A New Field Programmable Gate Array: Architecture and Implementation

Hanjin Cho, Young Hwan Bae, Nak Woong Eum and Inhag Park

CONTENTS

ABSTRACT

A new architecture of field programmable gate array for high-speed datapath applications is presented. Its implementation is facilitated by a configurable interconnect technology based on a one-time, two-terminal programmable, very low-impedance anti-fuse and by a configurable logic module optimized for datapath applications. The configurable logic module can effectively implement diverse logic functions including sequential elements such as latches and flip-flops, and arithmetic functions such as one-bit full adder and two-bit comparator. A novel programming architecture is designed for supplying large current through the anti-fuse element, which drops the on-resistance of anti-fuse below 20 Ω. The chip has been fabricated using a 0.8-μm n-well complementary metal oxide semiconductor technology with two layers of metalization.

I. INTRODUCTION

Programmable integrated circuits have been used widely for circuit board designs. Programmable logic devices (PLDs), which are arrays of programmable logic arrays (PLAs) with programmable interconnections, offer high speed but limited logic gate capacity and limited logic design flexibility. Mask programmable gate arrays (MPGAs) require significant manufacturing time and incur high initial costs. Field programmable gate array (FPGA) outperforms PLDs in implementing large circuits and MPGAs in field programmability [1].

The main measures of FPGA are the system speed of implemented circuits and the logic capacity which tells the size of circuits that can be implemented. There are several factors which decide the performance of FPGA; the overall architecture, the switch device, routing structure and logic module architecture. Since the overall architecture consists of two-dimensional array of logic modules separated by horizontal and vertical channels, the physical size of a logic module and routing channels determine the array size. As the size of a logic module becomes larger, there are more chances that a larger module for a given circuit is implemented in one logic module, which reduces the number of logic levels, and hence the overall delay will be decreased. The switch device which connects signal lines produces routing delay due to its on-resistance. To reduce its on-resistance, the larger area is needed for switch devices; the bigger switch transistors for static random access memory (SRAM) or electrically erasable programmable read only memory (EEPROM)-based FPGAs and the bigger programming transistors to flow large current through anti-fuse in anti-fuse based FPGAs [2]. Therefore, there are many ways to design an FPGA, involving tradeoffs between the complexity and flexibility of both the logic modules and interconnection resources. A number of different types of FPGA are available now. They have their own unique features; for example, reprogrammability for SRAM [3], [4] and EEPROM-based FPGAs [5] and small routing delay of the anti-fuse based FPGAs [6], [7].

There are several benchmarks for FPGAs [8], [9], which aid system designers to decide which device is the most appropriate for implementing their design. These benchmarks measure the device performance for several applications. Some applications are highly sequential, and others are highly combinational. As the application of FPGAs becomes wider, FPGAs dedicated to special applications are needed.

Our FPGA architecture is optimized for high-speed datapath applications. The implementation is facilitated by a configurable interconnect technology based on a one-time, two-terminal programmable, very low-impedance switch element and a configurable logic module optimized for datapath circuits. The configurable logic module can effectively implement diverse logic functions including sequential elements such as latches and flip-flops, and arithmetic functions such as one-bit full adder

and two-bit comparator. Therefore, the maximum potential capacity of registers and arithmetic functions are enhanced, which are essential to datapath applications. A novel programming architecture is designed for supplying large current through the anti-fuse element, which drops the on-resistance of anti-fuse below 20 Ω. The resulting low delay (RC time constant) makes this device suitable for very high-speed applications. The chip has been fabricated using a 0.8-μm n-well complementary metal oxide semiconductor (CMOS) technology with two layers of metalization.

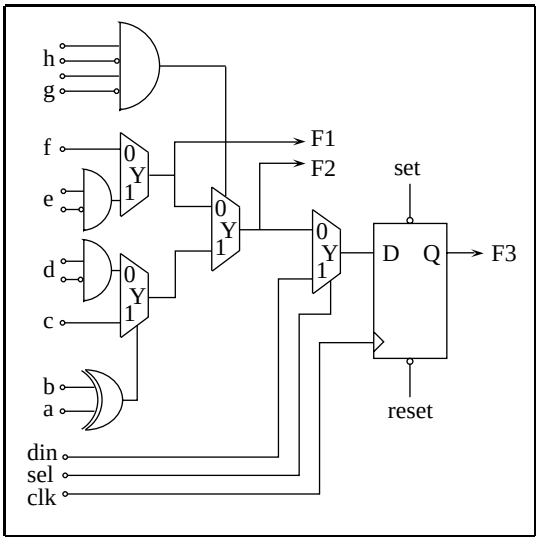

Fig. 1. The proposed logic module.

