A Low-Power FPGA Based on Self-Adaptive Multi-Voltage Control

Zhengfan Xia, Masanori Hariyama, and Michitaka Kameyama
Graduate School of Information Sciences, Tohoku University
Aramaki aza Aoba 6-6-05, Aoba, Sendai, Miyagi, 980-8579, Japan
E-mail: {xiazhengfan@, hariyama@, kameyama@}ecei.tohoku.ac.jp

***Abstract*—This paper presents a low-power FPGA that the supply voltage of each logic block autonomously changes to suit their deadlines. Dual-rail coding is used in FPGA datapaths to make data transfer time sensible in each pipeline stage. The deadline of the logic block in each pipeline stage is evaluated by comparing the data transfer time and the pipeline cycle time. When a low supply voltage does not violate the deadline, the supply voltage of the logic block is autonomously switched to the low voltage. This self-adaptive voltage control scheme saves power consumption without deteriorating the circuit performance. Moreover, level converters are unnecessary in the proposed FPGA which has a simple and efficient architecture.**

***Keywords*—FPGA, Multiple voltage, Asynchronous circuit.**

I. Introduction

Field-programmable gate arrays (FPGAs) are cost-effective and flexible for low volume applications because the users can freely change the function of the programmable logic blocks and the connection of the programmable switch blocks. Despite the advantages, FPGAs have a large power overhead compared to the custom VLSIs. High power consumption requires expensive packaging and cooling solutions. It also prohibits the use of FPGAs in battery-powered applications.

Reducing the supply voltage is an effective technique for reducing power in VLSI circuits. However, it also has negative affects on the circuit performance. A well-known technique to reap the benefits of voltage scaling without the performance penalty is the use of multiply supply voltages. The timing critical blocks operate on the normal supply voltage and the non-critical blocks operate on a low supply voltage. While this technique has been successfully applied in low-power custom ICs [1], it is difficult to be applied in FPGAs for power reduction.

The difficulty of designing a multiple supply voltage FPGA is that the optimal voltage assignment changes from one design to another. Voltage programmability is necessary to tune the voltage assignment according to the application. However, it is difficult to realize such a fine-grained multi-voltage design that the supply voltage of each logic block is programmable. The fine-grained design would cause large implementation overhead. Almost all previous works chose to use coarse-grained architecture, such as cluster-based architecture [2]. Despite the overhead for implementing voltage programmability, determining the voltage assignments to each logic block is a challenge. Especially, level converters are needed when a low supply voltage logic block drives a high supple voltage logic block. The imposed delay and energy overheads by level converters should be carefully considered when performing the voltage assignments.

This paper presents a low-power FPGA that the supply voltage of each logic block autonomously changes to suit their deadlines. Dual-rail coding is used in FPGA datapaths to make data transfer time sensible in each pipeline stage. A self-adaptive voltage controller is designed to evaluate the deadline of a logic block by comparing the data transfer time and the pipeline cycle time. When a low supply voltage does not violate the deadline, the supply voltage of the logic block is autonomously switched to the low voltage. Since this process is hardware control, it saves the design effort of offline analysis for the voltage assignments. The self-adaptive voltage controller has small overhead which is applied in a fine-grained level (each logic block). A global enable signal controls the controllers. When the voltage assignments are done, the enable signal disables the controllers to save power. Moreover, the architecture of the logic block is carefully designed that level converters are unnecessary in the proposed FPGA. This does not only avoid the level converter overhead problem but also maintain the flexibility of placement and routing. The evaluation result shows that the proposed logic block has no extra power consumption in its normal working state.

II. Architecture

A. Overview

The proposed FPGA is built on asynchronous island-style FPGA architecture, with the configuration stored in SRAM cells [3]–[5]. Fig.1 shows the overall structure. The proposed FPGA consists of logic blocks (LBs), connection blocks (CBs) and switch blocks (SBs). Each LB performs an arbitrary 4-input 1-output function. LBs are connected with each other through CBs and SBs. The programmable switches in CBs and SBs control the routing direction of the input and output ports of LBs. Because of the asynchronous architecture and dual-rail encoding datapath, a routing line consists of three wires: two wires for the data and encoded acknowledge signal transfer, one wire for the requirement signal transfer.

