

Operating Frequency Improvement on FPGA Implementation of a Pipeline Large-FFT Processor

Ting-wai Siu*, Chiu-Wing Sham[‡] and Francis C. M. Lau^{*†}

^{*}Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong

[‡]Department of Computer Science, The University of Auckland, New Zealand

[†]Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen, China

Emails: dtwsiu@gmail.com, b.sham@auckland.ac.nz, encmlau@polyu.edu.hk

Abstract—In this paper, circuit complexity reduction in FPGA implementation of large N -point Radix-2² FFT with single-path delay feedback architecture is reported. Memory requirement of the FFT in the FPGA consists of two parts, the RAM data storage of the feedback in each stage of the data flow and the twiddle factors prepared as ROM for each complex multiplication. Through address rearrangement, the ROM sizes for the twiddle factors are significantly reduced with the removal of redundancy. The reduction ratio is about $\frac{1}{3(\log_4 N - 1)}$. As a result, the signal critical path is reduced and the system clock frequency is increased. The proposed architecture is validated by the implementations of 1K and 4K Radix-2² FFTs in an Altera Cyclone IV FPGA, EP4CGX22, which is the second lowest capacity FPGA of the low cost series. For the 1K- and 4K-point FFTs, the operating frequencies are 231.11 MHz and 215.75 MHz, respectively, approaching 250 MHz which is the speed limit of the I/O ports of the FPGA [1].

Keywords—Fast Fourier Transform implementation, FPGA, 1K-FFT, 4K-FFT.

I. INTRODUCTION

The Fast Fourier Transform (FFT) processor is one of the core components in signal processing. Small N -point FFT implementations are commonly found. An example in daily life is the WiFi chips which are produced in millions annually, having either 64-point or 256-point FFT core implemented. The requirement of large N -point FFT is found in niche areas such as astrophysics and radar applications. In recent years, there have been applications of large N -point FFT in consumer automotive radar [2], software-defined radio [3] [4] and ultra-wideband spectrum scanning [5], etc. In the direct conversion of radio frequency (RF) signal to digital form by various high-speed analog-to-digital converters [6] or even with down-conversion to bring the RF signal to baseband, the wide bandwidth requirement and the fast real-time applications have favored the application-specific integrated circuit (ASIC) or FPGA implementation over general purpose processors. The challenges are the processing speed and the cost involved. For example, the Sparse Fast Fourier Transform (SFFT) [7] claims achieving million-point FFT per second with a constraint of maximum 500 non-zero frequency coefficients. Its architecture includes the Radix2² single-path delay feedback (SDF) decimation in frequency (DIF) 4096-point FFT with a maximum operating frequency of 121.2 MHz. There are other studies to

reduce the circuit complexity with algorithms to minimize the size and power of the FFT processor using coefficient memory reduction [8] [9] and switching activity analysis schemes [10] [11].

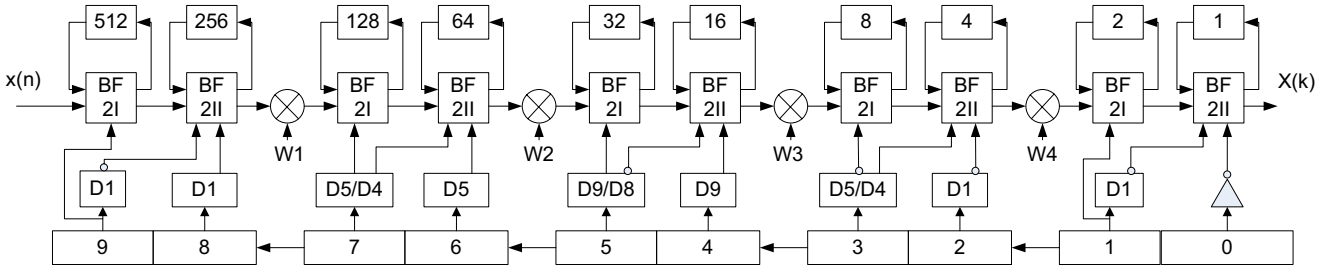
The architecture of Radix-2² SDF DIF N -point FFT [12] is considered as one with the least hardware resource requirement [13] in ultra-wideband scanning. This architecture is claimed to possess the features of simple control, $N - 1$ data memory size requirement with $\log_4 N - 1$ complex multipliers and $4 \log_4 N$ adders [14]. The twiddle factors may either be generated within the process using CORDIC algorithm or in advance and stored as Read Only Memory (ROM) which is easier in the implementation. In large N -point FFTs, the increase of memory size is significant and reduction mechanism is necessary. With pipeline process, the individual tasks per clock cycle are simple enough to support the increase of system clock frequency. However, the clock-cycle delays introduced in the data flow upset the arrangement of simple counter for the synchronization of various controls. Thus, an alternative arrangement is needed and is proposed in this paper.

