenecks the data transfer between the memory and PE array, zero- or forward-bias is given to the micro-controller and memory, while the PE array receives the reverse-bias to suppress the leakage power without degrading the performance [3].

Although the method to find the optimal body-bias voltage has been investigated [4], the same bias voltage is given to all PEs in the PE array in order to make the voltage management simple. If the body-bias control can be done for finer-grain (for example, a PE or a group of PEs), more leakage power can be reduced without degrading the performance. Although a few researches have been exerted to find the best domain size of body biasing[5][6], it has not been applied to CGRAs with SOTB. This paper investigates the impact of body-bias domain size on the leakage power and area overhead for CC-SOTB.

2. CMA WITH SOTB

A key concept of the CMA architecture is reducing any energy usage other than that required for computation. Another key concept of the CMA architecture is optimizing the energy of each target application by balancing the performance of the PE array and the micro-controller. For applications with a high degree of arithmetic intensity, the performance of the PE array is enhanced by using a power budget, while the power of the micro-controller is lowered. However, when the application requires a lot of data sets for a computation, the power budget is used for the micro-controller that manages the data transfer between the data memory and Launch/Gather registers. In the first prototype, CMA-1 [2] [7] independently changes the supply voltage of the PE array and the micro-controller. The problem of CMA-1 is the large leakage power consumed in the PE array.

Since it is critical in IoTs or wearable application, we adopted SOTB CMOS technology to suppress it. SOTB is classified as an FD-SOI technology where the transistors are formed on thin BOX (Buried Oxide) layer. The delay and leakage power consumption can be optimized by controlling the bias voltage to the body (back-gate). Here, we refer to the body-bias voltages of NMOS transistor and PMOS transistor as VBN and VBP, respectively. VBN for NMOS transistors is given to p-well. That is, if VBN=0, the transistor works with a normal threshold level. If reverse-bias ($VBN < 0$) is given, the threshold is raised; thus, the leakage current is reduced while the delay is stretched. On the

contrary, forward-bias ($VBN > 0$) lowers the threshold which enhances the operational speed with an increase of the leakage current. In the case of PMOS transistors, VBP is given to the n-well; thus, zero bias means $VBP = VDD$. When $VBP > VDD$, this corresponds to reverse-bias, while $VBP < VDD$ is for forward-bias.

Fig. 1 shows the block diagram of the CC-SOTB, a prototype CMA architecture using SOTB technology [3].

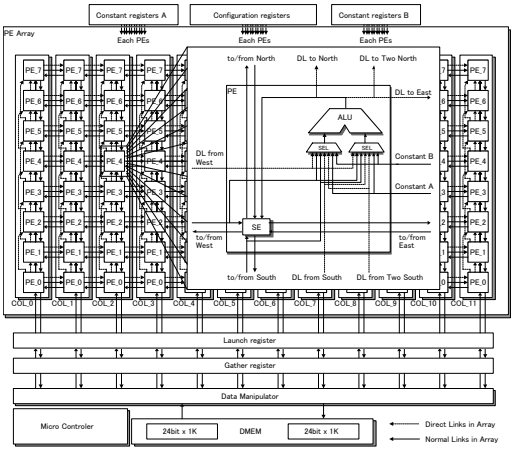

Fig. 1. Block diagram of CC-SOTB

A PE consists of a simple 24-bit ALU that executes multiply, add, subtract, shift, and logic operations, and a switching element (SE). It has a 12×8 PE array connected with a network using a two-channel island-style interconnection and direct links that connect to the north-east and east of the PE. The SEs transfer the input data from the PE in the south, west, and east of the PE and the output data of the ALU to the PE in the appropriate direction according to the configuration data. The micro-controller is a tiny microprocessor that executes a 14-bit micro-code stored in a 128-entry micro-memory. It reads eight data from the DMEM and sets the launch register with a single instruction. A dedicated memory controller triggered with the instruction executes the data transfer with eight clock cycles. Also, the data in the "Gather register" can be written back to the DMEM with a single instruction handled by another controller.

In CC-SOTB, unlike controlling independent power supply, independent body-bias is given to the PE array and micro-controller/data memory. For a target application with strong arithmetic intensity, the PE array is given a forward-bias while the micro-controller/data memory is given a reverse-bias. In contrast, if the data transfer has a bottleneck, the forward-bias is given to the micro-controller/data memory, and the reverse-bias is given to the PE array.

3. BODY BIAS DOMAIN DIVISION

The concept of domain division in the PE array is illustrated in Fig. 2 where each body-bias domain is enclosed in a red frame. White, gray, and black rectangles represent zero-bias, weak reverse-bias, and strong reverse-bias domains, respectively.

