A Method of Power Supply Voltage Assignment and Scheduling of Operations to Reduce Energy Consumption of Error Detectable Computations

Yuki Suda Kazuhito Ito

Graduate School of Science and Engineering, Saitama University
255 Shimookubo, Sakura-ku, Saitama, 338-8570, Japan
{suda,kazuhito}@elc.ees.saitama-u.ac.jp

Abstract— As the VLSI technology evolves, VLSI circuits are becoming more vulnerable to noises such as the crosstalk, the power supply fluctuation, and single event upsets (SEU). To detect an error caused by the SEU in functional units, operations are executed twice and the results are compared to check if those are identical or not. Such doubly executing operations and the comparison may require large energy consumption. In this paper a method of the power supply voltage assignment and the scheduling of operations is proposed to reduce the energy consumption of the error detectable circuits.

I. Introduction

As the feature sizes and the power supply voltages decrease and the operating frequency increases with the evolution of the VLSI technology, the noise margins of VLSI circuits have decreased considerably. Therefore, VLSI circuits are becoming more vulnerable to noises from many sources such as the crosstalk, the power supply fluctuation, and single event upsets (SEU) [1, 2, 3]. SEUs are caused by the radiation which deposits sufficient charge to alter the voltage, i.e., the logic level, of the node in the memories [1, 2, 3] and in the logic circuits [4, 5, 6]. These alteration of logic level is considered as the soft error.

There are several approaches to achieve the sufficient level of reliability of VLSI circuits against the SEU. The first is the fault-tolerant approach where the error is corrected at the system level. For example, with the triple modulo redundancy, the same operation is executed three times, the results are compared, and the majority is adopted as the correct result. The drawback of this approach is that many functional units or long operation time to execute the operation three times are necessary. The second is the circuit hardening where each elementary circuit is designed so that it is immune to SEUs [4, 5]. Although the hardening works effectively to the moderate charge deposited by an SEU, there can be an error when the charge more than the predetermined threshold is deposited. The third is the error detection where an error is detected by executing an operation twice and comparing the results [7]. If the results are not identical, it is known that an error has occurred. In this approach an error is detected but not corrected. The erroneous operation is executed again, or discarded and skipped to the next operation. Therefore the error detection approach is applicable to applications where the re-execution or the skip is allowed.

In this paper, the error detection approach is considered. In the CMOS technology, the execution of operations consumes energy. Therefore, the error detection approach requires much energy consumption because every operation is executed twice and the computed results need to be compared. It is well known that the energy consumption of CMOS circuits is remarkably reduced when the power supply voltage is lowered [8, 9, 10]. By executing some of the operations with the lower power supply voltage, it is expected that much energy consumption is saved. In this paper, a method is proposed to appropriately assign the power supply voltage to the operations so that the energy consumption is minimized with respect to the processing time and the functional unit constraints.

This paper is organized as follows. In Section 2, the problem is defined and a huristic algorithm to solve the problem is described. The experimental results are presented in Section 3. Section 4 concludes the work.

II. Proposed Method

A. Problem definition and a motivating example

The problem is defined as follows. For a given processing algorithm, any single error occurred in the computations during the execution of the processing algorithm is to be detected. The precondition is that at most one of the computations to implement the given processing algorithm produces an erroneous result. That computation is either one of the original computations of the given processing algorithm or the comparison computation to detect an error. The objective is to minimize the energy consumed in executing the processing algorithm with respect to the constraints that the maximum processing time and the maximum number of functional units (adders, multipliers, etc.) are specified.

The error detection is implemented as follows. The error in the computation is detected by comparing the results of doubly executed operations. That is, the same operation is executed twice and the execution results are compared. If those are identical, no error occurred. Otherwise, it is known that one of the two execution results contains an error. The error detection is illustrated in Fig. 1. Figure 1(a) shows a data flow graph (DFG)

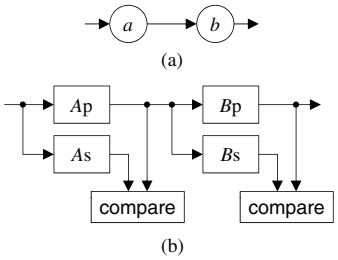

Fig. 1. The error detection method. (a) An example of operations. (b) The comparison of the doubly executed computations.

