Evaluation of a Self-Adaptive Voltage Control Scheme for Low-Power FPGAs

Shota Ishihara, Zhengfan Xia, Masanori Hariyama, and Michitaka Kameyama

Abstract—**This paper presents a fine-grain supply-voltage-control scheme for low-power FPGAs. The proposed supply-voltage-control scheme detects the critical path in real time with small overheads by exploiting features of asynchronous architectures. In an FPGA based on the proposed supply-voltage-control scheme, logic blocks on the sub-critical path are autonomously switched to a lower supply voltage to reduce the power consumption without system performance degradation. Moreover, in order to reduce the overheads of level shifters used at the power domain interface, a look-up-table without level shifters is employed. Because of the small overheads of the proposed supply-voltage-control scheme and the power domain interface, the granularity size of the power domain in the proposed FPGA is as fine as a single four-input logic block. The proposed FPGA is fabricated using the e-Shuttle 65 nm CMOS process. Correct operation of the proposed FPGA on the test chip is confirmed.**

Index Terms—**Reconfigurable VLSI, multiple supply voltages, dynamic voltage and frequency scaling, asynchronous architecture**

I. Introduction

Field-Programmable Gate Arrays (FPGAs) are widely used to implement special-purpose processors. FPGAs are cost-effective for small-lot production because functions and in terconnections of logic resources can be directly programmed by end users. In spite of their design cost advantage, FPGAs impose a large power overhead compared to custom silicon alternatives. The overhead limits integrations of FPGAs into portable devices.

Since the power consumption is proportional to the square of the supply voltage, lowering the supply voltage reduces the power consumption significantly. However, lowering the supply voltage also increases the delay of the block. Blocks on the critical path are supplied with the highest voltage since its performance determines the system performance. The rest blocks can be supplied with a lower voltage without system performance degradation [1]. Thus, a large proportion of blocks can be supplied with the lower voltage. This approach saves the power consumption significantly.

Multi-voltage techniques can be mainly classified as follows:

- Static Voltage Scaling (SVS): different blocks are given different, fixed supply voltages.
- Dynamic Voltage and Frequency Scaling (DVFS): blocks are dynamically switched between two or more voltage levels follow changing workloads.

DVFS is a more efficient technique for power reduction since the workload in a processor varies with time [2-4]. Fig. 1 shows a DVFS architecture. Each multi-voltage domain has a DVFS controller, a voltage selector and a programmable clock generator. The multi-voltage domains communicate via the asynchronous handshake protocol.

Although DVFS is efficient for power reduction, it is difficult to be implemented in conventional synchronous FPGAs. First, in DVFS, pairs of voltage and frequency values for each power domain need to be determined with sufficient margin to guarantee operation across the

Manuscript received Jun. 28, 2010; revised Aug. 25, 2010.
Intelligent Integrated Systems Laboratory, Graduate School of Information Science, TOHOKU University, Japan
E-mail : hariyama@ecei.tohoku.ac.jp

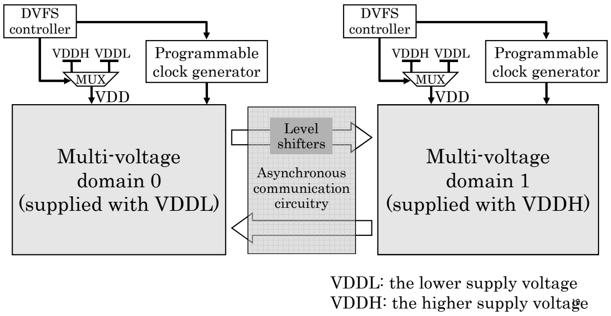

DVFS controller
VDDH VDDL
MUX
VDD
Programmable clock generator
Multi-voltage domain 0 (supplied with VDDL)
Level shifters
Asynchronous communication circuitry
DVFS controller
VDDH VDDL
MUX
VDD
Programmable clock generator
Multi-voltage domain 1 (supplied with VDDH)
VDDL: the lower supply voltage
VDDH: the higher supply voltage

Fig. 1. DVFS architecture.

entire range of the best and worst case process. Energy savings are possible only the system level performance requirements are understood. It is clear when the frequency can be lowered without missing deadlines. Second, each multi-voltage domain requires a local clock tree, however it is difficult to be implemented. In FPGAs, the local clock tree can be implemented using the programmable interconnection resources. However, the worst case of clock skew cannot be estimated since FPGA vendors do not guarantee the worst case of the minimum delay of components. As a result, it is impossible to guarantee that no hold-time violations occur [5]. For this reason, the Xilinx ISE reference manual [6] strongly recommends not customizing the fixed clock tree.

Moreover, the conventional synchronous FPGAs are difficult to implement fine-grain DVFS which archives lower power than coarse-grain DVFS. The power consumption of DVFS circuitry is mainly consumed by the DVFS controller, the programmable clock generator, the level shifters and the voltage-control signal distribution network. The fundamental challenge for any DVFS technique is to ensure that the saved power outweighs the power overhead of the DVFS circuitry. DVFS techniques are classified into two types: coarse-grain DVFS and fine-grain DVFS. In coarse-grain DVFS, a large number of LUTs (Look-Up-Tables) share a single DVFS controller so the area and power overheads of the DVFS controller are relatively small. However, if any LUT within a coarse-grain multi-voltage domain is supplied with a high voltage to achieve a high throughput, none of the LUTs which share the same voltage control signal can be supplied with a lower voltage even if they are not required to achieve a high throughput. FPGAs with coarse-grain DVFS also cause a large dynamic power and a large area overheads in the voltage-control signal distribution network since it is distributed to many LUTs through programmable interconnection resources. On the other hand, in fine-grain DVFS, each LUT has its own DVFS controller. Therefore, when any LUTs are not required to achieve a high throughput, they can be supplied with a lower voltage. This results in much lower power consumption compared to coarse-grain DVFS. Especially, for FPGAs, no programmable interconnection resource for distributing the voltage-control signal is required. In fine-grain DVFS, each LUT has its DVFS controller, the number of the DVFS controllers is much larger than that of coarse-grain DVFS. This results in large area and power overheads. Due to these overheads, fine-grain DVFS is commonly assumed to be less efficient than coarse-grain DVFS, although it has the potential to save more power.