A self-adaptive controller is embedded in each LB, which is globally controlled by an enable signal. When the pipeline circuits work at stable state, a high pulse enable signal is asserted for several pipeline cycle times. During the enable

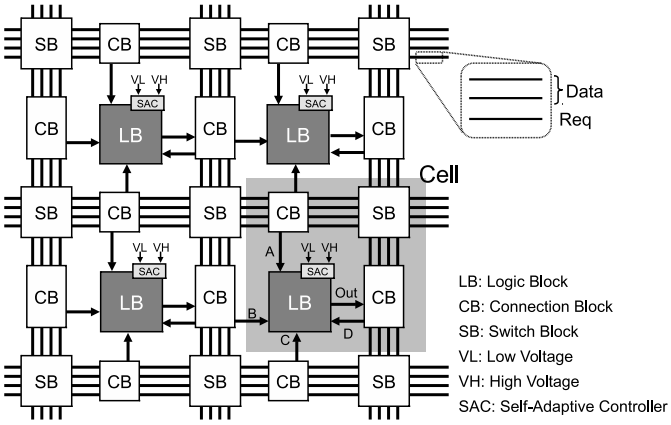

Fig. 1. Overall structure.

time, the controller evaluates the deadline of each LB and decides whether the low supply voltage satisfies the deadline. If the low supply voltage does not violate the deadline of a logic block, its supply voltage is autonomously switched to the low voltage. This self-adaptive voltage control scheme saves power consumption without deteriorating the circuit performance.

B. Asynchronous Architecture

In asynchronous design, there are mainly two encoding schemes: bundled-data encoding and delay-insensitive encoding [6]. Dual-rail encoding is one instance of delay-insensitive encoding that the handshake signal is encoded with data. It makes data arrival sensible, which is an essential feature used in the proposed self-adaptive voltage control scheme. Besides, delay-insensitive encoding is more suitable for dealing with reconfigurable datapath in FPGAs compared to bundled-data encoding.

1) Asynchronous pipelines: Fig.2 shows a simple bundled-data pipeline based on bundled-data encoding. Bundled-data encoding has an efficient handshake structure that a specific handshake line indicates the whole data transfer is done in the datapath. A delay element is analyzed and added at each pipeline stage to match the delay of the logic block. Bundled-data encoding is preferable in the design of custom ICs. The delay elements can be optimized accordingly. However, it is not suitable for FPGAs since the datapath is reconfigurable. Reconfigurable delay elements need to be designed and distributed, which causes large overhead and difficult to realize high performance [3].

On the other hand, delay-insensitive encoding encodes the handshake signal with data. A receiver knows the arrival of a data regardless of the delay on datapath. Therefore, matched delay elements are unnecessary in pipeline, which is suitable for the reconfigurable datapath in FPGAs [4]. Fig.2 shows a simple dual-rail pipeline based on dual-rail encoding.

2) 4-phase dual-rail protocol: Table I shows the code table of dual-rail encoding. Data 0 is encoded as $(0, 1)$ and data 1 is encoded as $(1, 0)$, the spacer is encoded as $(0, 0)$. Figure

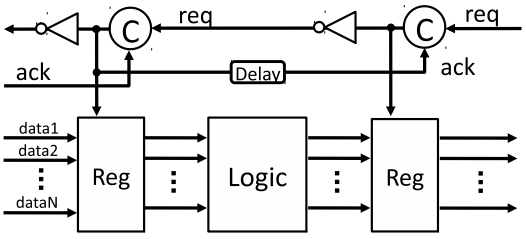

Fig. 2. A simple bundled-data pipeline.

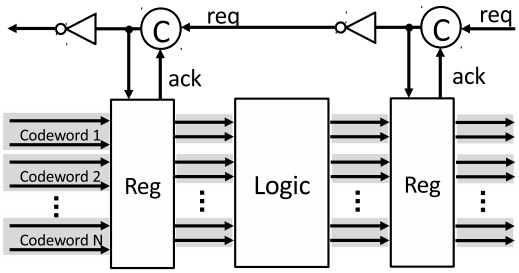

Fig. 3. A simple dual-rail pipeline.