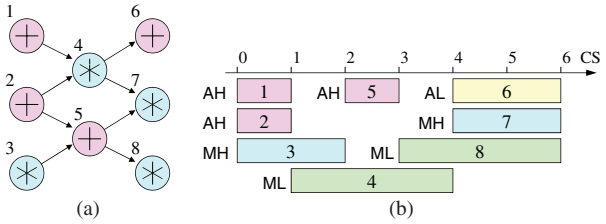

Fig. 2. The supply voltage assignment for computations. (a) An example DFG. (b) The voltage assignment and the scheduling result.

of an example processing algorithm. A node represents an operation and an edge represents a data dependency between the operations. In Fig. 1(a), the operation b uses the result of the operation a. The data dependency imposes a precedence relation between operations. The edge (a,b) implies that the computation of b must be executed after the computation of a is completed. The primary and the secondary computations for each operation in the processing algorithm are executed. In Fig. 1(b), a box represents a computation of the operation executed in a functional unit or a comparison executed in a comparator unit. 'Ap' is the primary computation of the operation a and 'As' is the secondary computation of the same operation a. If the results of these computations differ, an error is detected which occurred either in Ap or in As. It is important to note that there can be a case that both the primary and the secondary computations had no error, but the comparison had an error and falsely reported that either Ap or As contains an error. In this case, the comparison result is treated as it is and the computation results of the operation a are considered as erroneous. That is, the error detection method described here works in the safer side. The case, where the results of Ap and As differ by an error but the comparator falsely reports that the results are identical by another error, is not considered. This is because the situation involves two errors and contradicts to the precondition of a single error during a processing.

The primary computation 'Bp' and the secondary computation 'Bs' of the operation b are executed using the same data output by Ap and the results of Bp and Bs are compared to check if there is an error in any of Bp, Bs, and the comparison of Bp and Bs.

In the CMOS technology, the energy consumption by the processing of the circuit is proportional to the square of the power supply voltage V_{DD} and the processing latency is approximately inverse proportional to V_{DD} [8]. If multiple supply voltages are available, the energy consumption can be minimized by appropriately assigning a supply voltage to the operations. That is, the operations on the critical timing path are executed with a high V_{DD} (V_{DDH}) so that the constraint on the processing time is satisfied. On the other hand the operations with a time margin can be executed with a lower V_{DD} (V_{DDL}) to reduce the energy consumption.

Figure 2(a) shows an example DFG. The nodes 1, 2, 5, and 6 are additions and the nodes 3, 4, 7, and 8 are multiplications. It is required that all the computations are completed within 6 control steps (CS). The lower supply voltage V_{DDL} is assume to be $(2/3)V_{DDH}$. Therefore, the computation durations are 1 CS for an addition with V_{DDH}, 2 CSs for an addition with V_{DDL}, 2 CSs for a multiplication with V_{DDH}, and 3 CSs for a multiplication with V_{DDL}. The supply voltage assignment and the scheduling result are shown in Fig. 2(b). The additions 1, 2, and 5 are assigned V_{DDH}, which is denoted as AH in the figure. The addition 6 is assigned V_{DDL} (AL), the multiplications 3 and 7 are assigned V_{DDH} (MH), and the multiplications 4 and 8 are assigned V_{DDL} (ML). While the operations 1, 2, 3, 5, and 7 are executed with the higher supply voltage to satisfy the processing time constraint, the remaining operations are assigned the lower supply voltage to reduce the energy consumption. The required functional units are two AHs, one AL, one MH, and one ML. The multiplier is pipelined and the operations 4 and 8 are executed in the same multiplier.

The error detection of computations may require large energy consumption because every operation in the processing algorithm needs to be executed twice for the primary and the secondary computations. In addition, the comparisons of the results of the primary and the secondary computations also consume the energy. As described above, the result of the primary computation (e.g. Ap) is used as the input data to the computations of the succeeding operation (e.g. Bp and Bs). Therefore it is better to assign V_{DDH} to the primary computations so that the succeeding operations can be started as early as possible. Meanwhile, the secondary computation can be assigned V_{DDL} to reduce the energy consumption. This is because the result of the secondary computation is used only by the comparison to detect an error and hence the execution of other operations is not affected.