In order to implement fine-grain DVFS to FPGAs for lower power, this paper proposes a supply-voltage-control scheme called self-adaptive voltage control, which detects the critical path in real time with small overheads by exploiting features of asynchronous architectures. In the proposed supply-voltage-control scheme, logic blocks on the sub-critical path are autonomously switched to a lower supply voltage to reduce the power consumption without system performance degradation. The supply voltage of each power domain is self-adaptive to the workload, data path, process variation and temperature. Note that no software or complex off-line analyze is required. Moreover, in order to reduce the overheads of level shifters used at the power domain interface, a Look-Up-Table (LUT) without level shifters is employed [7]. Because of the small overheads of the proposed supply-voltage-control scheme and the power domain interface, the granularity size of the power domain in the proposed FPGA is as fine as a single four-input logic block.

This paper is an extension of conference paper [8] with detail implementation and new evaluations.

II. Related Work

1. Autonomous Fine-Grain Power Gating

A hardware-based fine-grain power gating technique, called autonomous fine-grain power gating has been proposed [9, 10]. The power gating technique directly detects the activity of each power-gated domain by exploiting the features of asynchronous architectures, and uses this activity to determine when to shut down and wake up the power-gated domain. The activity of a power-gated domain can be easily detected by comparing the phases of the input data with that of the output data. Since the activity of each power-gated domain can be detected easily, the area and the power overheads of the sleep controller are small. Thanks to the small power overhead of the sleep controller, the energy penalty for entering the sleep-state from the active-state and exiting the active-state from the sleep-state is small. As a result, the granularity size of the power-gated domain is small, and the energy breakeven time is short.

2. Asynchronous FPGAs

In order to solve the problems caused by the clock and the clock tree, asynchronous FPGAs have been proposed [11-16]. Asynchronous encoding schemes are mainly classified into

- Single-rail encoding (ex. bundled-data encoding)
- Dual-rail encoding (ex. four-phase dual-rail encoding)

The bundled-data encoding is the most common one in the single-rail encoding. Fig. 2 shows a simple bundled-data pipeline. In the bundled-data encoding, request and value are splits into separate wires. The value is encoded as in a synchronous circuit using N wires to denote a N-bit number, and request is encoded using a dedicated request wire denoted by *Req*. The bundled-data encoding requires the explicit insertion of matching delays in *Req* to ensure that a request is never received before the bundled value is valid. The bundled-data encoding is the most frequently-used way in ASICs since its hardware overhead is relatively small. This is because the *Req* wire is shared among all the *N* wires. Hence, to transfer an *N*-bit value, only *N+2* wires are required. The major disadvantage of the bundled-data encoding is that it requires the constraint of the delay length. If the data path is fixed in advance, it is relatively easy to meet the constraint by optimizing the layouts. However, for FPGAs, since the data path is programmable, complex programmable delay elements are required. As a result, the bundled-data encoding is not suitable for FPGAs.

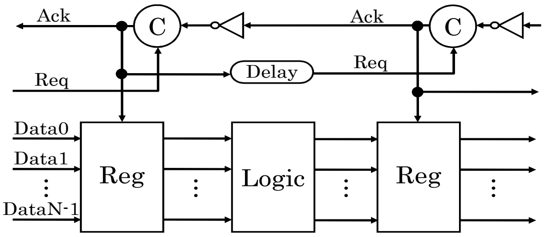


Fig. 2. Simple bundled-data pipeline.

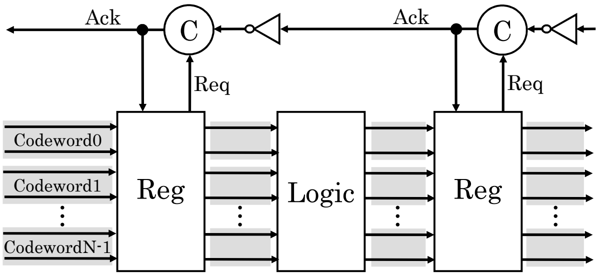


Fig. 3. Simple dual-rail pipeline.

The dual-rail encoding encodes a bit onto two wires. Fig. 3 shows a simple dual-rail pipeline. In the dual-rail encoding, value is made implicit in the request and no delay insertion is therefore required [17]. Hence, the dual-rail encoding is the ideal one for FPGAs. In the dual-rail encoding, to transfer an *N*-bit value, *2N+1* wires are required. The four-phase dual-rail encoding is the most common one in dual-rail encodings. Table 1 shows the code table of the four-phase dual-rail encoding. The four-phase dual-rail encoding has two phases: “data” and “spacer”. The data value “0” is encoded as (0, 1) and “1” is encoded as (1, 0). Moreover, the spacer is encoded as (0, 0). Fig. 4 shows the example where data values “0”, “0” and “1” are transferred. The main feature is that the sender sends a spacer after a data value. The receiver knows the arrival of a data value by detecting the change of either bit: “0” to “1”. As a result, the data itself has the information of data arrival, and no delay element is

Table 1. Code table of the four-phase dual-rail encoding

	Codeword (t, f)
Data value 0	(0, 1)
Data value 1	(1, 0)
Spacer	(0, 0)

* Codeword (1, 1) is not used

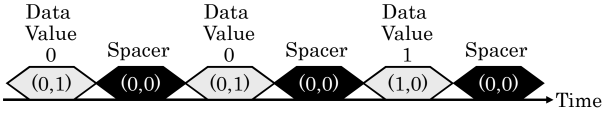


Fig. 4. Example of the four-phase dual-rail encoding.

required. Hence, the four-phase dual-rail is suitable for FPGAs.