II. CHIP ARCHITECTURE

1. Logic module

As shown in Fig. 1, the configurable logic module has 12 inputs and 3 outputs. This module was chosen for its efficiency in implementing both combinatorial and sequential circuits and for its optimum utilization of routing resources. Among the 12 inputs, 4 inputs are inverted for inverting signals. It is composed of three two-to-one multiplexors, with an XOR gate on the first stage's select input and an AND gate on the last stage's select input. A large number of logic functions can be implemented by using an appropriate subset of the inputs and tying the remaining inputs high or low. This module implements all three-variable functions and up to some 12 variable functions. Figure 2 shows the number of logic functions which can be implemented in one logic module. Multiple outputs allow to pack unrelated logic functions into a single cell to maximize silicon utilization. A hardwired flip-flop is implemented in this logic module for se-

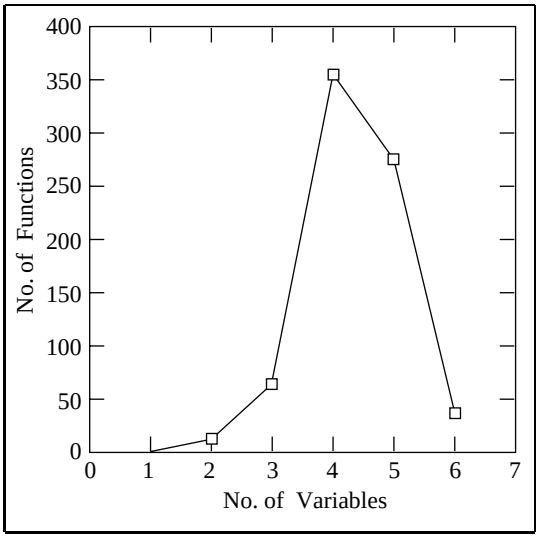

Fig. 2. The number of logic functions which can be implemented in one logic module.

Table 1. The mapping results of MCNC benchmark.

CIRCUIT	Number of logic module			CIRCUIT	Number of logic module		
	BDD	ours	ratio (%)		BDD	ours	ratio (%)
5xp1	39	30	-30.0	misex1	27	22	-22.7
9sym	27	18	-50.0	misex2	47	40	-17.5
alu4	156	113	-38.0	rd53	10	8	-25.0
apex1	590	439	-34.4	sqrt8	30	19	-57.9
apex2	91	66	-37.9	squar5	30	24	-25.0
apex3	617	551	-12.0	xor5	5	2	-150.0
apex4	1027	1047	+0.02	vg2	49	34	-44.0
apex5	434	98	-343.0	sao2	81	56	-44.6
bl2	42	30	-40.0	seq	509	383	-33.0
bw	84	69	-21.7	ex1010	1391	1048	-32.7
clip	51	39	-30.8	spla	180	142	-27.0
con1	14	9	-55.5	rd73	17	12	-41.7
cordic	29	15	-93.3	rd84	23	18	-27.8
cps	416	334	-24.5	table3	498	374	-33.1
duke2	194	145	-33.8	table5	427	324	-31.8
e64	95	95	0.0	misex3	247	189	-30.7
ex4p	281	201	-39.8	misex3c	271	193	-40.0
ex5p	156	127	-22.8	t481	17	11	-54.5
inc	59	47	-25.5	pdc	184	137	-47.0

quential circuits.

The efficiency of the proposed logic module is evaluated using MCNC benchmark [10]. Table 1 shows the result of benchmark comparing it to a logic module which has a two-input OR gate on the last stage's select input [6] using binary decision diagram (BDD) [11] mapping algorithm. The number represents the number of logic modules needed to implement each circuit. This result shows that the logic capability of a logic module can be dramatically enhanced by using appropriate control gates to the select inputs of the multiplexors.

It may also be connected to form latches or a flip-flop. Especially, this module can implement arithmetic functions such as one-bit adder or two-bit comparator in one logic module. Table 2 shows the input pattern to implement such macro functions. To form latches

Table 2. Various macro functions configured from the proposed logic module.