If the comparison to detect an error can be delayed a little, the execution of the secondary computation may have more margin time. Then more secondary computations can be assigned V_{DDL} to further reduce the energy consumption. However, delaying the error detection does not always reduce the energy consumption. When an error is detected in a computation of the processing, that processing would be terminated, any intermediate results would be discarded, and the processing is re-executed or skipped to the next processing. Therefore, to minimize the energy consumption, it is important to detect an error as early as possible so that the number of computations which use the erroneous data and the results are eventually dis-

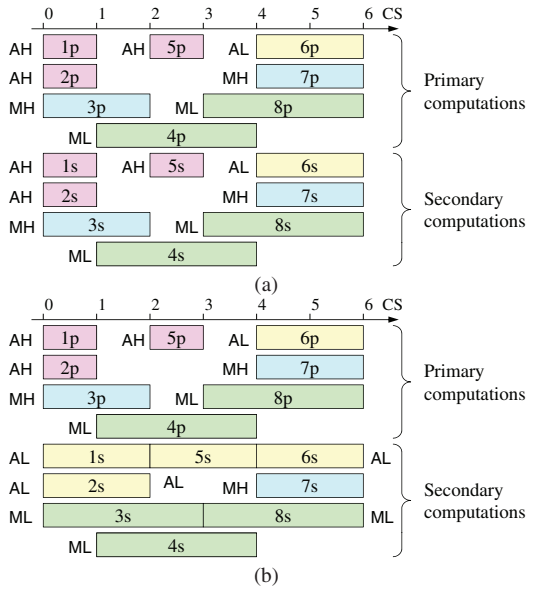

Fig. 3. The operation computation schedule for the error detection. (a) The computations are simply executed twice. (b) The modified voltage assignment.

carded becomes as small as possible. Here a parameter d_E is introduced which specifies the allowable delay of the error detection.

Using the schedule shown in Fig. 2(b) as the schedule of the primary computations, the secondary computations are added to achieve the error detection. The initial schedule of the secondary computations is just the same as the schedule of the primary computations. Hence twice the number of functional units are allocated. Figure 3(a) shows the initial schedule of the primary and the secondary computations for the DFG of Fig. 2(a). The required numbers of functional units are 4 for AH, 2 for AL, two for MH, and two for ML. These numbers are used as the constraint of the numbers of functional units. By setting the parameter $d_E = 1$, that is, the comparison can be delayed at most one CS, some of the secondary computations are assigned V_{DDL} to reduce the energy consumption in such a way that the comparison is not delayed more than one CS and the required number of functional units does not exceed the constraint. In this example, all the secondary computations except for 7s are assigned V_{DDL} as show in Fig. 3(b) to minimize the energy consumption. Although 4 adders with V_{DDH} (AH) are available, only two AHs are used after the modification of the supply voltage assignment. Those unused functional units are simply removed.

B. *A heuristic algorithm for voltage assignment and scheduling*

The algorithm to assign the supply voltage to operations and determine the execution schedule of the computations is described as follows. The inputs to the algorithm are (1) a DFG of the processing algorithm, (2) a constraint on the processing time, (3) the maximum number of functional units, (4) the execution duration of each functional unit, (5) and the parameter d_E. First the voltage assignment and the scheduling of the primary computations are determined so that the energy consumption of the primary computations is minimized. Then the voltage assignment and the scheduling of the secondary computations are determined to minimized the total energy consumption with respect to the constraints of the processing time, the number of functional units, and the parameter d_E.