III. Architecture

1. Overview

Fig. 5 shows the overall structure of the proposed FPGA and Fig. 6 shows the programmable interconnection resources. Similar to conventional synchronous FPGAs, the proposed FPGA consists of Logic Blocks (LBs), Connection Blocks (CBs) and Switch Blocks (SBs). The LB computes an arbitrary four-input function. The LB accesses nearby communication resources through the CB, which connects LB input and output terminals to routing resources through programmable switches. The SB consists of diamond switches that allow a signal on a track to connect to another track. Because dual-rail encodings are employed, four wires are required for a data bit. Two wires for the data encoded in dual-rail encoding, one wire for the acknowledge and one wire for the supply-voltage-control signal. The four wires for a data bit are controlled by a single configuration memory bit.

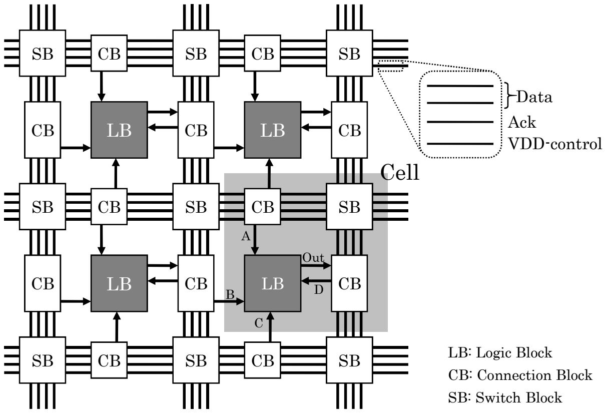


Fig. 5. Overall structure.

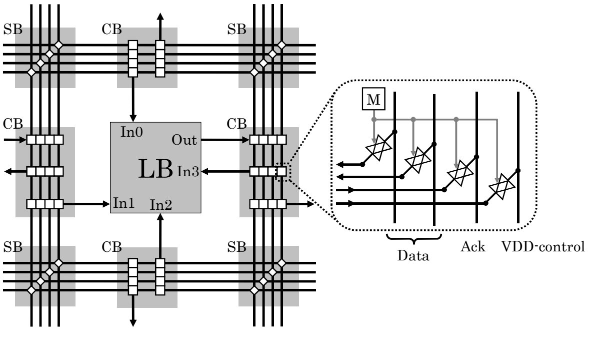


Fig. 6. Programmable interconnection resources around an LB.

2. Fundamental Principle of Self-Adaptive Voltage Control

In an asynchronous architecture, it is easily detected which input data arrives earlier at an LB. Using the data arrival information, the critical path can be detected in real time. Fig. 7 demonstrates the principle of the data arrival detection. For simplicity, each LB is assumed to operate a two-input function. As the initial state, the phases of the input and output data are spacer. If *In0* arrives earlier than *In1*, the phase of *In0* changes to data, and the phase of *In1* does not change (Case0 shown in Fig. 7). On the other hand, if *In1* arrives earlier than *In0*, the phase of *In1* changes to data, and that of *In0* does not change (Case1 shown in Fig. 7). Therefore, the data arrival comparator just extracts and compares the phases of the input data. As a result, the area and power overheads of the data arrival comparator are small.

Fig. 8 shows the critical path detection using the data arrival information. For each LB, the solid line from the previous LB denotes the data that arrives earlier; the dotted line denotes the data that arrives later; the dot-dash line denotes the data arrives at the same time as the other. Beginning with the last LB, the critical path is

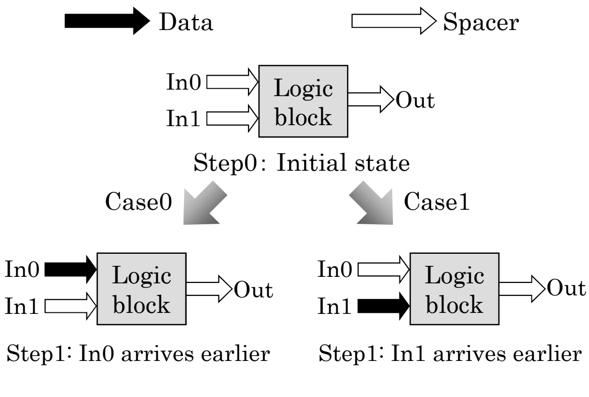


Fig. 7. Data arrival detection using an asynchronous architecture.