In this paper, we begin with a brief explanation of the FFT architecture and describe the implementation of individual modules. Fixed-point arithmetic operation within the FFT is adopted. Scaling is introduced to convert the twiddle factors into integer values and the round-off error is reduced [15]. The reduction of the ROM sizes of the twiddle factors for the complex multiplications through address re-arrangement is explained. Implementations of 1K- and 4K-point FFTs with memory arrangement under the same architecture are presented. Controls with different delays are introduced to match the clock-cycle delays in the data flow pipeline. The physical implementation issues are discussed. We conclude with the presentation of the improved throughput performance with system clock frequency approaching the speed limit of the I/O ports of the FPGA which is 250 MHz.

II. THE PIPELINE FFT PROCESSOR ARCHITECTURE

The Radix-2² FFT equation using the common factor algorithm (CFA) to decompose the twiddle factors is shown as

$$X(k_1 + 2k_2 + 4k_3) = \sum_{n_3=0}^{N/4-1} \{H(k_1, k_2, n_3)W_N^{n_3(k_1+2k_2)}\}W_{N/4}^{n_3 k_3} \quad (1)$$

Fig. 1: Radix 2^2 SDF DIF pipeline FFT architecture for $N = 1024$.

with $k_1, k_2 \in \{0, 1\}$. $H(k_1, k_2, n_3)$ is named as Butterfly-II and can be expressed as

$$H(k_1, k_2, n_3) = B_{N/2}^{k_1}(n_3) + (-1)^{(k_1+2k_2)} B_{N/2}^{k_1}(n_3 + N/4) \quad (2)$$

where $B_{N/2}^{k_1}(n_3)$ and $B_{N/2}^{k_1}(n_3 + N/4)$ are named as Butterfly-I with

$$B_{N/2}^{k_1}(n_3) = x(n_3) + (-1)^{k_1} x(n_3 + (N/2)) \quad (3)$$

$$B_{N/2}^{k_1}(n_3 + N/4) = x(n_3 + (N/4)) + (-1)^{k_1} x(n_3 + (3/4)N). \quad (4)$$

Based on the above equations, the SDF pipeline FFT architecture is divided into several modules shown in Fig. 1. Referring to the figure, data are delayed and passed to the butterfly operation module via the shift-register feedback. The two butterfly operations are equivalent to the Radix-4 FFT butterfly operation and are followed by the non-trivial complex multiplication with the twiddle factor. These modules form one stage. There are $\log_4 N$ stages with different parameters and the last stage has no complex multiplication. Selection of the word length of the multiplication products are based on the balance between the minimization of round-off error and the circuit complexity. The twiddle factors are prepared in advance and stored in the ROMs. The sizes are reduced by the redundancy elimination.

A. ROM Approach of The Twiddle Factors Generation with Address Re-arrangement

The values of the twiddle factors are prepared in advance with MATLAB in HEX format [1] and stored in the ROMs. The twiddle factors $W_N^{n_3(k_1+2k_2)}$ from (1) with $k_1, k_2 \in \{0, 1\}$ and $n_3 = 0, \dots, N/4 - 1$ can be categorized into four groups — W^0 , W^{2k} , W^k and W^{3k} each of size $N/4$. The data of each group are passed onto the next stage of FFT with $N/4$ -point. Butterflies with the decomposed twiddle factors are shown in Fig. 2.