4 shows an example of data transfer based on 4-phase dual-rail protocol. Each data is separated by a spacer. A receiver knows the arrival of a data or a spacer by detecting the signal changes. As a result, the receiver can get a valid data without considering the delay on datapath.

C. Self-Adaptive Multi-Voltage Control

1) Handshake delay model: Fig. 5 shows a handshake delay model, which consist of two pipeline stages. It is used to easily understand the handshake process in asynchronous pipeline. Req signal indicates stage1 requires a data (req=1) or a spacer (req=0). Ack signal indicates the data (ack=1) or the spacer (ack=0) is ready in stage0. C-element is an important handshake element in asynchronous design. When both inputs are 0 the output is set to 0, and when both inputs are 1 the output is set to 1. For other input combinations the output does not change.

In the handshake delay model, the out of C-element is considered to be a start point. When stage1 requires a spacer (req=0) and the spacer is ready in stage0 (ack=0), the output of C-element is set to 0. From this start point, stage1 starts to absorb the spacer from stage0 and set req signal to 1 (require a data) after logic delay and handshake delay, $t_{l1} + t_{h1}$. At the same time, stage0 starts to prepare a data. The data will be ready after handshake delay and logic delay, $t_{l0} + t_{h0}$. When req=1 and ack=1, the output of C-element is set to 1. Then, stage1 starts to absorb the data from stage0, and stage0 starts to prepare next spacer. The handshake enters next cycle.

2) Multi-voltage design: The operating rate of an asynchronous pipeline is limited by the slowest cycle stage. Therefore, the fast cycle stages can slow down the cycle speed by low down the supply voltage of logic blocks. However, the modified cycle speed cannot slower than the slowest cycle stage, or it would deteriorate the pipeline performance.

TABLE I
CODE TABLE OF DUAL-RAIL ENCODING

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

* Codeword (1,1) is not used

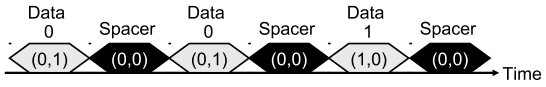

Fig. 4. Example of data transfer based on 4-phase dual-rail protocol.

Fig. 5 shows a handshake timing when $t_{l0} + t_{h0} < t_{l1} + t_{h1}$. The handshake timing is used to explain the multi-voltage design on logic block in stage0. Assuming that low down the supply voltage of logic block in stage0 impose a delay time Δt, and the delay time $t_{l0} + t_{h0} + \Delta t$ is still smaller than $t_{l1} + t_{h1}$.

At the first start point, the output of C-element is low. Req signal becomes high (require data) after the delay time $t_{l1} + t_{h1}$. In high, or normal, supply voltage, ack signal becomes high (data ready) after the delay time $t_{l0} + t_{h0}$. Because $t_{l0} + t_{h0} < t_{l1} + t_{h1}$, ack signal becomes high earlier than req signal. Ack signal needs to wait req signal for $t_{wait} = (t_{l1} + t_{h1}) - (t_{l0} + t_{h0})$. If low down the supply voltage of logic block in stage0, the logic delay becomes $t_{l0} + \Delta t$. Because $\Delta t < t_{wait}$, ack signal satisfies the deadline of second start point, which does not affect the pipeline cycle speed. Therefore, low supply voltage can be assigned to logic block in stage0 to save power.

In other conditions, low down the supply voltage of logic block in stage0 would affect the circuit performance. For example, if $\Delta t > t_{wait}$, the low supply voltage makes ack signal arrive later than req signal. This would postpone the second start point and increase the pipeline cycle time.

3) Self-adaptive voltage controller: According to the handshake timing analysis in previous section, req signal can be considered as a standard of timing reference. Comparing the arriving time of ack signal and req signal, a pipeline stage's cycle speed can be evaluated. If the arriving time of ack signal is earlier than req signal, the pipeline stage is a fast cycle stage. The pipeline stage can adjust the cycle speed by controlling the supply voltage of the logic block. A low supply voltage would impose some delay on the logic block. If the imposed delay does not make the arriving time of ack signal later than req signal, the low supply voltage can be applied on the logic block. As a result, self-adaptive voltage control can be realized by comparing the imposed delay on the logic block and the different arriving time between ack signal and req signal.