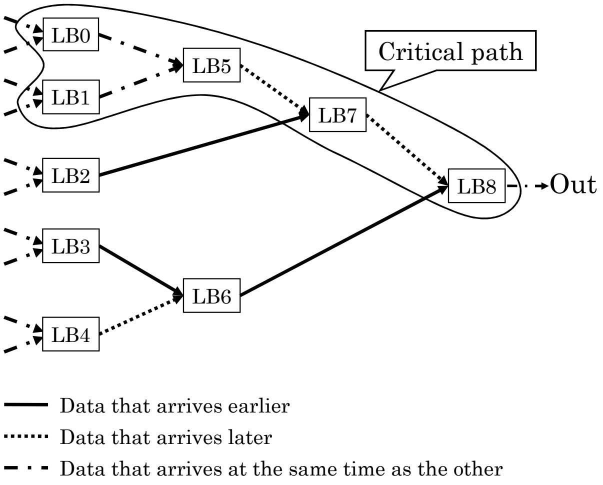
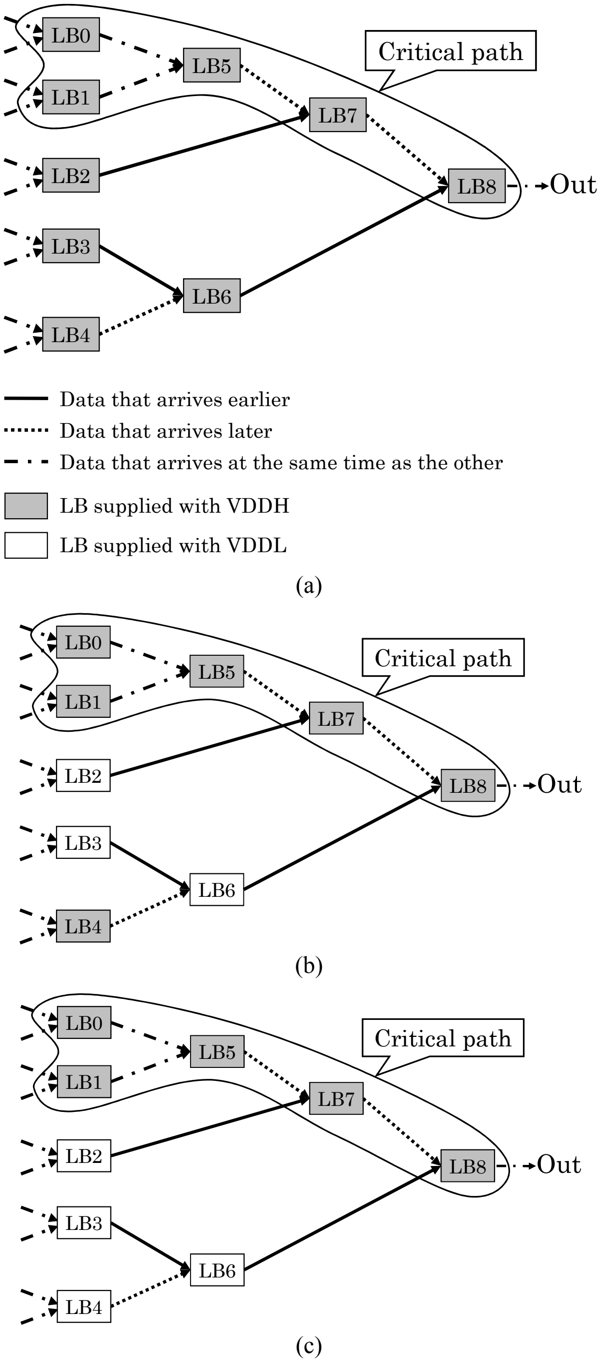

LB0
LB1
LB2
LB3
LB4
LB5
LB6
LB7
LB8
Out
Critical path
Data that arrives earlier
Data that arrives later
Data that arrives at the same time as the other

Fig. 8. Critical path detection using the data arrival information.

gradually found by repeating the comparison of data arrival as follows. First, the last LB (LB8) is focused since it is mathematically guaranteed that the last LB is on the critical path. The predecessors of LB8 are LB7 and LB6. Data from LB7 arrives at LB8 later than that from LB6. This means that LB7 is on the critical path. Next, LB7 is focused. Data from LB5 arrives at LB7 later than that from LB2. This means that LB5 is on the critical path. Finally, LB5 is focused. Data from LB0 and LB1 arrive at LB5 at the same time. This means that LB0 and LB1 are on the critical path. As a result, the path: LB0, LB1, LB5, LB7, LB8 is the critical path. Thus, by exploiting the data arrival information, the critical path is detected in real time. Since the critical path is detected, LBs on the sub-critical path can be autonomously supplied with a lower voltage to reduce the power consumption without system performance degradation.

Fig. 9 shows an example of the proposed supply-voltage-control scheme. In the example, two supply voltages are used. The gray blocks are the LBs supplied with the higher voltage (VDDH), while the white blocks are the LBs supplied with the lower voltage (VDDL). As shown in Fig. 9(a), as the initial state (Step0), all LBs are supplied with VDDH. As shown in Fig. 9(b), in Step1, LBs whose data arrives earlier at the successor are switched to VDDL. Although many LBs are supplied with VDDL, some LBs which are not on the critical path are supplied with VDDH. As shown in Fig. 9(c), in Step2, LBs which send data to the LB supplied with VDDL are switched to VDDL. All LBs on the sub-critical path are supplied with VDDL by repeating Step2. In this scheme, only the LBs on the critical path are supplied with VDDH, and the other are supplied with VDDL. As a result, the power consumption is reduced without system performance degradation.



(a)

(b)

(c)

Fig. 9. Example of the proposed supply-voltage-control scheme (a) Step0 (b) Step1 (c) Step2.

Fig. 10 shows the function of the supply-voltage-controller (VDD-controller) in an LB. In the proposed supply-voltage-control scheme, each LB is dynamically

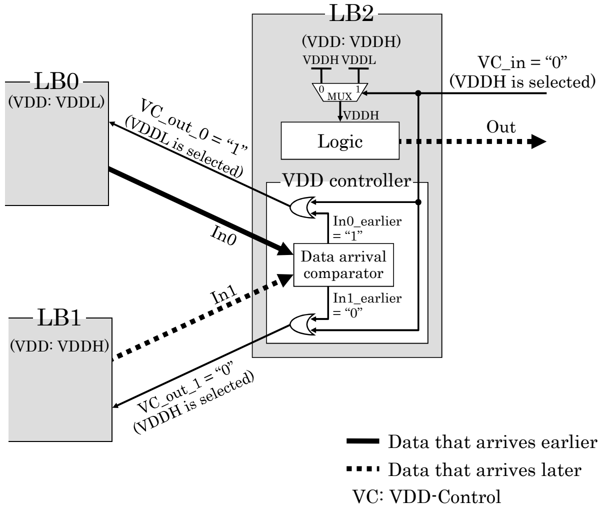


Fig. 10. Function of the VDD-controller.

switched to VDDH or VDDL according to the VDD-control signal from the next LB. For simplicity, two-input LBs are shown. CBs and SBs are also omitted. In the example, LB2 is assumed to be on the critical path and to be supplied with VDDH. The supply voltages of LB0 and LB1 are controlled by LB2. The data from LB0 (*In0*) arrives at LB2 earlier than that from LB1 (*In1*) for a predefined threshold time. The threshold time is determined such that *In0* arrives earlier than *In1* even if the supply voltage of LB0 is VDDL. LB2 detects the data arrival information of each input data, and sends the VDD-control signal (*VC_out_0*) to LB0 for setting its supply voltage to VDDL. Since *In0* arrives earlier than *In1* even if the supply voltage of LB0 is VDDL, the power consumption is reduced by lowering the supply voltage without system performance degradation. Note that if LB2 is not on the critical path and is supplied with VDDL, LB0 and LB1 are also not on the critical path. In this case, LB0 and LB1 are controlled to be supplied with VDDL. In this way, in the proposed supply-voltage-control scheme, the VDD-control signal is propagated together with the data and the acknowledge signal as shown in Fig. 6. Therefore, no extra configuration memory bits are required to route the VDD-control, and the extra routing resources are only pass-gates and wires.