In the next stage, the twiddle factor size of each set of FFT is $1/4$ of $N/4$, i.e., $N/4^2$. The decomposition continues until it arrives at the 4-point FFT. Note that for the list of twiddle factors in the group W^k ($k = 0, \dots, N - 1$), it is sufficient to record those for $k = 0, \dots, N/4 - 1$. As shown in Fig. 3, we define the real part and the imaginary part as R_k and $-I_k$, respectively, $k = 0, \dots, N/4 - 1$. By re-labeling these real parts

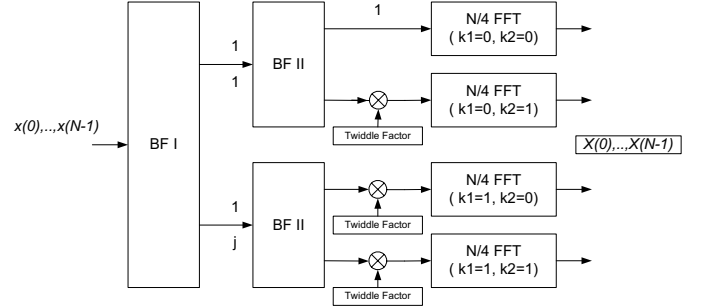


Fig. 2: Butterflies with decomposed twiddle factors

and imaginary parts as R_h and $-I_h$ where $h = 0, \dots, N/4 - 1$, the list of twiddle factors W^k can be defined as

$$\begin{aligned} W^k_{(k=0, \dots, N/4-1)} &= R_k - jI_k = R_h - jI_h \\ W^k_{(k=N/4, \dots, N/2-1)} &= -I_{k-N/4} - jR_{k-N/4} = -I_h - jR_h \\ W^k_{(k=N/2, \dots, 3N/4-1)} &= -R_{k-N/2} + jI_{k-N/2} = -R_h + jI_h \\ W^k_{(k=3N/4, \dots, N-1)} &= I_{k-3N/4} + jR_{k-3N/4} = I_h + jR_h. \end{aligned} \quad (5)$$

It is thus obvious that only $1/4$ of the data within the group is necessary.

The other lists of twiddle factor group W^{2k} and W^{3k} can also be represented by the same sets of values $\{R_h\}$ and $\{-I_h\}$ with some modifications. The values for the first half of the list of the twiddle factor group W^{2k} can be directly obtained from the two sets of values. The second half, which is on the second quadrant of the unit circle, is mapped back with the address pointers to the ROM as $2k - N/4$, i.e., the pointer to $N/4$ is pointed back to 0, the first location of the list. The modification of the values is the swapping of assignment and 2's complement operation, i.e., $-I_h - jR_h$. The list of twiddle factor group W^{3k} is divided into three parts. The first and second part follow the same arrangements. The third part with values in the third quadrant of the unit circle are mapped back with address pointers to the ROM as $3k - N/2$. The modification is 2's complement operation, i.e., $-R_h + jI_h$.