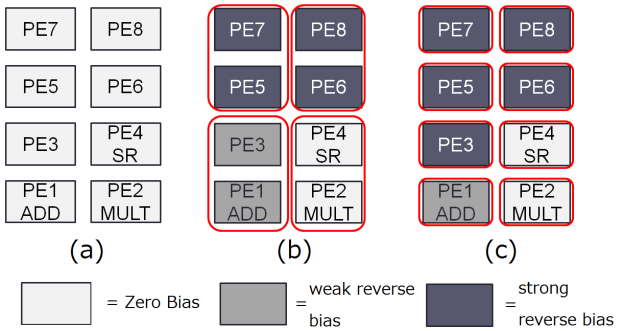

Fig. 2. Example of body-bias domain division (a) current CMA (b) division size:1x2 (c) division size:1x1

The current CMA uses a single body-bias for the whole PE array; thus, the body bias voltage for the critical path must be given to all PEs (Fig.2(a)) even if one or more PEs do not work at all. This incurs a waste of leakage power. When we divide the partition into groups each of which has 2x1 PEs, we can use a strong reverse-bias to the unused PEs, as depicted in Fig. 2(b). Furthermore, and as represented in Fig. 2(c), we can save more leakage power by giving a weak reverse-bias to PEs which are not on the critical path. Obviously, the smaller the group size of PEs is, the more the leakage reduction can be achieved without degrading performance, if the appropriate body-bias is assigned. However, the domain partitioning increases the chip area for two reasons. First, in order to apply different body-bias voltage, the substrate must be separated with a certain distance. Second, two body-bias lines for PMOS transistors (VBP) and NMOS transistors (VBN) must be delivered to each power domain. Such body-bias power distribution requires additional area overhead.

Here, first, we propose the body-bias assignment algorithm for each PE group. Then, the benefit of body-bias partitioning is evaluated based on simple application programs. Considering the area overhead based on the layout, we investigate the optimized size of the PE group.

3.1. Body Bias Assignment Algorithm

3.1.1. Delay and leakage power table

Body-bias voltage must be selected considering which type of calculation is done in each PE. Now, in CMA program design, a data-flow graph is extracted from the application program written in C-like language. Then it is mapped onto the PE array with Blackdiamond[8] tool which uses a simulated annealing for the mapping. Here, our body-bias assignment algorithm searches an optimized setting without changing the initial mapping obtained with Blackdiamond. For a large sized PE partitioning, the mapping which considers the PE group would achieve more efficient results. However, this is will be considered in our future work. Unlike algorithms for

dual-V_{th} or dual-V_{dd} FPGA design [9][10], the bias voltage can be changed widely and delicately in SOTB.

We previously [3] proposed a method to control the body-bias where accurate parameters of the used formulas are obtained from real chip measurements. However, in order to give an appropriate body-bias voltage considering the operation executed in each PE, we need a more precise delay estimation of each operation for each body-bias voltage. Therefore, we firstly made a table of delay and leakage power for each operation in a PE. Here, the balanced body-biasing, which gives the same bias voltage to PMOS and NMOS transistors, is used. That is, $VBN+VBP = VDD$. It means that the bias voltage can be represented only by VBN. Here, we evaluated the maximum delay and leakage power when varying VBN from -1.0V to 0.4V with a 0.2V interval for each instruction (ADD, SUB, MULT, PASS, NOUSE, AND, OR, SL, SR). The table's results are obtained from the simulation of the PE layout design using Synopsys's HSIM, a light-weight SPICE simulator.

3.1.2. Genetic algorithm for assignment

Since there is an enormous number of combinations for the body-bias assignment, we used Genetic Algorithm (GA) to find the optimal combinations. An element of the algorithm is a domain or a PE group which shares the body-bias with the assigned body-bias voltage. An individual is a vector formed by concatenation of all elements in order. Each element calculates the delay of paths which go through the PE group by referring to the delay table and configuration data for each application data flow. Here, GA is designed as follows: First, the fitness function of individual i is determined with the following expression. Here, a low fitness is more preferable.

$$f(i) = \begin{cases} -Li(iff \forall p \in PD_{i,p} < D_{critical}) \\ -\alpha \times max_{p \in P}(D_{i,p})(otherwise) \end{cases}$$

$D_{i,p}$ is the delay of a path p which goes through the PE group i. L_i is the total leakage power of PE groups represented with the individual i. By setting an enough large α, we can avoid the assignment which stretches the delay time of the path. Tournament selection with size 3 is used. That is, three individuals are selected randomly from 1400 individuals, and the one with the best fitness is selected and left for cross-over. Two-point cross-over is applied with a probability of 0.2. The mutation probability is set to 0.2, and each element is changed randomly with a probability of 0.3. 1400 generations are computed.