3. Circuit Implementation

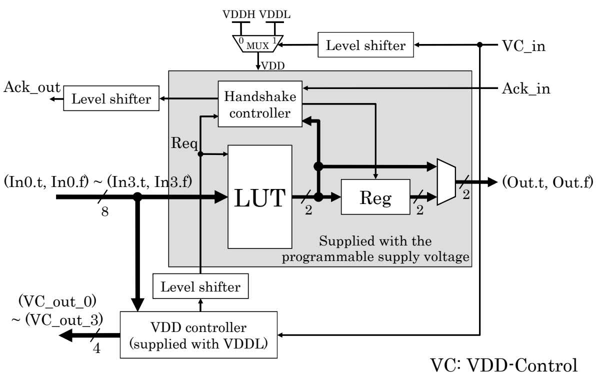


Fig. 11. Logic block structure.

Fig. 11 shows the structure of an LB. The LB mainly consists of an LUT, an output register, a handshake controller, a programmable power supply, a VDD controller and three level shifters. The VDD controller is supplied with VDDL, and the gray region is supplied with the programmable power supply. The inputs of the VDD controller are the input data from the previous LBs of the data path. If the VDD controller is supplied with the programmable power supply, the level shifters are required for each input signals. In the proposed FPGA, the VDD controller is supplied with VDDL since the delay overhead is smaller than that of the level shifters. As mentioned in the later, since the LUT is based on dynamic circuit, no level shifter is required for the inputs. Hence, in the LB, only three level shifters are required for control signals. As a result, the power and area overheads of the power domain interface are small.

Fig. 12 shows the structure of the VDD controller. For simplicity, instead of a four-input VDD controller which is used in the actual LB, a two-input VDD controller is shown. The VDD controller mainly consists of a data arrival detector, a programmable delay, pass-gates and latches. The data arrival detector is also used to generate the request signal (*Req*) for the LUT and the handshake controller. Since the *Req* generating circuit is also required for conventional asynchronous FPGAs, the

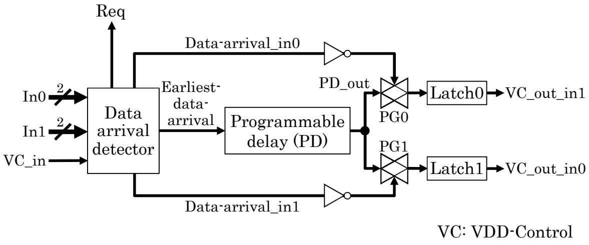


Fig. 12. VDD controller.

overheads of the VDD controller are small. The behavior of the VDD controller can be classified into two cases. Fig. 13 shows the case where one data arrives earlier than the other data for a predefined threshold time *Tth* (Case0), and Fig. 14 shows the case where the two data arrive at almost the same time (Case1).

In Case0, as shown in Fig. 13, *In0* arrives earlier than *In1*. When *In0* arrives, *Data-arrival_in0* is set from “0” to “1”. When *In1* arrives, *Data-arrival_in1* is set from “0” to “1”. Since *In0* arrives earlier than *In1*, the waveform of *Earliest-data-arrival* is same as that of *Data-arrival in0*. *Earliest-data-arrival* propagates through the programmable delay, and the delayed signal is *PD out*. The delay time of the programmable delay is used as the predefined threshold time *Tth*, and is determined such that the earlier arriving data (*In0*) arrives earlier than the other data (*In1*) even if the supply voltage of the LB which sends the earlier arriving data (*In0*) is VDDL. When *In0* arrives, pass-gate *PG0* is opened. Since *PG0* is opened before *PD out* arrives at *Latch0*, *VC_out_in1* is kept as “0”. Then the supplied voltage of the LB which sends *In1* is kept as VDDH. When *In1* arrives, *PG1* is opened. Since *PG1* is opened after *PD out* arrives at *Latch1*, *VC_out_in0* is set to “1”. Then the supplied voltage of the LB which sends *In0* is switched to VDDL.

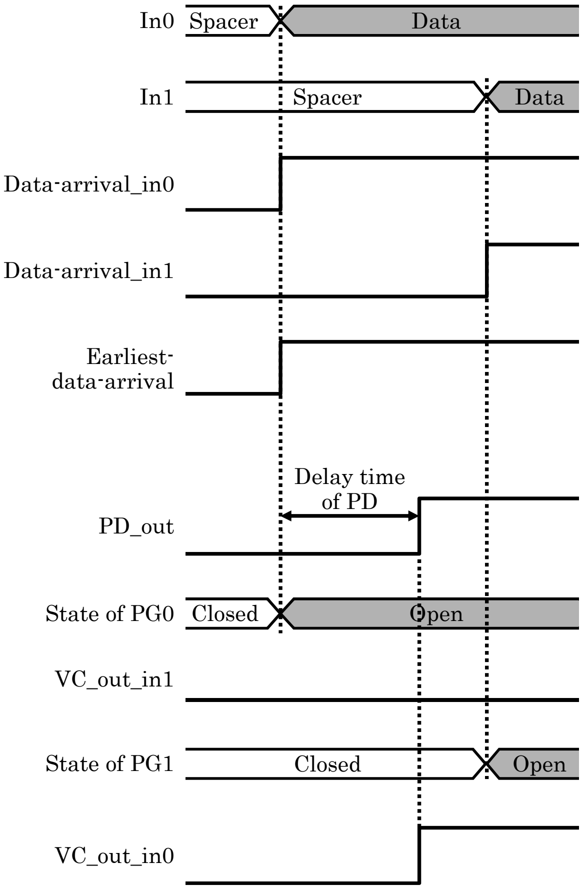


Fig. 13. Behavior of the VDD controller (Case0: *In0* arrives earlier than *In1* for the predefined threshold time *Tth*).