The total ROM size before reduction is $N \times (\log_4 N - 1)$

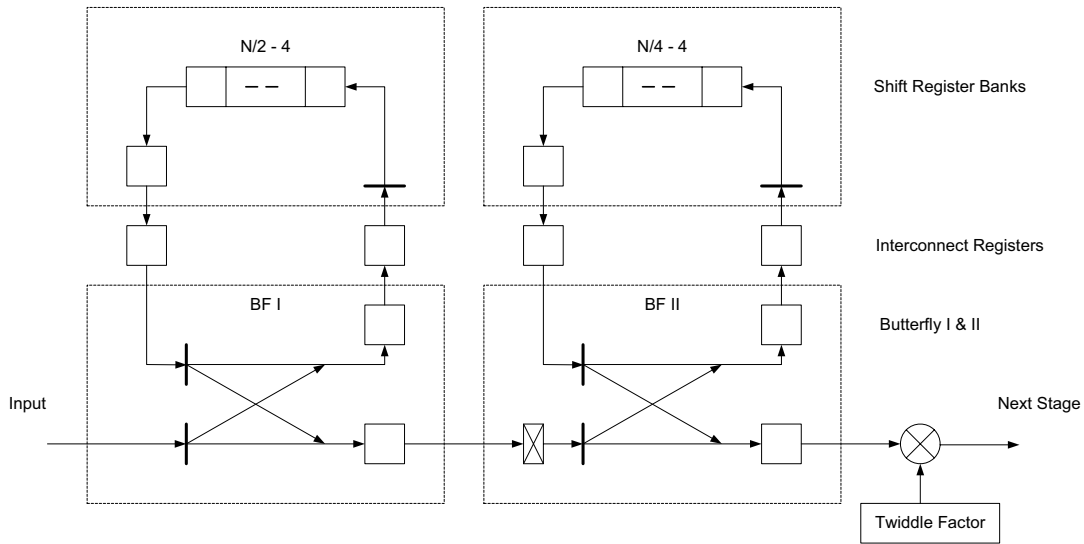


Fig. 4: Shift-register banks and butterflies of the first stage with interconnect registers.

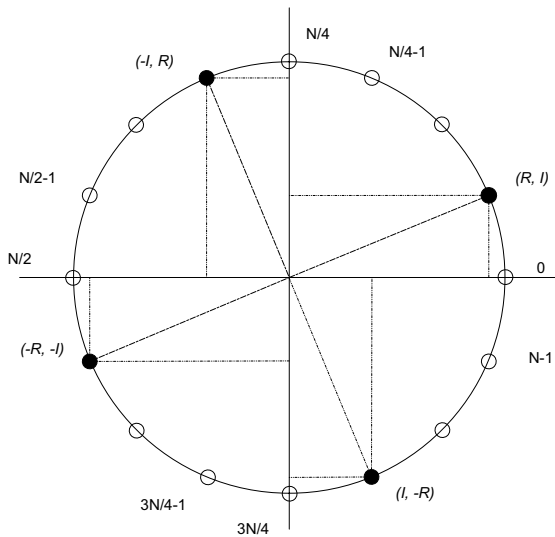


Fig. 3: Twiddle factors on a unit circle

while after reduction, the total ROM size becomes

$$\text{ROMsize} = \left(\frac{1}{4}\right)N + \left(\frac{1}{4^2}\right)N + \dots + \left(\frac{1}{4^{\log_4 N - 1}}\right)N. \quad (6)$$

The twiddle factor reduction ratio is therefore given by

$$\begin{aligned} & \frac{\left(\frac{1}{4}\right)N + \left(\frac{1}{4^2}\right)N + \dots + \left(\frac{1}{4^{\log_4 N - 1}}\right)N}{(\log_4 N - 1)N} \\ &= \frac{1 - \left(\frac{1}{4}\right)^{\log_4 N - 1}}{3(\log_4 N - 1)} \simeq \frac{1}{3(\log_4 N - 1)} \end{aligned} \quad (7)$$

The total numbers of twiddle factors for 1024-point and 4096-point FFTs under the memory re-arrangement are 340 and 1,364, respectively, compared with the standard requirement of $N(\log_4 N - 1)$ which are 4K and 20K, respectively. The reduction in ROM size requirement in FPGA implementation is therefore significant. The size reduction of the address multiplexers for the ROMs also leads to the reduction of the critical logic path delay in the FPGA implementation. The consequence is an increase in the maximum allowable system clock frequency which then improves the system throughput.

B. Shift Registers and Butterfly

In the SDF architecture, it is efficient to use shift registers to store the butterfly output. Shift-register implementation does not require the address multiplexing feature for the memory blocks. The saving is obvious. The use of shift registers is also reported in [13]. In hardware description language VHDL, the sequential process has registers holding output data. Therefore, the shift register bank sizes are $N/2^m - 2$, ($m = 1, \dots, \log_4 N - 1$). The shortfall of two registers is made up by the output registers of the butterfly and the corresponding shift-register bank. In large N -point FFT implementations, the number of stages is large. Consequently, the signal distances between the butterflies and the shift-register banks are large. The path delays become critical and affect the operating frequency. These

FFT size (N)	1024	4096
Logic elements	3295	4981
Registers	2453	3855
Embedded Multipliers	56	72
Memory bits	42,696	174,712
f_{MAX} (MHz)	231.11	215.75

TABLE I: Fixed-point radix-2² FFT implementation on Cyclone IV devices.

paths are shortened by introducing interconnect registers. With the input and output interconnect registers, the total number of registers in individual shift-register bank is further reduced by two. The first stage of the configuration is shown in Fig. 4.