Logic Functions	Input Pattern							
	g	h	c	d	e	f	a	b
One-bit adder (F2=A+B+Ci)	1	1	Ci	Ci	B	Ci	A	B
Two-bit comparator (F2=1 when A0=B0 & A1=B1	$\overline{B0}$	1	A0	0	$\overline{A0}$	0	B1	A1
D-type flip-flop (F2=Data at pos. edge of CLK)	1	$\overline{CLK}$	Data	F1	F2	F2	CLK	0

or a flip-flop, the outputs are connected back to the input ports as shown in Table 2. Since each logic cell can be configured to one-bit full adder or registers, the maximum potential capacity of arithmetic functions and registers are enhanced for datapath designs.

2. Programmable interconnect architecture

The programmable interconnect architecture, shown in Fig. 3, consists of routing channels, anti-fuses and programming transistors. The routing channels contain routing tracks which contain predefined wiring segments of various length to enhance routability. This segmentation is based on statistics from a large number of design examples. Anti-fuse elements are located at the intersection of the horizontal and vertical wire segments, and between adjacent horizontal and vertical segments. To program an element, high voltage is applied across its terminals, while all other elements are subjected to no more than half that voltage. This is accomplished by a procedure that utilizes the wiring segments, programming transistors connected to each segment with shared control lines, module select transistors which select the logic module to be programmed with shared control lines, and serial addressing circuitry at the periphery of the device.

To program an anti-fuse A1 in Fig. 3, programming voltage Vpp is applied between horizontal track HT1 and vertical track VT1. This can be accomplished by applying Vpp to the supply line VSUP connected to the track VT1 and GND to the supply line HSUP connected to the track HT1 and turning on the module select transistors P3 and N3 and programming transistors P12 and N12. To prevent programming voltage from being applied to other tracks, the module selection transistor P2 and

