Software Defined Radio Prototype (I) – System Design and Performance Evaluation

Hiroyuki Shiba†, Yushi Shirato, Hiroshi Yoshioka, and Ichihiko Toyoda

Abstract

A software defined radio (SDR) prototype has been developed based on a multiprocessor architecture. It successfully implements software for the Japanese personal handy-phone system (PHS: a 2G mobile system) and IEEE 802.11 wireless LANs, which have a much wider bandwidth than 2G systems. It uses a flexible-rate pre-/post-processor to achieve the required flexibility and wideband performance. This paper describes the design of the SDR prototype and evaluates its performance through experiments.

1. Introduction

Recent progress in digital signal processing devices enables the prototyping of software radio equipment [1], [2]. The software radio architecture and various primitive test beds have already been reported [3]-[5]. However, few papers have described software radio prototypes of actual wireless systems that include not only physical layer processes but also high-layer protocols. Moreover, although download procedures and the network architecture for over-the-air (OTA) downloading has been discussed [6], [7], neither the implementation of an OTA download protocol nor the performance evaluation of prototypes has been reported. In addition, system re-configuration, which is one of the most important characteristics of software radio, has not been reported.

In 2000, we reported an SDR prototype that supported only narrow bandwidth systems (several hundred kilohertz) such as PDC (Personal Digital Cellular) system and PHS (Personal Handy-phone System) [8]. Because pre-/post-processors (PPPs) are used to achieve high speed, real-time signal processing becomes a bottleneck: the prototype could not implement wideband wireless communication systems such as wireless LANs. This paper describes an advanced SDR prototype that uses a newly developed flexible-rate pre-/post-processor (FR-PPP) to offer improved bandwidth (>20 MHz) and flexibility. It can handle wireless LAN systems that use the direct sequence spread spectrum (DSSS) scheme. The prototype has successfully demonstrated switching between PHS and a wireless LAN (IEEE 802.11). It also implements OTA downloading.

† NTT Network Innovation Laboratories
Yokosuka-shi, 239-0847 Japan
E-mail: shiba.hiroyuki@lab.ntt.co.jp

2. Design of the platform

The prototype consists of three stages: the radio frequency (RF), intermediate frequency (IF), and baseband (BB) stages. The RF and IF stages consist of multiband analog circuits. Analog-to-digital (A/D) conversion is performed at the IF. The digital IF and BB stages consist only of programmable devices. Figure 1 shows a block diagram of the platform. Table 1 lists its major device specifications. It uses the multiprocessor architecture, shown in Fig. 2, which consists of four digital signal processors (DSPs), a central processing unit (CPU), and three flexible-rate pre-/post-processors (FR-PPPs). A 64-bit VME bus connects the DSPs, CPU, and external interface modules. The DSP module is a general-purpose DSP board consisting of four fixed-point arithmetic chips, each with a 200-MHz clock and maximum operating power of 1600 MIPS. The CPU is a

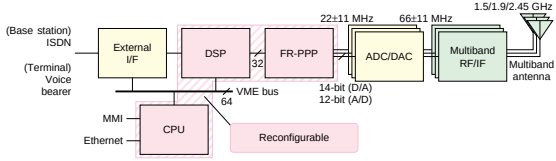

Fig. 1. Block diagram of the prototype.

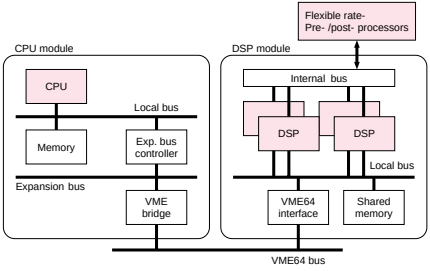

Fig. 2. Multiprocessor architecture.

Table 1. Major device specifications of the platform.

DSP	TMS320C6201×4 (1600×4=6400 MIPS, 200 MHz)
CPU	PowerPC750 (400 MHz)
OS	VxWorks 5.2 (real-time operating system)
ADC	IF undersampling (12-bit, 88 MSPS)

400-MHz PowerPC. The real-time operating system is VxWorks. Memories, a hard disk, and man-machine interface devices for user operation are connected to the CPU via local and expansion buses. For PHS operation, the external interface module provides an ISDN service in the cell station (CS) platform, and voice and bearer services in the personal station (PS) platform. This is the only difference between the CS and PS platforms. For the wireless LAN operation, the CPU module provides an Ethernet interface in both the access point (AP) and station (STA) platforms. The platform has three independent RF/IF branches to support simultaneous multimode operation and/or smart antennas in the future.