(Voltage assignment and scheduling of the primary computations)

1. The operations in the DFG are scheduled with the list scheduling method by assuming that all the computations are assigned the higher supply voltage V_{DDH}. If a feasible schedule where all the computations are executed by the constrained processing time with respect to the given maximum number of functional units is not obtained, report that the combination of the constraints on the processing time and the numbers of functional units is too tight and exit.
2. The priority is given to all the computations so that the computation executed in the later CS is given the higher priority. All the computations are marked as 'unprocessed'.
3. An unprocessed computation with the highest priority is picked up. Let X denote the computation. If X can be assigned the lower supply voltage V_{DDL} so that the schedule of other computations are not affected and the maximum number of the functional units of the lower supply voltage does not exceed the constraint, then V_{DDL} is assigned to X. In addition, the execution CS of X is delayed as long as any precedence relation is violated. In other words, X is rescheduled as late as possible. Then X is marked as 'processed'.
4. While there is an unprocessed computation, repeat the step 3.

(Modifying voltage assignment and schedule of the secondary computations)

5. The voltage assignment and the schedule of the computations obtained above are considered as those of the primary computations. Duplicate the voltage assignment and the schedule of the primary computations to those of the secondary computations.
6. The secondary computations to which V_{DDH} is assigned are marked as 'candidate'.
7. A secondary computation marked as 'candidate' is picked up arbitrary. Let Ys denote the computation. The lower supply voltage V_{DDL} is assigned to Ys if the end CS of Ys is increased no more than d_E and the required number of the functional units for V_{DDL} does not exceed the maximum. Then Ys is marked as 'processed'.
8. While there is a computation marked as 'candidate', repeat the step 7.

TABLE I
CHARACTERISTICS OF FUNCTIONAL UNITS

FU	type	V_{DD}	CS	Energy
AH	Adder	V_{DDH}	1	4.653
AL	Adder	V_{DDL}	2	2.068
SH	Subtractor	V_{DDH}	1	5.129
SL	Subtractor	V_{DDL}	2	2.280
MH	Multiplier	V_{DDH}	2	48.14
ML	Multiplier	V_{DDL}	3	21.40
LS_LH	Level shifter	$V_{DDL} \rightarrow V_{DDH}$	—	0.5638

III. EXPERIMENTAL RESULTS

The proposed method was implemented using C++ programming language and was run on a 2.2GHz PC with 2GB memory.

Assuming a 0.18μm CMOS process as the target, the characteristics of the 16-bit functional units are summarized in Table I. Shown in Table I are from left to right, the name of the functional unit, the computation type, the power supply voltage (the higher voltage V_{DDH} or the lower voltage V_{DDL}), the computation duration in control step (CS), and the average energy consumption in pJ. In the CMOS technology, when transferring a signal generated in the circuit with V_{DDL} to the circuit with V_{DDH}, the level (voltage) of the signal must be converted between these two circuits. This is because the higher logic level in V_{DDL} is not recognized as the higher logic level by the circuit with V_{DDH}. A level shifter [8] can be constructed as a CMOS circuit and placed between the circuits with different supply voltages. The level shifter consumes energy when converting the level of signals. Usually any level shifter is not necessary to transfer a signal from V_{DDH} to V_{DDL}. The characteristic of the level shifter is also shown in Table I.

The example processing algorithms used in the experiments are the 5th order wave elliptic filter (WEF) [11] which consists of 8 multiplications and 26 additions, and the 8-point DCT [12] which consists of 11 multiplications and 29 additions. The constraints for the voltage assignment and the scheduling are summarized in Table II. Shown in Table II are from left to right, the identification of the constraint, the DFG to apply the constraint, the processing time constraint in CS (M), the maximum numbers of functional units of AH, MH, and ML for the primary computations, and the maximum number of functional units of AH, MH, and ML for the total of the primary and the secondary computations. For example, the constraint W11 is applied to the processing algorithm WEF, where all the computations must be done in 18 CSs, the primary computations are executed using at most 3 AHs, one MH, and one ML, and both the primary and the secondary computations are executed using at most 6 AHs, two MHs, and two MLs. In the case of constraints W10, W20, D10, and D20, functional units with V_{DDL} are not used.