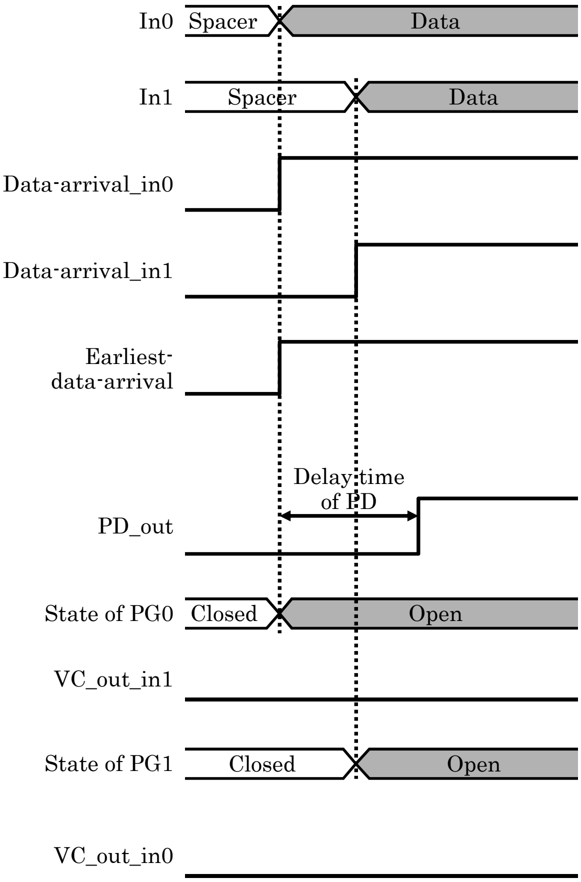


Fig. 14. Behavior of the VDD controller (Case1: *In0* and *In1* arrive at almost the same time).

In Case1, as shown in Fig. 14, *In0* and *In1* arrive at almost the same time. Since *PG0* and *PG1* are opened before *PD out* arrives at the latches. Then the supplied voltage of the LBs which send *In0* and *In1* are kept as VDDH.

Fig. 15 shows the structure of the LUT. For simplicity, instead of a four-input LUT which is used in the actual LB, a two-input LUT is shown. As a typical manner, to implement LB-level fine-grain DVFS for an FPGA, each input of LBs requires a level-shifter. Moreover, since the dual-rail encoding is used, the number of level shifters in the asynchronous FPGA is twice as many as that in a synchronous FPGA. The large number of level shifters

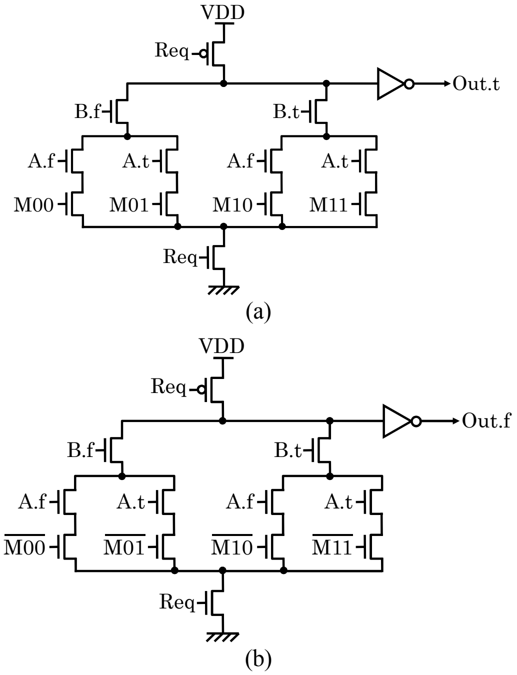


Fig. 15. LUT structure (a) Out.t generating circuit (b) Out.f generating circuit.

causes a large power and a large area overheads. In order to resolve this problem, dynamic circuit is used to reduce the number of level-shifters. The request signal (*Req*) is used as the pre-charge signal. That is the circuits pre-charge when *Req=0*, evaluate when *Req=1*. Since pull-down network is used, regardless the voltage of the supply voltage used in previous LBs and the current LB, no level-shifter is required for the LUT [7]. The behavior of the LUT is as follows. When *Req* =0, the values of *Out.t* and *Out.f* are "0", that is the output represents a spacer in the four-phase dual-rail encoding. When *Req* =1, the values of *Out.t* and *Out.f* are depend on the operation. Note that the value of *Out.f* is the same as $\overline{Out.t}$, that is the output represents a data in the four-phase dual-rail encoding.

IV. Evaluation

An FPGA based on the proposed supply-voltage scheme is fabricated using the e-Shuttle 65 nm CMOS process. Fig. 16 and Table 2 show the micro-photograph and the features of the proposed FPGA, respectively. The chip includes 100 cells, where a cell consists of an LB, a CB and an SB as shown in Fig. 17. Correct operation of the proposed FPGA on the test chip is confirmed.

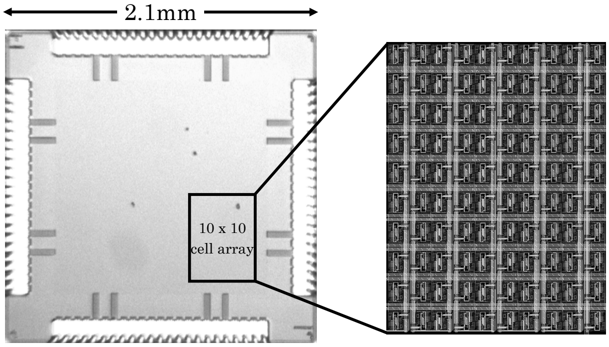


Fig. 16. Chip micro-photograph of the proposed FPGA.