Implementation of the butterfly operation is quite standard. The output data word length is expanded by one bit to cover the summation operation within the butterfly. Each butterfly operation is a sequential process in VHDL. Due to the register holding the output, a register is introduced between successive butterflies. As such, one clock-cycle delay per butterfly is inevitably introduced into the data flow pipeline.

C. Complex Multiplication and Signaling Control

The simple large single counter used to synchronize the controls of all the stages as stated in [12] is possible only if no clock delay is inserted into the data flow pipeline. As shown above, registers between stages have to be included, and so are the clock delays. In this paper, a two-bit-counter-per-stage signal control is adopted. Extra logics to cope with the clock-cycle delays at the stages are introduced for connecting up all the two-bit counters. Furthermore, the four-multiplier-and-two-adder configuration for implementing the complex multiplier introduces a two-clock-cycle delay. Together with the output data register of the complex multiplier, a total of four-clock-cycle delay is inserted. This is equivalent to a complete cycle of the two-bit counter, i.e., $n * 2^2$ cycles, and this arrangement avoids the extra complex logic requirement for the implementation of the carry increment to the next stage two-bit counter. This is the reason for our selection over the three-multiplier-and-five-adder configuration. With the various adjustments of the two-bit counter information, which are shown in Fig. 1, all stages of the pipeline process are synchronously controlled. Four-multiplier-and-two-adder instead of three-multiplier-and-five-adder configuration is chosen for the implementation of the complex multiplier. Besides allowing a matched number of clock-cycle delay, it reduces a clock-cycle delay at each complex multiplier. The system latency is hence improved.

III. IMPLEMENTATION RESULTS

The VHDL of the SDF FFT has been designed using the Quarter II from Altera. The design has been tested on ModelSim and evaluated on the Cyclone IV hardware development kit. The word length of the twiddle factors is chosen as 16 bits. The values are scaled up to maximize the utilization of the word length. The FFT is implemented with fixed-point

arithmetic and the round-off issue is handled with the natural growth of the word length. In each addition operation, the word length (of the sum) is increased by one. On the complex multiplications, one bit increment of the word length is also allowed for resolution improvement. With the input data word-length of the 1024-point FFT selected as 14 bits, the output data word-length is 28 bits. With the same input data word-length for the 4096-point FFT, the output data word-length is 31 bits. The performance parameters are listed in Table I. The system clock frequencies achieved are 231.11 MHz and 215.75 MHz, respectively, for the 1024-point and 4096-point FFTs with device EP4CGX22. The achieved frequencies are much higher than the claim of 121.2 MHz in [13].

IV. CONCLUSION

The 1K- and 4K-FFTs are implemented with the radix-2² SDF pipeline fixed-point FFT algorithm in FPGA. The ROM sizes for the twiddle factors are reduced. The four-multiplier-and-two-adder configuration is chosen for the implementation of the complex multiplier with relatively simple control logic. The maximum allowable system clock frequency is increased and the throughput is improved compared with the reported results. The improved efficiency will contribute to the construction of multiple FFTs in a single low-cost FPGA for applications such as software-defined radio, ultra-wideband spectrum scanning or consumer automotive radar.

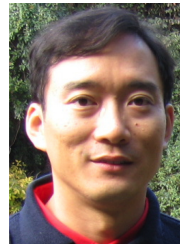
ACKNOWLEDGMENT

The work described in this paper was partially supported by a grant from the RGC of the Hong Kong SAR, China (Project No. PolyU 152088/15E) and by NSF of China (Grant No. 61372095).

