IEEE Asian Solid-State Circuits Conference
November 12-14, 2007 / Jeju, Korea

# A Field-Programmable VLSI Based on an Asynchronous Bit-Serial Architecture

Masanori Hariyama, Shota Ishihara, Chang Chia Wei and Michitaka Kameyama
Graduate School of Information Sciences, Tohoku University
Aoba 6-6-05, Aramaki, Aoba, Sendai, Miyagi,980-8579, Japan
Email: {hariyama@, ishihara@kameyama., chang13@kameyama., kameyama@}ecei.tohoku.ac.jp

*Abstract*— **This paper presents a novel asynchronous architecture of Field-programmable gate arrays (FPGAs) to reduce the power consumption. In the dynamic power consumption of the conventional FPGAs, the power consumed by the switch blocks and clock distribution is dominant since FPGAs have complex switch blocks and the large number of registers for high programmability. To reduce the power consumption of switch blocks and clock distribution, asynchronous bit-serial architecture is proposed. To ensure the correct operation independent of data-path lengths, we use the level-encoded dual-rail encoding and propose its area-efficient implementation. The proposed field-programmable VLSI is implemented in a 90nm CMOS technology. The delay and the power consumption of the proposed FPVLSI are respectively 61% and 58% of those of 4-phase dual-rail encoding which is the most common encoding in delay sensitive encoding.**

## I. Introduction

Field-programmable gate arrays (FPGAs) are widely used to implement special-purpose processors. FPGAs are cost-effective for small-lot production and flexible because functions and interconnections of logic resources can be directly programmed by end users. Despite their design cost advantage, FPGAs impose large power consumption overhead compared to custom silicon alternatives [1]. The overhead increases packaging costs and limits integrations of FPGAs into portable devices.

One efficient way for low power is asynchronous architecture, where timing is managed locally (as opposed to globally with a clock system as in synchronous architecture). Asynchronous design can reduce power consumption by avoiding two of the problems of synchronous design:

- all parts of a synchronous design are clocked, even if they perform no useful function;
- the clock line itself is a heavy load, requiring large drivers, and a significant amount of power is wasted just in driving the clock line.

There are synchronous solutions to these problems such as clock-gating. However, the solutions are complex and the problems can often be avoided with no extra effort or complexity when asynchronous design.

Recently, asynchronous architecture is employed in microprocessors such as ARM and some ASICs for mobile applications. In such custom VLSIs, bundled-data encoding is adapted because of its small hardware overhead. The bundled-data encoding requires the explicit insertion of delay in a control-signal wire to ensure that a request is never received before the bundled data value is valid. However, the bundled-data encoding is not suitable for reconfigurable VLSIs such as FPGAs since it is sensitive to variations of data-path delays.

To ensure the correct operation independent of data-path lengths, we use the level-encoded dual-rail encoding and propose its area-efficient implementation. As long as we know, this is the first implementation of a FPGA based on the LEDR encoding.

This paper presents a novel asynchronous architecture of Field-programmable gate arrays (FPGAs) to reduce the power consumption. In the dynamic power consumption of the conventional FPGAs, the power consumed by the switch blocks and clock distribution is dominant since FPGAs have complex switch blocks and the large number of registers for high programmability. To reduce the power consumption of switch blocks and clock distribution, asynchronous bit-serial architecture is proposed. To reduce the overhead of the LEDR encoding, it is combined with the bit-serial architecture which minimizes the complexity of switch blocks.

## II. Architecture

### A. Fine-grained pipelining bit-serial architecture

As shown in Fig. 1, the FPVLSI consists of a mesh-connected cellular array based on a bit-serial architecture to reduce the complexity of the switch block. As described below, we emply dual-rail encoding which is sutable for reconfigurable VLSIs. Hence, bit-serial architecture requires 3 wires: 2 for a data value and request, 1 for acknowledge. Each cell is connected to only four adjacent cells, and the number of programmable switches is reduced in comparison with the typical FPGA. The specification of the functionality of a logic block provides a great impact on the complexity of the switch block, the area of the logic block, and the delay of the logic block. The higher functionality of the logic block requires the larger number of inputs and outputs of a logic block. This increases the complexity of the switch block. On the other hand, the higher functionality of the logic block reduces the number of the logic blocks required for implementing a target function, and reduces the total delay. Based on this observation, the functionalities of the fine-grain logic block are specified as follows.

- 1-bit addition with carry storage.