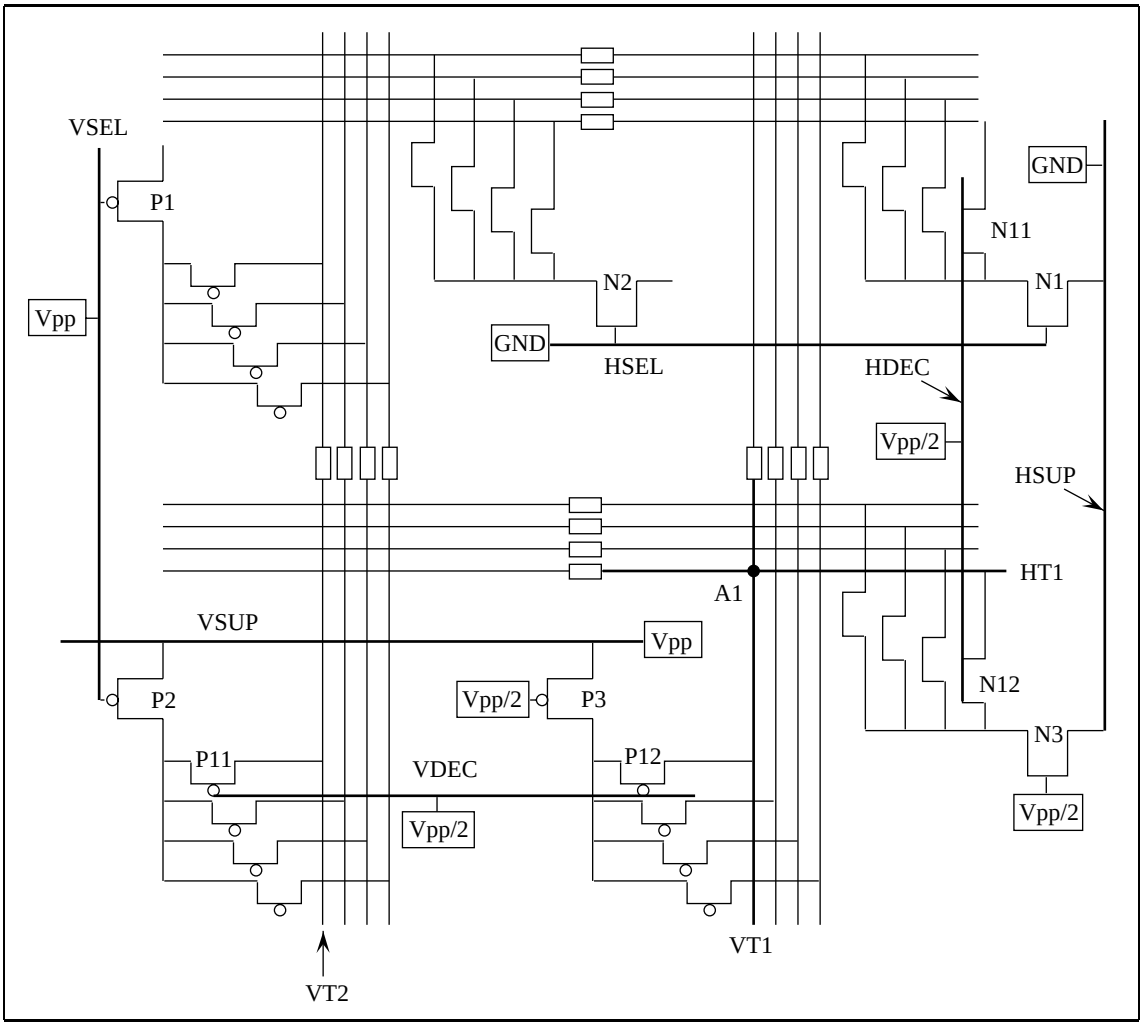

Fig. 3. The programmable interconnect architecture.

N1 of which control lines are tied common to transistors P3 and N3, respectively, are turned off by applying appropriate voltage to the control lines as shown in Fig. 3.

Since the programming voltage Vpp is above the breakdown voltage of the transistor of the 0.8 μm process, a special programming architecture is designed, which does not need extra processes for the programming transistors. By separating the supply voltage range into two ranges, P-channel metal oxide semiconductor (PMOS) transistor chains are used for the range of Vpp and Vpp/2 and N-channel metal oxide semiconductor (NMOS) transistor chain for the range of Vpp/2 and GND. Therefore, the maximum voltage applied be-

tween the terminals of program transistors is about Vpp/2, which is well below the breakdown voltage. Since the on-resistance of anti-fuse element depends on the current level during programming, programming architecture is designed for supplying large current through the element, up to 25 mA, at which the on-resistance drops to 20 Ω.

The device features a highly flexible and low-skew clock architecture. Clock signals can originate from the core array or directly from an I/O pin. Four independent clock networks are accessible by every cell in the array. These lines can also be used as global signals such as power-on enable or reset. The location and drive capability of clock buffers are optimized to reduce delay and skew.

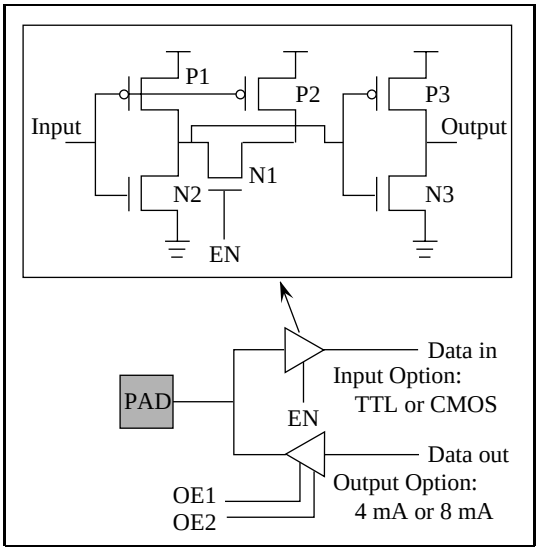

Fig. 4. I/O cell structure. Inset: configurable input buffer.

The I/O architecture is also quite flexible. Any I/O cell may be configured as input, output, or bi-directional I/O by programming the appropriate anti-fuse elements. Options are available to select drive strength, input voltage (CMOS or transistor transistor logic [TTL]), and weak pull-up. Figure 4 shows the I/O cell structure. By programming the control signals OE1 and OE2, the I/O cell can be configured as 4 mA or 8 mA output. The control signal EN is used to implement input options. The inset of Fig. 4 shows the schematics of configurable input buffer. When signal EN is high, transistor N1 is turned on. Therefore, the first inverter consists of PMOS transistors P1 and P2, and NMOS transistor N2. The logic threshold voltage of the first inverter goes down to TTL level. When signal EN is low, transistor N1 is turned off. So the input buffer operates at normal CMOS level.

3. Circuitry for chip test

An important aspect of the device is that its design is testable. There are two kinds of test modes; one is a scan test mode which test all the sequential elements in the device, the other is a probe test mode which enable the test of the programmed anti-fuse to verify whether it is programmed properly. The overhead of these testability is minimized by sharing some elements with the progrmming circuits.