The results of the proposed method for the cases $d_E = 0$ and $d_E = 1$ are shown in Table III. Table III shows the constraint imposed in the voltage assignment and the scheduling, the number of the primary multiplication computations executed with V_{DDL} (MLp), the number of total multiplication computations, either the primary or the secondary, executed with V_{DDL} (MLt), and the energy consumption (E). The energy consumption E includes the energy consumed by the comparisons to detect an error, which are implemented by subtractors with V_{DDL}. It can be seen from Table III that as the number of multipliers with V_{DDL} is increased, more multiplication computation is assigned V_{DDL} and more energy consumption is reduced. Up to 39% of the energy can be saved by the proposed method. The CPU time of the proposed method is less than 1 second for each run.

TABLE II
SCHEDULING CONSTRAINTS

Const.	DFG	M	Primary			Total		
			AH	MH	ML	AH	MH	ML
W10	WEF	18	3	1	0	6	2	0
W11	WEF	18	3	1	1	6	2	2
W12	WEF	18	3	1	2	6	2	4
W13	WEF	18	3	1	3	6	2	6
W14	WEF	18	3	1	4	6	2	8
W20	WEF	17	4	2	0	8	4	0
W21	WEF	17	4	2	1	8	4	2
W22	WEF	17	4	2	2	8	4	4
W23	WEF	17	4	2	3	8	4	6
D10	DCT	13	3	1	0	6	2	0
D11	DCT	13	3	1	1	6	2	2
D12	DCT	13	3	1	2	6	2	4
D13	DCT	13	3	1	3	6	2	6
D14	DCT	13	3	1	4	6	2	8
D20	DCT	9	4	2	0	8	4	0
D21	DCT	9	4	2	1	8	4	2
D22	DCT	9	4	2	2	8	4	4
D23	DCT	9	4	2	3	8	4	6
D24	DCT	9	4	2	4	8	4	8

TABLE III
VOLTAGE ASSIGNMENT AND SCHEDULING RESULTS

Const.	$d_E = 0$			$d_E = 1$		
	MLp	MLt	E	MLp	MLt	E
W10	0	0	1152.8 (100%)	0	0	1152.8 (100%)
W11	2	4	1047.0 (90.8%)	2	6	993.5 (86.2%)
W12	3	6	994.1 (86.2%)	3	9	913.9 (79.3%)
W13	4	8	941.2 (81.6%)	4	11	860.9 (74.7%)
W14	4	8	941.2 (81.6%)	4	12	834.2 (72.4%)
W20	0	0	1152.8 (100%)	0	0	1152.8 (100%)
W21	1	2	1099.9 (95.4%)	1	6	992.9 (86.1%)
W22	2	4	1047.0 (90.8%)	2	8	940.0 (81.5%)
W23	2	4	1047.0 (90.8%)	2	10	886.5 (76.9%)
D10	0	0	1494.4 (100%)	0	0	1494.4 (100%)
D11	4	8	1282.7 (85.8%)	4	8	1282.7 (85.8%)
D12	8	16	1071.1 (71.7%)	8	16	1071.1 (71.7%)
D13	10	20	965.2 (64.6%)	10	20	965.2 (64.6%)
D14	11	22	912.3 (61.0%)	11	22	912.3 (61.0%)
D20	0	0	1494.4 (100%)	0	0	1494.4 (100%)
D21	2	4	1388.6 (92.9%)	2	4	1388.6 (92.9%)
D22	4	8	1282.7 (85.8%)	4	8	1282.7 (85.8%)
D23	6	12	1176.9 (78.8%)	6	12	1176.9 (78.8%)
D24	7	14	1124.0 (75.2%)	7	14	1124.0 (75.2%)

TABLE IV
COMPARISON BETWEEN THE HEURISTIC ALGORITHM AND THE ILP MODE FOR WEF WITH CONSTRAINT W13, $d_E = 1$

	HA	ILP
MLp	4	4
MLt	11	12
E	860.9 (74.7%)	834.2 (72.4%)
CPU	$<$ 1 sec.	$\simeq$ 30 min.

In the case of WEF, more secondary computations are assigned V_{DDL} when $d_E = 1$ and hence the energy consumption is more reduced than the case when $d_E = 0$. On the other hand, in the case of DCT, the voltage assignment and the scheduling results are the same for $d_E = 0$ and $d_E = 1$. This is because as many secondary computations as possible are assigned V_{DDL} when $d_E = 0$ and no additional computation is assigned V_{DDL} when d_E is increased to 1.

