Chao Ma, Zi-bin Dai, Wei Li, Hai-juan Zang, 2017. A highly efficient reconfigurable rotation unit based on an inverse butterfly network. *Frontiers of Information Technology & Electronic Engineering*, **18**(11): 1784-1794.
http://dx.doi.org/10.1631/FITEE.1601265

A high-efficiency reconfigurable rotation unit based on an inverse butterfly network

Key words: Rotation operations; Self-routing; Control-bit generation algorithm; Inverse butterfly network

Corresponding author: Zi-bin Dai
E-mail: Daizb2004@126.com

Motivation

1. The operations of rotation and bit-level permutation are of prime importance for many emerging applications, such as cryptography, image processing, and bioinformatics.

2. Rotation and permutation operations are both bit-rearrangement operations, yet they are classified into different types and implemented separately. Therefore, an overhead resulting from additional hardware re-sources is incurred.

3. Although several studies unified these two operations in a hardware architecture (Hilewitz and Lee, 2009; Chang *et al.,* 2013), the routing algorithms for rotation operations are not efficient.

Main idea

1. We propose a highly efficient, low-cost algorithm based on the inverse butterfly network, which can generate control bits for the rotation, bi-directional rotation, and parallel sub-word rotation operations

2. A novel hardware unit called the highly efficient reconfigurable rotation unit (HERRU) is designed. It can supports a large number of bi-directional sub-word rotation operations.

Method

1. According to the characteristics of binary address change, we propose a generation algorithm for crude control bits for rotations based on the ‘XOR ’ operation
2. Study on the change rules of data position when they are executed rotation operations based on the topology of the inverse butterfly network. Then refine the above algorithm.
3. Design a hardware unit for rotation operations based on inverse butterfly network and evaluate its performance.

Major results

- Our HERRU has more functions than pervious designs.

Table 3 Operations supported by rotation shifters

Functional unit	Log rotation shifter	Chang's basic shifter	Hilewitz and Lee's basic shifter	Our basic shifter	Our L_R shifter	Our HERRU shifter
64-bit PEX		√*	√*	√*	√*	√*
64-bit IBFLY		√*	√*	√*	√*	√*
64-bit logic shift	√~	√	√	√~	√	√
64-bit rotation L_R	√~	√	√	√~	√	√
32-bit rotation L_R						√
16-bit rotation L_R						√
8-bit rotation L_R						√
4-bit rotation L_R						√

~ Represents that the relevant operation can only support single-direction rotation operation.
∗ Represents that the relevant operation can be supported by adding a new control bits generation circuit.
Chang's shifter (2013) is based on the inverse butterfly network using the SMIC 65-nm standard cell library.
Hilewitz and Lee's shifter (2009) is based on the inverse butterfly network using the TSMC 90-nm standard cell library.
PEX: parallel extraction; IBFLY: inverse butterfly.

Major results (Cont'd)

- Our HERRU can mostly achieve better performance than pervious designs.

Table 4 Comprehensive performance compar ison

Functional unit	Total area (um^2)	Relative area	Buffer area (um^2)	Latency (ns)	Relative latency	Efficiency (area×latency)
Log-rotation shifter	2846.16	1.00	551.88	0.53	1.00	1.00
Hilewitz and Lee's basic shifter (2009)	-	1.38	-	-	1.18	1.62
Chang's basic shifter (2013)	4293.32	1.51	758.32	0.60	1.13	1.71
Our basic shifter	2776.32	0.98	432.36	0.55	1.03	1.01
Our *L_R* shifter	3037.32	1.06	573.12	0.58	1.09	1.16
Our HERRU	3124.44	1.09	670.68	0.61	1.15	1.25

Conclusions

1. A highly low-cost, reconfigurable algorithm for the rotation, bi-directional rotation, and parallel sub-word rotation operations is proposed.

2. Experiments evaluated our HERRU with better functionality and higher efficiency than previously proposed methods.

3. We recommend our HERRU to be embedded in the PEX hardware circuit, which has been integrated into the Intel Haswell processor.