Figure 5 shows the circuit of probing test. Register B1 controls the tristate buffer T1 and T2. Depending on the logic value of register B1, the logic value of register B2 is supplied to the track through tristate buffer T1 or the

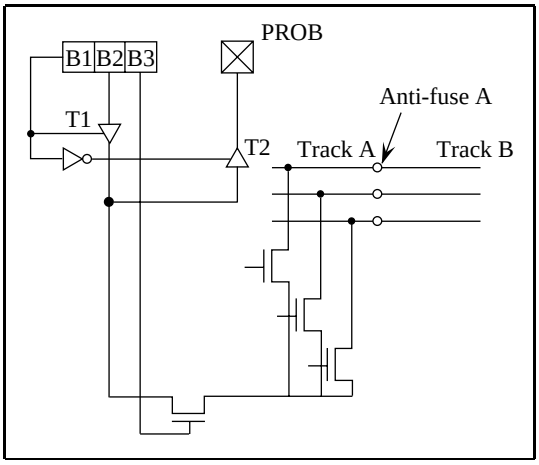

Fig. 5. The circuit diagram of probe mode.

value of the track is passed to the output PROB through tristate buffer T2. To test whether anti-fuse A is programmed properly, the program transistors connected to track A and B are turned on. After supplying ('1', 'logic value', '1') to the registers connected to track A and ('0', 'X', '1') to the registers connected to track B, the proper programming of anti-fuse A can be verified by checking if the logic value of output PROB follows the supplied logic value of the register B2. Also, this feature can be used to debug an implemented circuit by probing suspicious nodes.

4. Implementation

The layout of the device as shown in Fig. 6 consists of core and periphery sections. The peripheral sections are partitioned and drawn for pitch matching with the core section.

The core section is a two-dimensional array of configurable logic modules. The inputs of logic modules are connected to dedicated horizontal metal wire segments, which are extended across both left and right channels of the logic modules. Therefore, each input is accessible from either channel. Also, the common input ports can be used as routing resources, which enhance the routability and reduce the routing delay.

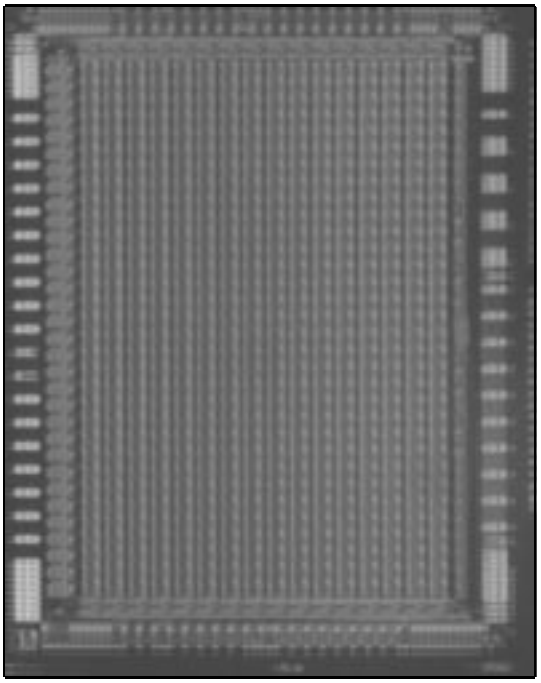

Fig. 6. The layout of the device.

The peripheral sections include serial addressing flip-flops, decoders for the control signals of programming transistors, and voltage level shifters for programming voltage. Since the peripheral modules are designed according to the width and length of the logic module, the different size can be generated by

simply adjusting according to the array size of logic module. For top and bottom peripheral sections, each block includes 12 output decoders and 12 level shifters for controlling program transistors of horizontal tracks of one logic module which has total 24 program transistors for horizontal tracks. For left peripheral section, each block has 8 output decoders and 8 level shifters for controlling program transistors for vertical tracks of one logic module. Each block decoder is enabled by setting the value of the dedicated register to 'high.'

The device has been fabricated using 0.8 μm N-well CMOS technology with two-layer metalization. The key characteristics of the device are listed in Table 3.

Table 3. The key characteristics of the device.