1-4244-1360-5/07/$25.00 

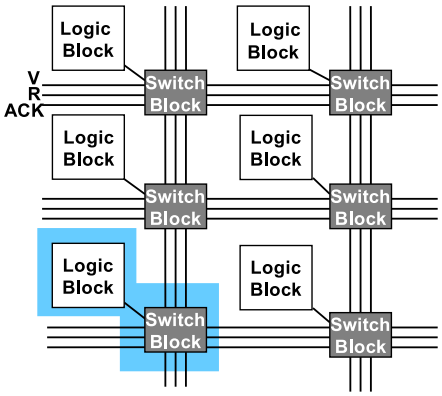


Fig. 1. Overall architecture.

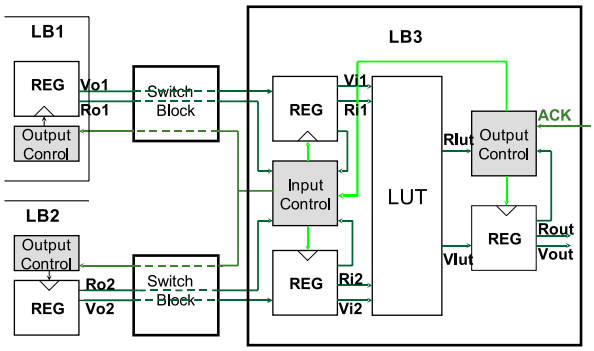


Fig. 2. Block diagram of the logic block.

- Arbitrary logic function of two inputs.
- 1-bit storage.

The specification allows to perform a bit-serial addition on a single cell without carry ripple, and the high functionality can be achieved with the minimum number of inputs and outputs of the logic block. This results in further reduction of interconnection complexity. Figure 2 shows the block diagram of the logic block. In the case of a single context, the performance of the synchronous fine-grained bit-serial architecture is more than two times higher than that of the conventional FPGA architecture under a constraint of the same chip area[2].

### *B. Asynchronous architecture based on LEDR encoding*

Asynchronous encoding schemes are classified into

- Bundled-data encoding
- Delay insensitive encoding (usulally dual-rail encoding)

Figure 3 shows the overall architecture for the bundled-data encoding. The bundled-data encoding splits request and value into separate wires. The value is encoded as in a synchronous circuit using $N$ wires to denot 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 delay in the request wire(denoted by 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 $N$ value wires. Hence, to transfer an $N$-bit value, only $N + 2$ wires are required.

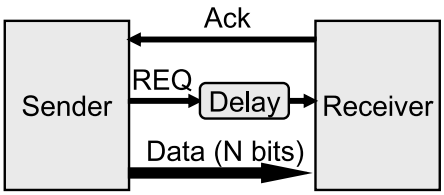


Fig. 3. Bundled-data encoding for $N$-bit data.

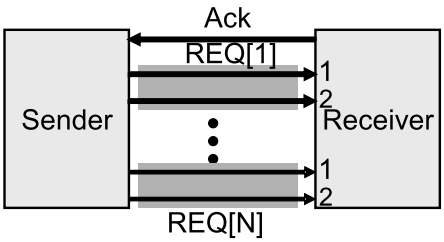


Fig. 4. Dual-rail encoding for $N$-bit data.

The major disadvantage 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 layouts of wires. On the other hand, in reconfiguralble VLSIs such as FPGAs, it is not easy to always meet the constraint since the data path is programmable.

The delay insensitive encoding makes value implicit in the request and no delay insertion is therefore required. Hence, the delay insensitive encoding is the ideal one for reconfigurable VLSIs. The most common delay insensitive encoding is dual-rail encoding. Figure 4 shows the overall architecture for dual-rail encoding. The dual-rail encoding uses two request wires to send a single bit of data. Hence, to transfer an $N$-bit value, $2N + 1$ wires are required. The disadvantage of the dual-rail encoding is the large hardware overhead since it requires as twice wires as the synchronous manner. The bit-serial architecture described in the previous section is one most efficient way of reducing the hardware overhead of wires because of its inherent wire overhead.

There are two major methods for dual-rail encoding:

- 4-phase dual-rail encoding
- Level encoded 2-phase dual-rail encoding(LEDR)

Figure 5 shows the code table of the 4-phase dual-rail encoding, which is the most common one in the dual-rail encoding. Figure 6 shows the example where data values 0, 0 and 1 are transferred. The main feature is that the sender sends spacer $(0, 0)$ after a data value. The receiver knows the arrival of a data value by detecting the change of either bit: 0 to 1. The drawback of the 4-phase dual-rail encoding is low throughput because of insertion of spacers.