Table 2. Features of the proposed FPGA

Process	e-Shuttle 65 nm CMOS
Standard supply voltage	1.2 V
Chip size	2.1 mm × 2.1 mm
Number of cells	10 × 10

Table 3 shows the comparison between a conventional asynchronous FPGA using a single supply voltage and the proposed FPGA. The evaluation circuit is a cell as shown in Fig. 17. For example, VDDH and VDDL are respectively set as 1.2 V and 0.9 V in the proposed FPGA. Compared to the conventional asynchronous FPGA, the transistor count is increased by 29%. This hardware overhead is mainly caused by the VDD-controller in the LB and the programmable switches for the VDD-control signal in the CB and the SB. Since the proposed FPGA supplied with VDDL uses a lower supply voltage than the conventional asynchronous FPGA, the processing time is increased by 92% and the processing energy is reduced by 33%. The processing energy of the proposed FPGA supplied with VDDH is

Table 3. Comparison between cells of an asynchronous FPGA using a single supply voltage and the proposed FPGA

	Single supply voltage	Multiple supply voltages (Proposed)
Supply voltage	1.2V	VDDH: 1.2V VDDL: 0.9V
Transistor count	836	1082
Processing-time / data-set	336ps	VDDH: 455ps VDDL: 646ps
Processing-energy / data-set	248fJ	VDDH: 208fJ VDDL: 166fJ
VDD-control-energy / data-set	-	30fJ

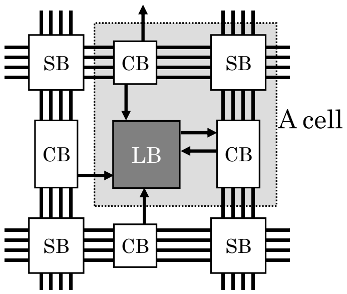


Fig. 17. Cell structure.

lower than that of the conventional asynchronous FPGA. This is because the data arrival detector is always supplied with VDDL. For the same reason, the processing time of the proposed FPGA supplied with VDDH is larger than that of the conventional asynchronous FPGA.

Fig. 18 shows the relationship between the energy consumption and VDDL, and Fig. 19 shows the relationship between the processing time and VDDL. The evaluation circuit is a cell. The energy consumption in Fig. 18 consists of the processing energy and the VDD-control energy. The processing time and energy consumption of the cell supplied with VDDH depends not only on VDDH but also on VDDL. This is because the data arrival detector is always supplied with VDDL.

Fig. 20 shows the relationship between the energy consumption and the ratio of the LBs supplied with VDDL to the total LBs. The evaluation circuit includes 100 cells. In an MPEG4 video codec application, 68% of the LBs can be supplied with VDDL [18]. If 0.7 V is chosen for VDDL, the energy is reduced by 35% compared to the conventional asynchronous FPGA.

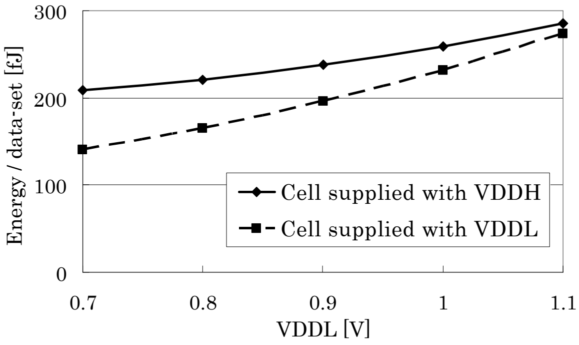


Fig. 18. Relationship between the energy consumption and VDDL.

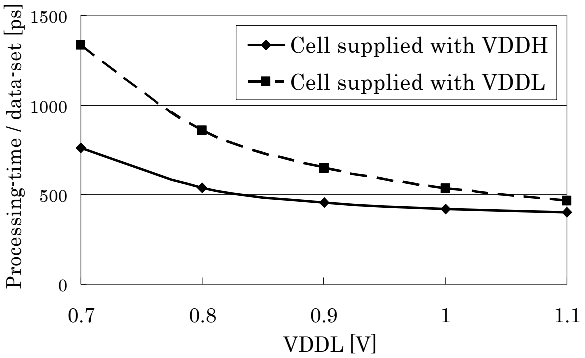


Fig. 19. Relationship between the processing time and VDDL.

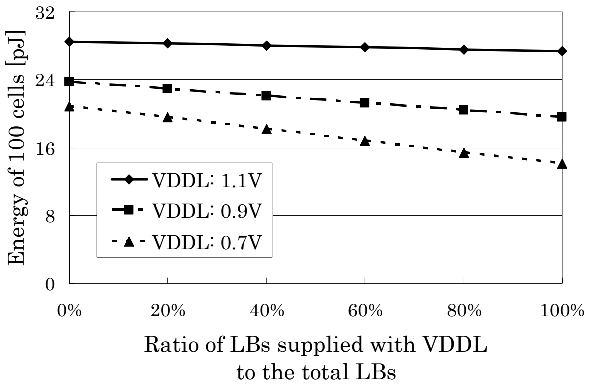


Fig. 20. Relationship between the energy consumption and the ratio of the LBs supplied with VDDL to the total LBs (100 cells).

V. CONCLUSIONS

This paper proposed a fine-grain DVFS scheme called self-adaptive voltage control for low-power FPGAs. In order to decide the supply voltage for each LB, an asynchronous architecture is exploited. LBs on the sub-critical path are switched to a lower supply voltage to reduce the power consumption. In the proposed FPGA, the supply voltage of each LB is self-adaptive to the workload, data path, process variation and temperature. Moreover, in order to reduce the overheads of level shifters used at the power domain interface, an LUT without level shifters is employed. Because of the small overheads of the proposed supply-voltage-control scheme and the power domain interface, the granularity size of the power domain in the proposed FPGA is as fine as a single four-input logic block.