Fig. 7 shows the proposed self-adaptive voltage controller. Delay element has a delay time Δt which is equal to the imposed delay on a logic block when the low supply voltage is applied. When ack signal becomes high, the signal needs

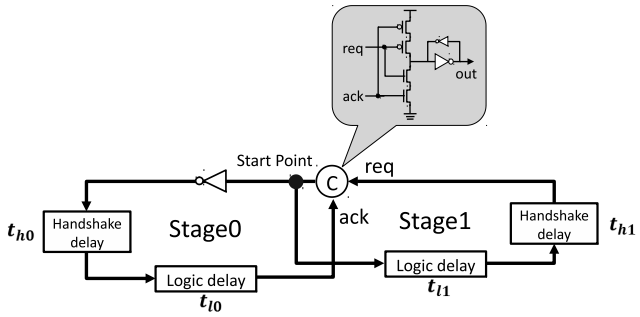

Fig. 5. Handshake delay model.

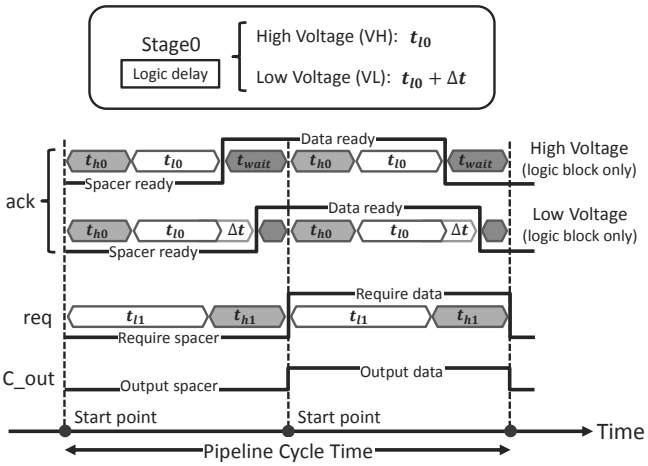

Fig. 6. Handshake timing ($t_{l0} + t_{h0} < t_{l1} + t_{h1}$).

a delay time Δt to arrive at domino AND gate. If req signal becomes high during Δt, domino AND gate maintains untriggered and the supply voltage uses high voltage. If req signal is still low, domino AND gate outputs 1 and the supply voltage switches to low voltage. When the voltage assignment task is finished, the assignment information is stored in latch and the controller is disabled to save power.

4) Logic block structure: Fig. 8 shows the structure of the proposed LB. There are two power domains in the LB. The gray region shows the multi-voltage power domain, which consists of a 4-input 1-output dual-rail LUT, a RS latch, and an OR gate. Handshake circuits and self-adaptive voltage controller are in the high, or normal, voltage power domain. The delay element in self-adaptive voltage controller is sized equal to the imposed delay when low voltage is chosen in the multi-voltage domain. This structure guarantees the correctness of self-adaptive multi-voltage control and makes level converter unnecessary.

High supply voltage of handshake circuits prevents the low voltage signal transfer on routing lines. In FPGAs, the routing directions of the input and output ports of LBs are controlled by programming CBs and SBs. Therefore, the reconfigurable routing line has an unpredictable and considerable delay. The signal voltage would have a great impact on this delay. When low voltage is applied on handshake circuits and low voltage

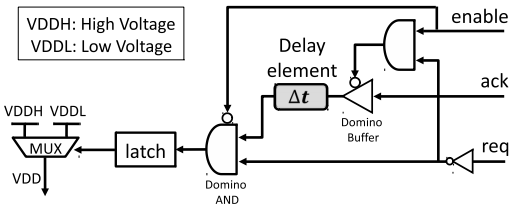

Fig. 7. Self-adaptive voltage controller.