The LEDR encoding enhances the throughput of the delay insensitive encoding[3]. Figure 7 shows the code table of the LEDR encoding. Note that each data value has two types of code words with different phases. For example, data value 0 is encoded as $(0, 0)$ in phase 0 and $(0, 1)$ in phase 1. The code word consists of $V$(Value bit) and $R$(Redundant bit). The value $V$ is encoded as in a synchronous circuit. The redundant bit $R$ is defined by EXOR-ing $V$ and $Phase$ so that $R$ includes

| | Code word (T, F) |
|---|---|
| Data 0 | (0,1) |
| Data 1 | (1,0) |
| Spacer | (0,0) |

* Code word (1,1) is not defined

Fig. 5. Code table of the 4-phase dual-rail encoding.

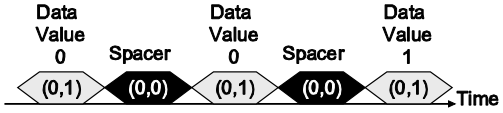


Fig. 6. Example of 4-phase dual-rail encoding.

| | | Code word (V, R) |
|---|---|---|
| Phase 0 | Data 0 | (0,0) |
| | Data 1 | (1,1) |
| Phase 1 | Data 0 | (0,1) |
| | Data 1 | (1,0) |

Fig. 7. Dual-rail encoding with 2-phase signaling.

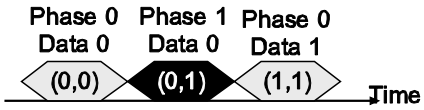


Fig. 8. Example of LEDR encoding.

the information on Phase and consecutive code words get different only by hamming distance 1. Figure 8 shows the example where data values 0, 0 and 1 are transferred. The main feature is that the sender sends data values alternately in phase 0 and phase 1. The receiver knows the arrival of a data value by detecting the change of phase, and data values are continuously transferred between the sender and the receiver without any break. The throughput is doubled in an ideal case in comparison with the 4-phase dual-rail encoding. The drawback of the LEDR encoding is that it requires slightly complex hardware to support the high throughput.

In the follwing, we describe the area-efficient implementation of the FPVLSI for the LEDR encoding. For the LEDR-based FPVLSI, the major consern is desiging the compact LUT. Figure 9 shows the conventional multiplexer-based LUT for the LEDR, where only $V_{out}$ is illustrated, and another LUT is necessary to obtain $R_{out}$. Based on two 2-bit input $(V_a, R_a)$ and $(V_b, R_b)$, $V_{out}$ is determined. The previous output is kept by the feed back loop if the the combination of the inputs is invalid. For example, the combination of inputs with different phases is invalid. To make a correct output for such invalid combination of inputs, the number of multiplexers becomes large. In the LUT for $V_{out}$, 8 memory bits and 8 fed-back bits are used as inputs of the multiplexer-tree.

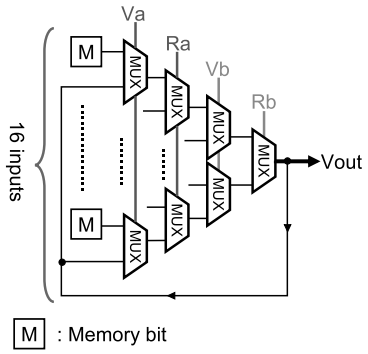


Fig. 9. The multiplexer-based LUT for the LEDR encoding (Only $V_{out}$ is illustrated).

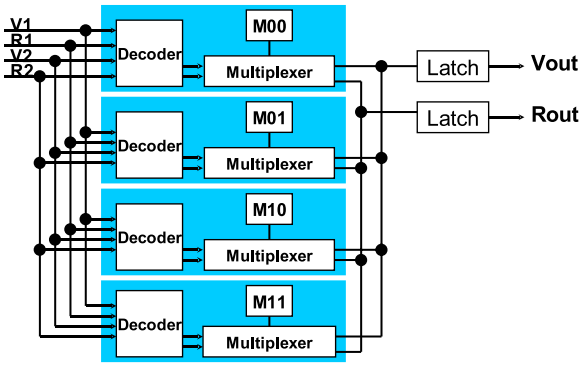


Fig. 10. LUT for the LEDR encoding based on hybrid of decoders and multiplexers.