In the implementation, Python and GA library called Deap [11] are used.

While this algorithm has an advantage to seek the sub-optimal solution in a short time, it should be noted that it does not guarantee the accuracy of seeking the exact optimal solution. To examine the quality of the GA solution, a brute-force-search (BFS) was done for the limited exploration space of the simplest application called "alpha-blender".

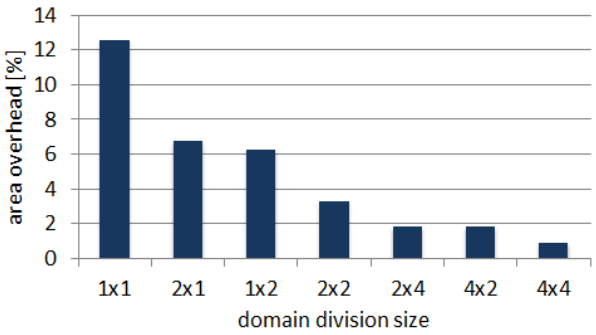

Fig. 3. Estimated area overhead caused by the domain separation

For this application, the results from GA and BFS were exactly the same. The computation time of GA is about five minutes while BFS requires S^P iterations where S is the number of V_{bb} sources and P is the number of PEs in the PE array. Therefore, BFS requires 8^{96} iterations for 12x8 PE array and eight V_{bb} sources which makes it not suitable for complex applications.

4. EVALUATION

4.1. Overhead of domain separation

First, overhead of domain separation is estimated. We assumed the same process technology: 65nm 7-metal Renesas SOTB used in CC-SOTB[12]. SOTB provides a triple-well structure in which each body-bias domain has independent N-well and P-well. In order to avoid interference between them, each domain must keep a space of 5.2μm vertically and 7.2μm horizontally. The restriction introduces an area overhead of the domain separation. Also, we must consider the space of wire straps to deliver the body bias. Fig.3 shows the estimated area overhead of each domain size. Here, the body bias domains are classified into three categories depending on their size ratios: 1) "1:1" ratio domains (e.g., 1x1 and 2x2 PE domain sizes), 2) "1:2" ratio domains (e.g., 1x2 and 2x4 PE domain sizes), and 3) 2:1 ratio domains (e.g., 2x1 and 4x2 PE domain sizes). We opted for this classification since domains with long height or width introduce other layout challenges that would like to analyze. The overhead of the finest domain (1x1 PE domain) is about 12.6%. It can be reduced to about 6% by merging two neighboring PEs together in a single domain. Note that the overhead of 1x2 domain is slightly smaller than that of 2x1 domain, since the vertical straps can be shared.

4.2. Leakage power reduction

The evaluation target uses the same semiconductor process and design tools shown in Table 1. Simple image processing programs shown in Table 2 are used for the evaluation. Table 2 also shows the PE utilization. $alpha$ and $sepia$ use 8bit input and so the arithmetic intensity is not so large, while af and $gray$ use relatively a large number of PEs.

The PE array used here is 12x8, the same design as that of CC-SOTB. The size of a PE group in the same domain

Table 1. Specification of CC-SOTB

Chip	Process Size I/O	LEAP 65nm SOTB 7-metal 5mm × 5mm 208pins
Tools	Design Synthesis P&R	Verilog HDL Synopsys Design Compiler 2011.09-SP2 Synopsys IC Compiler 2010.12-SP5

Table 2. Application used for the evaluation

Name	Function	PE utilization
alpha	8-bit alpha blender	24/96 (25%)
sepia	8-bit sepia filter	60/96 (62.5%)
af	24-bit RGB alpha blender	72/96 (75%)
gray	24-bit RGB gray scale filter	84/96 (87.5%)

is represented by $verticalnumber \times horizontalnumber$. Here, seven sizes: 1x1, 1x2, 2x1, 2x2, 2x4, 4x2 and 4x4 are evaluated.

Fig. 4 shows the power reduction ratio compared to the case when all PEs use the zero-bias. Note that the body bias voltages optimized by the proposed algorithm are assumed to be given to each power domain directly.