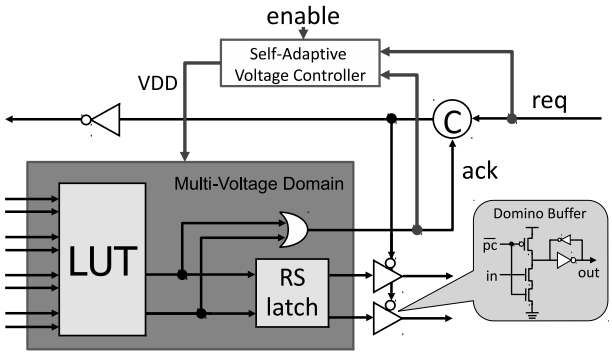

Fig. 8. Logic block structure.

signal is transferred on the routing line, the imposed delay becomes unpredictable. As a result, the self-adaptive multi-voltage control cannot correct work. Although level converter can be used to convert low voltage signal to high voltage signal, it increases delay and power overheads.

In the proposed LB, the interface between low voltage block and high voltage block is done by using domino gate. Fig. 8 shows that the outputs of RS latch are connected to domino buffers. Because C-element always outputs high voltage signal, domino buffer protects short-circuit current occur when RS latch outputs low voltage signal. The domino buffer in self-adaptive voltage controller serves the same function when the ack signal is low voltage signal. In addition, C-element can also protects short-circuit current occur since the req signal is always high voltage signal. As a result, level converters are unnecessary in the proposed FPGA.

III. Evaluation

The proposed FPGA is designed and evaluated by using HSPICE in a 65nm design technology. The multiple supply voltages use 1.2V for high voltage (VDDH), and 1V for low voltage (VDDL) as an example.

Table II shows the evaluation results of the proposed LB. Compared to normal LB with normal supply voltage, the energy consumption is reduced by 25.3% and the data processing speed is reduced by 47.8% when the proposed LB uses VDDL as the supply voltage. Although the proposed LB in VDDH has same supply voltage with the normal LB, its data processing speed is a little slower. This is because the voltage selecter (MUX) causes some voltage drop. The actual voltage in multi-voltage domain is a little lower than 1.2V. The low voltage slightly slows down the data processing speed, but also reduces some energy consumption. The evaluation is done when the self-adaptive voltage controller is disabled.

TABLE II
Evaluation results of the proposed logic block

	Normal Supply Voltage	Multiple Supply Voltage	
Supply Voltage	VDD: 1.2V	VDDH: 1.2V	VDDL: 1V
Transistor Count	336	391	
Processing time/data-set	450ps	500ps	665ps
Energy/data-set	0.146pJ	0.142pJ	0.109pJ

IV. Conclusion

This paper introduces a low-power FPGA based on self-adaptive multi-voltage control. A self-adaptive voltage controller is designed and embedded in each LB. The controller evaluates the deadline of each LB and selects a low voltage if it does not violate the deadline. This control scheme saves power without deteriorating the pipeline performance. Level converters are unnecessary in the proposed FPGA which has a simple and efficient architecture.

Acknowledgment

This work is supported by JSPS KAKENHI Grant Number 25·6949, 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. Takahashi et.al., "A 60-mW MPEG4 Video Codec Using Clustered Voltage Scaling with Variable Supply-Voltage Scheme," In IEEE Journal of Solid-State Circuits, Vol. 33, No. 11, Nov. 1998.

[2] F. Li, D. Chen, L. He, and J. Cong, "Low-Power FPGA using Pre-Defined Dual-Vdd/Dual-Vt Fabrics," In Proceedings of ACM/SIGDA International Symposium on Field-programmable gate arrays, 2003.

[3] R. Payne, "Asynchronous FPGA architectures," IEE Proceedings Computers and Digital Techniques, Volume 143, Issue 5, pp. 282-286, Sep. 1996.

[4] J. Teifel and R. Manohar, "An asynchronous dataflow FPGA architecture," IEEE Transactions on Computers, Volume 53, Issue 11, pp. 1376-1392, 2004.

[5] P. Chow, Soon Ong Seo, J. Rose, K. Chung, G. Paez-Monzon and I. Rahardja, "The design of a SRAM-based field-programmable gate array-Part II: Circuit design and layout," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, Volume 7, Issue 3, pp. 321-330 1999.

[6] J. Sparsø and S. Furber, "Principles of Asynchronous Circuit Design: A Systems Perspective," Kluwer Academic Publishers, 2001.