Chip size	1.14×0.8 cm^2
Package	180 PGA
Number of modules	512 (32×16)
Number of programming elements	312,000
Number of transistors	120,000
Logic module performance	3.8 ns
Maximum number of user defined pin	120
Configurable I/O buffers Input buffer Output buffer	 TTL or CMOS 4 mA or 8 mA

III. Conclusion

A novel architecture of FPGA is presented, which is optimized for high-speed datapath applications. By using very low-impedance switching element and flexible configurable logic module, high-speed DSP circuits and telecommunication circuits can be implemented effectively in this device. Also, a novel programming architecture preventing programming transistors from breakdown is presented, which needs no extra process for fabricating programming transistors.

REFERENCES

[1] Stephen D. Brown, et al., *Field-Programmable Gate Arrays*. Boston: Kluwer Academic Publishers, pp. 1-4, 1992.

[2] Esmat Hamdy, et al., "Dielectric based antifuse for logic and memory ICs," *International Electron Devices Meeting*, San Francisco, CA, Dec. 11-14, 1988, pp. 4.8-4.11.

[3] Barry K. Britton, et al., "Optimized reconfigurable cell array architecture for high-performance field programmable gate arrays," *IEEE 1993 Custom Integrated Circuits Conference*, San Diego, CA, May 9-12, 1993, pp. 7.2.1-7.2.5.

[4] Hung-cheng Hsieh, et al., "Third-generation architecture boosts speed and density of field-programmable gate arrays," *IEEE 1990 Custom Integrated Circuits Conference*, Boston, MA, May 13-16, 1990, pp. 31.2.1-31.2.7.

[5] Sau C. Wong, et al., "A 5000-gate CMOS EPLD with multiple logic and interconnect arrays," *IEEE 1989 Custom Integrated Circuits Conference*, San Diego, CA, May 15-18, 1989, pp. 5.8.1-5.8.4.

[6] Abbas El Gamal, et al., "An architecture for electrically configurable gate arrays," *IEEE J. of Solid-State Circuits*, vol. 24, no. 3, pp. 394-398, 1989.

[7] David Marple, "An MPGA-like FPGA," *IEEE Design & Test of Computers*, pp. 51-60, Dec. 1992.

[8] Bob Osann and Abbas El Gamal, "Compare ASIC capacities with gate array benchmarks," *Electronic Design*, pp. 93-97, Oct. 1988.

[9] Dennis McCarty, Donald Faria, and Peter Alfka, "PREP benchmarks for programmable logic devices," *IEEE 1993 Custom Integrated Circuits Conference*, San Diego, CA, May 9-12, 1993, pp. 7.7.1-7.7.6.

[10] S. Yang, "Logic synthesis and optimization benchmarks user guide - version 3.0," Microelectronic Center of North Carolina, Jan. 1991.

[11] R. Murgai, R. K. Brayton, and A. Sangiovanni-Vincentelli, "An improved synthesis algorithm for multiplexor-based PGAs," *ACM/SIGDA First International Workshop on Field-Programmable Gate Arrays*, 1992, pp. 97-102.

Hanjin Cho was born in Seoul, Korea on July 8, 1960. He received the B. S. degree in electronic engineering from Hanyang University in 1982. He received M. S. degree and Ph. D. degree in electrical engineering from New Jersey Institute of Technology in 1987, and from University of Florida in 1992, respectively. He joined ETRI in 1992, where he currently works in FPGA development.

Young Hwan Bae was born in Seoul, Korea on Oct. 29, 1962. He received the B. S. and M. S. degrees in electronic engineering from Hanyang University in 1985 and 1987, respectively. He joined ETRI in 1987, where he currently works in computer aided design for VLSI.

Nak Woong Eum received B.S. degree in electrical engineering from Kyungbook National University in 1984 and M.S. degree in electrical engineering from Korea Advanced Institute Science and Technology (KAIST) in 1987. He is currently working on Ph. D. degree in the Dept. of Electrical Engineering at KAIST, specializing in design automation for VLSI systems. He joined ETRI in 1987, where he currently works in FPGA CAD development. His current research interests include algorithm and CAD for field-programmable multichip systems, and layout synthesis.

Inhag Park was born in Seoul, Korea on Mar. 6, 1958. He received the B. S. and M. S. degrees in electronic engineering from Korea University in 1980 and 1983, respectively. He received Ph. D. degree in microelectronics from Institue of National Polytechnique at Grenoble, France, in 1992. He joined ETRI in 1982, where he currently works in computer aided design for VLSI.