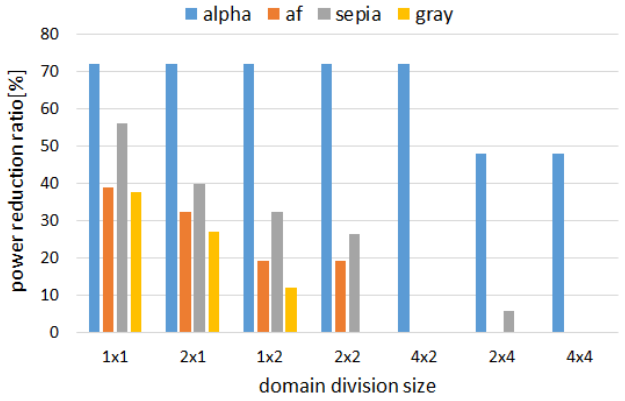

Fig. 4. Power reduction ratio of each division size

As expected, in $alpha$ that uses a small number of PEs, a high degree of leakage reduction can be achieved even with the size of 4x4. However, in other application programs, none or small leakage current is reduced with larger size than 2x4. 1x1 achieves the best leakage reduction that reaches 40% in average. If we do not care about the 12.6% area overhead, obviously it is the best solution.

2x1 achieves more reduction than that of 1x2. It comes from the fact that the data flow is assigned from the lower rows to upper rows in Blackdiamond. Thus, the horizontally longer domain can make the better use of reverse bias than the vertically longer ones. Unfortunately, the area overhead of 2x1 is slightly larger than that of 1x2; but, the difference is quite small. Considering the small area overhead (6%) and the large power reduction (35% on average), 2x1 is the best domain size in most cases.

5. CONCLUSION

The leakage power reduction and area overhead of the body bias domain separation applied to an energy efficient CGRA were analyzed. By using Genetic Algorithm based body bias assignment method, the leakage reduction of various grain sizes was evaluated. As a result, a domain with 2x1 PEs achieved about 40% power reduction with a 6% area overhead.

In the proposed algorithm, the application mapping onto the PE array did not consider the body bias domain. By using the application mapping algorithm considering body bias domain, the leakage power reduction in larger domains will be improved. Improvement of the mapping algorithm is our future work.

Acknowledgment

This work was done in "Ultra-Low Voltage Device Project" of LEAP funded and supported by METI and NEDO. This work is also supported by VDEC, the University of Tokyo in collaboration with Cadence Design Systems, Inc.

6. REFERENCES

[1] "Low Power Electronics Association & Project," http://www.leap.or.jp/.

[2] N. Ozaki et al., "Cool Mega Arrays: Ultra-low-Power Reconfigurable Accelerator Chips," *IEEE Micro, vol.31, No.6*, pp. 6–18, 2011.

[3] H. Su , H. Amano, "Real Chip Evaluation of a low power reconfigurable accelerator with SOTB Technology," *TECHNICAL REPORT OF IEICE, vol.133, No.325*, pp. 71–76, 2013.

[4] Yu Fujita et al., "Power optimization considering the chip temperature of low power reconfigurable accelerator CMA-SOTB," *Proc. of The Third International Symposium on Computing and Networking(CANDAR)*, pp. 21–29, 2015.

[5] M. Hioki, et al., "Fully-Functional FPGA Prototyple with Fine-Grain Programmable Body Biasing," *Proc. of 21st ACM/SIGDA International Symposium on FPGA* , pp. 73–80, Feb. 2013.

[6] J. M. Kuehn, et al., "Spatial and Temporal Granularity Limits of Body Biasing in UTBB-FPSOI," *Proc. of Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 876–879, Grenoble, France, May. 2015.

[7] N. Ozaki et al., "Cool Mega Array: A highly energy efficient reconfigurable accelerator," *FPT 2011*, pp. 1–8, 2011.

[8] V. Tunbunheng, et al., "Black-Diamond: a Retargetable Compiler using Graph with Configuration Bits for Dynamically Reconfigurable Architectures," in *Proc. of The 14th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI)*, 2007, pp. 412–419.

[9] Y. Hu, et al. , "Simultaneous time slacking budgeting and retiming for dual-Vdd FPGA power reduction," in *Proc. of 43rd ACM/IEEE Design Automation Conference*, 2006, pp. 478–483.

[10] C. Q. Tran, et al., "95% Leakage-Reduced FPGA using Zigzag Power-gating, Dual-Vth/VDD and Micro-VDD-Hopping," in *Proc. of Asian Solid-Stage Circuits Conference*, 2005, pp. 149–152.

[11] F. -A. Fortin, et al., " DEAP: Evolutionary Algorithms Made Easy ," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, 2012.

[12] K. Ishibashi, et al., "A Perpetuum Mobile 32bit CPU on 65nm SOTB CMOS Technology with Reverse-Body-Bias Assisted Sleep Mode," in *IEICE Transactions on Electronics*, July 2015, pp. 536–543.