By fully exploiting the adaptivity of asynchronous architectures, control parameters such as threshold voltages and a degree of parallelism will be self-adaptive to the workload, data path, process variation and temperature. Hence, the self-adaptive scheme is also suitable for dynamically reconfigurable processors since their data paths change dynamically and frequently, it is more difficult than FPGAs to determine the control parameters for each LB using offline analysis.

ACKNOWLEDGMENTS

This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with STARC, e-Shuttle, Inc., Fujitsu Ltd., Cadence Design Systems Inc., Synopsys Inc. and Mentor Graphics, Inc.

REFERENCES

[1] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual: For System-on-Chip Design*. Springer, 2007.

[2] G. Magklis, G. Semeraro, D. H. Albonesi, S. G. Dropsho, S. Dwarkadas, and M. L. Scott, "Dynamic Frequency and Voltage Scaling for a Multiple-Clock-Domain Microprocessor," *IEEE Micor*, Vol.23, No.6, pp.62-68, 2003.

[3] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott, "Energy-Efficient Processor Design Using Multiple Clock Domains with Dynamic Voltage and Frequency Scaling," in *Proc. International Symposium on High-Performance Computer Architecture*, 2002, pp.29-40.

[4] V. Kursun and E. G. Friedman, *Multi-voltage CMOS Circuit Design*. Wiley, 2006.

[5] *Gated clock conversion with Synplicity's synthesis products*, Synplicity Inc. Application Note, Jul., 2003.

[6] *Synthesis and Simulation Design Guide*, Xilinx Inc., 2008.

[7] W. Chong, M. Hariyama, and M. Kameyama, "Low-Power Field-Programmable VLSI Using Multiple Supply Voltages," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, Vol.88, No.12, pp.3298-3305, 2005.

[8] S. Ishihara, Z. Xia, M. Hariyama, and M. Kameyama, "Architecture of a Low-Power FPGA Based on Self-Adaptive Voltage Control," in *Proc. International SoC Design Conference (ISOCC)*, 2009, pp.274-277.

[9] M. Hariyama, S. Ishihara, , and M. Kameyama, "A Low-Power Field-Programmable VLSI Based on a Fine-Grained Power-Gating Scheme," in *Proceedings of the IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug., 2008, pp.430-433.

[10] S. Ishihara, M. Hariyama, and M. Kameyama, "A Low-Power FPGA Based on Autonomous Fine-Grain Power-Gating," in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan., 2009, pp.119-120.

[11] K. Maheswaran and V. Akella, "PGA-STC: programmable gate array for implementing self-timed circuits," *International Journal of Electronics*, Vol.84, No.3, pp.255-267, 1998.

[12] R. Payne, "Asynchronous FPGA Architectures," *IEE Computers and Digital Techniques*, Vol.143, No.5, 1996.

[13] J. Teifel and R. Manohar, "An asynchronous dataflow FPGA architecture," *IEEE Transactions on Computers*, Vol.53, No.11, pp.1376-1392, 2004.

[14] M. Hariyama, S. Ishihara, C. C. Wei, and M. Kameyama, "A Field-Programmable VLSI Based on an Asynchronous Bit-Serial Architecture," in *Proceedings of the IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Nov., 2007, pp.380-383.

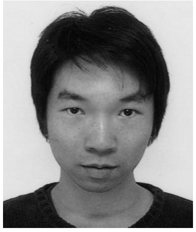
[15] M. Hariyama, S. Ishihara, and M. Kameyama, "Evaluation of a Field-Programmable VLSI Based on an Asynchronous Bit-Serial Architecture," *IEICE Trans. Electron*, Vol. E91-C, No.9, pp.1419 -1426, 2008.

[16] S. Ishihara, Y. Komatsu, M. Hariyama, and M. Kameyama, "An Asynchronous Field-Programmable VLSI Using LEDR/4-Phase-Dual-Rail Protocol Converters," in *Proceedings of International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, 2009, pp.145-150.

[17] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design: A Systems Perspective*. Kluwer

Academic Publishers, 2001.

[18] T. Kuroda and M. Hamada, "Low-Power CMOS Digital Design with Dual Embedded Adaptive Power Supplies," *IEEE Journal of Solid-State Circuits*, Vol.35, No.4, pp.652-655, 2000.



Shota Ishihara received the B.E. degree in Information Engineering and M.S. degree in Information Sciences from Tohoku University, Sendai, Japan, in 2007 and 2009, respectively. He is currently working toward the Ph.D. degree in Graduate School of Information Sciences, Tohoku University. His research interests include reconfigurable computing and asynchronous architecture.

Zhengfan Xia received the B.E. degree in Electronic and Information Engineering from China University of Geosciences, Beijing, China, in 2008. He is currently working toward the M.S. degree in Graduate School of Information Sciences, Tohoku University. His primary research interest is in the area of asynchronous architecture.

Masanori Hariyama received the B.E. degree in electronic engineering, the M.S. degree in information sciences, and the Ph.D. degree in information sciences from Tohoku University, Sendai, Japan, in 1992, 1994, and 1997, respectively. He is currently an Associate Professor with the Graduate School of Information Sciences, Tohoku University, Sendai, Japan. His research interests include VLSI computing for real-world application such as robots, high-level design methodology for VLSIs and reconfigurable computing.

Michitaka Kameyama received the B.E., M.E. and D.E. degrees in Electronic Engineering from Tohoku University, Sendai, Japan, in 1973, 1975, and 1978, respectively. He is currently Dean and Professor in the Graduate School of Information Sciences, Tohoku University. His general research interests are intelligent integrated systems for real-world applications and robotics, advanced VLSI architecture, and new-concept VLSI including multiple-valued VLSI computing. He received the Outstanding Paper Awards at the 1984, 1985, 1987 and 1989 IEEE International Symposiums on Multiple-Valued Logic, the Technically Excellent Award from the Society of Instrument and Control Engineers of Japan in 1986, the Outstanding Transactions Paper Award from the IEICE in 1989, the Technically Excellent Award from the Robotics Society of Japan in 1990, and the Special Award at the 9th LSI Design of the Year in 2002. Dr. Kameyama is an IEEE Fellow.