REFERENCES

- [1] *Altera Cyclone IV Device Handbook*. Altera Corporation, 2014.
- [2] R. J. Evans, P. M. Farrell, G. Felic, H. T. Duong, H. V. Le, J. Li, M. Li, W. Moran, M. Moreland, and E. Skafidas, "Consumer radar: Technology and limitations," in *International Conference on Radar*, 2013, pp. 21–26.
- [3] J. Brakensiek, B. Oelkrug, M. Bückner, D. Uffmann, A. Dröge, M. Darianian, and M. Otte, "Software radio approach for re-configurable multi-standard radios," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, 2002, pp. 110–114.
- [4] F. K. Jondral, "Software-defined radio: Basics and evolution to cognitive radio," *EURASIP Journal on wireless communications and networking*, pp. 275–283, 2005.
- [5] H. Abdollahifakhr, N. Belanger, Y. Savaria, and F. Gagnon, "Power-efficient hardware architecture for computing Split-Radix FFTs on highly sparsely spectrum," in *IEEE International New Circuits and Systems Conference (NEWCAS)*, 2015, 2015, pp. 1–4.
- [6] M. Mishali, Y. C. Eldar, O. Dounaevsky, and E. Shoshan, "Xampling: Analog to digital at sub-Nyquist rates," *IET Circuits, Devices & Systems*, vol. 5, pp. 8–20, 2011.
- [7] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi, "GHz-wide sensing and decoding using the sparse Fourier transform," in *INFOCOM*, 2014, pp. 2256–2264.
- [8] F. Qureshi, S. A. Alam, and O. Gustafsson, "4K-Point FFT Algorithms based on optimized twiddle factor multiplication for FPGAs," in *Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, 2010, pp. 225–228.

- [9] H. Cho, M.-S. Kim, D.-B. Kim, and J.-U. Kim, "R²SDF FFT Implementation with Coefficient Memory Reduction Scheme," in *IEEE Vehicular Technology Conference*, 2006, pp. 1–4.
- [10] F. Qureshi and O. Gustafsson, "Twiddle factor memory switching activity analysis of Radix-2² and equivalent FFT algorithms," in *IEEE International Symposium on Circuits and Systems*, 2010, pp. 4145–4148.
- [11] S. Lee, D.-b. Kim, and S.-C. Park, "Power-efficient design of memory-based FFT Processor with new addressing scheme," in *IEEE International Symposium on Communications and Information Technology*, vol. 2, 2004, pp. 678–681.
- [12] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in *International Parallel Processing Symposium*, 1996, pp. 766–770.
- [13] A. Agarwal, H. Hassanieh, O. Abari, E. Hamed, D. Katabi *et al.*, "High-throughput implementation of a million-point sparse Fourier Transform," in *International Conference on Field Programmable Logic and Applications*, 2014, pp. 1–6.
- [14] S. He and M. Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," in *Custom Integrated Circuits Conference*, 1998, pp. 131–134.
- [15] R. W. S. Alan V. Oppenheim, *Discrete-Time Signal Processing (3rd Edition)*. Prentice-Hall, 2009.



Francis C. M. Lau received the BEng (Hons) degree in electrical and electronic engineering and the PhD degree from King's College London, University of London, UK. He is a Professor and Associate Head at the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. He is also a Fellow of IET and a Senior Member of IEEE.

He is the co-author of *Chaos-Based Digital Communication Systems* (Heidelberg: Springer-Verlag, 2003) and *Digital Communications with Chaos: Multiple Access Techniques and Performance Evaluation* (Oxford: Elsevier, 2007). He is also a co-holder of four US patents and one pending US patent. He has published over 280 papers. His main research interests include channel coding, cooperative networks, wireless sensor networks, chaos-based digital communications, applications of complex-network theories, and wireless communications. He was the Chair of Technical Committee on Nonlinear Circuits and Systems, IEEE Circuits and Systems Society in 2012–13. He served as an associate editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II in 2004–2005 and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I in 2006–2007, and IEEE CIRCUITS AND SYSTEMS MAGAZINE in 2012–2015. He has been a guest associate editor of INTERNATIONAL JOURNAL AND BIFURCATION AND CHAOS since 2010 and an associate editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II since 2016.



Ting-wai Siu received his MSc degree in Telecommunication Engineering from University of Essex, UK; and Engineering Doctorate degree from The Hong Kong Polytechnic University. He has been working in the industry for many years. Currently, he is with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University.



Chiu-Wing Sham received the B.Eng., M.Phil. and Ph.D. degrees from The Chinese University of Hong Kong, Hong Kong. He was a Research Engineer with Synopsys, China, and an Electronic Engineer with ASM (HK) before joining The Hong Kong Polytechnic University in 2006. Currently, he is a lecturer at the Department of Computer Science, The University of Auckland, New Zealand. He received the Best Paper Award in ISQED 2013. His research interests include design automation of VLSI, design optimization of digital VLSI systems and embedded

systems.