To confirm the effectiveness of the proposed method, the problem was formulated as an integer linear programming (ILP) model in the similar way as proposed in [13]. Then the ILP is solved by the ILP solver lp_slove [14]. The results by the proposed heuristic algorithm (HA) and the ILP model were compared as shown in Table IV. The ILP solver found the optimal solution for WEF with the constraint W13 and $d_E = 1$. The energy consumption of the result obtained by the proposed HA is about 3.2% larger than the optimal. The comparison result shows that the proposed method can find a reasonably good solution within a much shorter CPU time than the ILP solver.

IV. CONCLUSIONS

In this paper, a method of the power supply voltage assignment and the scheduling of operations is proposed to reduce the energy consumption of the error detectable circuits. The experimental results show the effectiveness of the proposed method. Consideration for the register allocation and the voltage assignment of registers, the static energy minimization for both the functional units and the registers, are important tasks. In addition, the cases where more than two power supply voltages are available and/or the power supply voltages can be dynamically changed, need to be considered and it remains as a future work.

ACKNOWLEDGEMENT

This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Cadence Design Systems, Inc. and Synopsys, Inc.

REFERENCES

[1] P. Hazucha, C. Svensson, "Impact of CMOS Technology Scaling on the Atomospheric Neutron Soft Error Rate," IEEE Trans. Nucl. Sci., vol. 47, pp. 2586–2594, 2000.

[2] P. Hazucha, C. Svensson, "Cosmic Ray Neutron Multiple-Upset Measurements in a 0.6-μm CMOS Process," IEEE Trans. Nucl. Sci., vol. 47, pp. 2995–2602, 2000.

[3] J A. Maestro, P. Reviriego, "Study of the Effects of MBUs on the Reliability of a 150 nm SRAM Device," Proc. DAC 2008, pp. 930–935, 2008.

[4] M. P. Base, S. P. Buchner, D. McMorrow, "A Digital CMOS Design Technique for SEU Hardening," IEEE Trans. Nucl. Sci., vol. 47, pp. 2603–2608, 2000.

[5] R. Garg, N. Jayakumar, "A Design Approach for Radiation-hard Digital Electronics," Proc. DAC 2006, pp. 773–778, 2006.

[6] R. Garg, C. Nagpal, S P. Khatri, "A Fast, Analytical Estimator for the SEU-induced Pluse Width in Combinational Designs," Proc. DAC 2008, pp. 918–923, 2008.

[7] S. Matsuzaka, K. Inoue, "A Dependable Processor Architecture with Data-Path Partitioning," IPSJ Tech. Report, vol. 2004-SLDM-117, pp. 7–11, 2004.

[8] T. Kuroda, T. Sakurai, "Overview of Low-Power ULSI Circuit Techniques," IEICE Trans. Electron., Vol. E78-C, No. 4, pp. 334–344, 1995.

[9] Y. Zhang, X. S. Hu, D. Z. Chen, "Task Scheduling and Voltage Selection for Energy Minimization," Proc. DAC 2002, pp. 183-188, 2002.

[10] D. Chen J. Cong, Y. Fan, J. Xu, "Optimality Study of Resource Binding with Multi-Vdds," Proc. DAC 2006, pp. 580–585, 2006.

[11] S. M. Heemstra de Groot, S. H. Gerez, and O. E. Herrmann, "Range-Chart-Guided Iterative Data-Flow Graph Scheduling," IEEE Trans. Circuits Syst.-I: Fund. Theory & Appl., vol. 39, pp. 351–364, 1992.

[12] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical Fast 1-D DCT Algorithms with 11 Multiplications," Proc. IEEE ICASSP '89, pp. 988–991, 1989.

[13] K. Ito, L. E. Lucke, K. K. Parhi, "Module Selection and Data Format Conversion for Cost-Optimal DSP Synthesis," Proc. ICCAD 1994, pp. 322–329, 1994.

[14] lp_solve, http://lpsolve.sourceforge.net/5.5/.