To solve this problem, the LUT based on a hybrid of decorders and mutiplexers is proposd. Figure 10 shows the block diagram of the proposed LUT, which consists of 4 sub-modules. Each sub-module consists of a decorder, a multiplexer and a memory bit. The memory bit $M_{mn}$ is used to make the output for $V_a = m$ and $V_b = n$. Figure 11 shows the detailed structure of a sub-module. If the combination of inputs is invalid, the output of a sub-module becomes Hi-z according to the outputs of the decoder, and the latch output is not changed. Otherwise, a sub-module outputs the data value according to the value of the memory bit $M_{mn}$.

## III. Evaluation

The asynchronous FPVLSI is fabricated in a 90nm CMOS process. Figure 12 and Table I show the micro-photograph and the features of the FPVLSI, respectively. The chip includes 600 cells on 1.5mm×0.75mm area.

| Technology | 90nm CMOS |
|---|---|
| Supply voltage | 1V |
| Area | 1.5mm × 1.5mm (Core) |
| Cell size | 30 $\mu$m × 46$\mu$m |
| Number of cells | 20 × 30 |

TABLE I
Features of the FPVLSI.

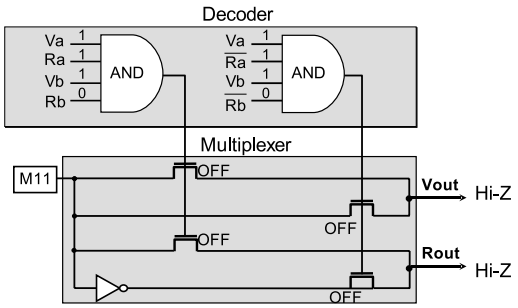


Fig. 11. Detailed structure of the sub-module of the LUT.

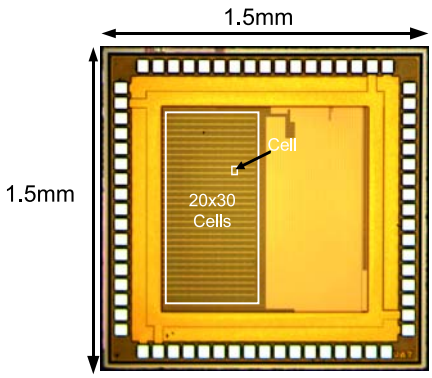


Fig. 12. Chip micro-photograph of the FPVLSI.

| | 2-rail and 4-phase | Proposed |
|---|---|---|
| The number of Tr. per cell | 453 | 513 |
| Delay of a cell | 0.90ns | 0.55ns |
| Power consumption per data set of a cell | 0.136pW | 0.079pW |

TABLE II
COMPARISON RESULT BETWEEN THE FPVLSI AND 2-RAIL AND A 4-PHASE ASYNCHRONOUS FPGA.

Table II shows the comparison between the FPVLSI based on LEDR and the FPGA based on the 4-phase dual-rail encoding. The number of transistors per cell of the FPVLSI is only by 13% larger than that of the 2-rail and 4-phase encoding. The delay of a cell and the power consumption per data set are respectively reduced to 61% and 58%.

## IV. CONCLUSION

We proposed the FPVLSI based on a asynchronous fine-grained pipelining architecture. The key technologies are the LEDR encoding to enhance the performance, and its area-efficient circuits . The asynchronous FPGA has a great potential for low power. The asynchronous architecture is very suitalble for fine-grained power gating since the data word includes information on whether the block is used or not. Therefore, the power gating control is easily achived using the information. On the other hand, in the synchronous FPGA, the fine-grained power gating may be impractical since it requires significant overhead of control circuits. The asynchronous FPVLSI with fine-grained power gating is now undergoing.

## ACKNOWLEDGMENT

This work is supported by VLSI Design and Education Center(VDEC), the University of Tokyo in collaboration with Synopsys, Inc., and Cadence Design Systems, Inc.. Authors thank Prof. Takahiro Hanyu, Research Institute of Electrical Communication, Tohoku University, Japan for his helpful support in CAD environment.

## REFERENCES

[1] V. George, H. Zhang, and J. Rabaey, “The design of a low energy FPGA,” Proceedings of the 1999 International Symposium on Low Power Electronics and Design, California, USA, pp.188–193, Aug. 1999.
[2] M. Hariyama, W. Chong, M. Kameyama,”Field-Programmable VLSI Based on a Bit-Serial Fine-Grain Architecture”,IEICE Trans. Electron., Vol.E87-C, No.11(2004)
[3] W. J. Bainbridge and S. B. Furber. Delay insensitive system-on-chip interconnect using 1-of-4 data encoding. In Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, pages 118.126. IEEE Computer Society Press, March 2001.