2.1 Multiband RF/IF and A/D and D/A circuits

The RF signal received by the antenna is down-converted by analog circuits to an IF signal and then A/D converted. A multiband RF/IF circuit based on the super-heterodyne scheme was developed. Multiband operation at 1.5/1.9/2.45 GHz was achieved using a single amplifier by switching bandpass filters, according to the PDC, PHS, and IEEE 802.11 standards. A multiband monopole antenna that resonates at these frequencies is used.

A/D conversion is performed by under-sampling the IF signal with center frequency of 66 MHz and bandwidth of 22 MHz using an analog-to-digital converter (ADC) with 12-bit resolution and sampling rate of 88 MSPS. An automatic gain control (AGC) circuit is set before the ADC. D/A conversion is performed using a digital-to-analog converter (DAC) with 14-bit resolution after up-sampling the BB signal with center frequency of 22 MHz and bandwidth of 22 MHz. The imaging at the center frequency of 66 MHz and bandwidth of 22 MHz is used as the IF signal.

2.2 Reconfigurable digital IF part

The IF corresponds to a clock-rate that is too high to be handled by the BB processors, so PPPs are used to achieve the required high-speed real-time digital processing, including filtering, waveform-shaping, and spectrum de-spreading.

Our new FR-PPP consists of field programmable gate arrays (FPGAs) and a direct digital synthesizer (DDS). Each branch uses two one-megagate FPGAs for pre- and post-processing. Figure 3 shows the configurations of the receiving blocks of a conventional PPP and the FR-PPP. Conventional PPPs (commercially available digital up/down converters) are composed of parameter-preset hard-wired circuits including various kinds of filters to support the wireless systems targeted [9]-[12]. Therefore, their circuit scale is excessive and their bandwidth is restricted to about 1–5 MHz. On the other hand, the circuit scale of FR-PPPs is much smaller because the FPGA can flexibly act as the filters needed for each system. In addition, while conventional PPPs use complicated interpolation circuits composed of a numerically controlled oscillator (NCO) and a re-sampler to support the various clock-rates of the targeted wireless systems, the DDS in the FR-PPP directly generates the required clock-rates in an arbitrary manner. This also reduces the circuit scale and offers high-speed operation. These breakthroughs enable a small circuit scale (75% smaller), a wide bandwidth (>20 MHz), and a very flexible SDR that can support wireless LANs as well as 2G systems such as PHS. Figures 4 and 5 show detailed configurations of the FR-PPP for PHS mode and wireless LAN mode, respectively [13].

2.3 Reconfigurable baseband (BB) part

The prototype uses a CPU and DSPs to perform BB processing and control. The CPU handles high-layer protocols including PHS call control and media access control (MAC) for the wireless LAN. Physical layer processes such as modulation and demodulation and voice coding and decoding are handled by the DSPs. Each DSP has computational power of 1600 MIPS, and both transmitting and receiving processes are performed by one DSP. Table 2 summarizes the function assignment to the processors.

3. Design of the software

3.1 Software architecture

The SDR prototype operates as the terminal of a specific wireless system after the system's software has been loaded into its processors. The programs written for the prototype reproduce almost all of the key operating functions of PHS and the DSSS scheme of IEEE 802.11 wireless LAN (infrastructure mode) offered by regular commercial terminals. PHS is a four-channel TDMA-TDD (Time Division Multiple Access, Time Division Duplex) wireless system that was standardized by Japanese RCR STD-28 [14]. The codec uses a 32-kbit/s ADPCM and the modulation/demodulation schemes are π/4-shift QPSK/incoherent detection. The IEEE 802.11 wireless LAN

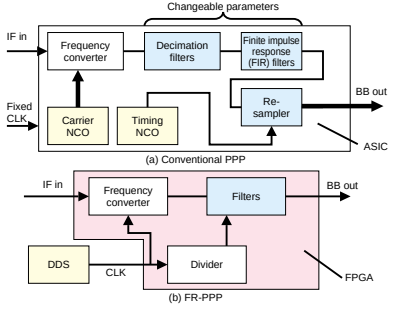

Fig. 3. Configurations of conventional PPP and FR-PPP (receiving blocks).

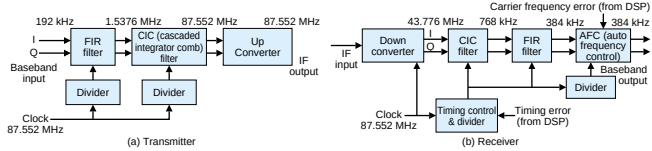

Fig. 4. FR-PPP configuration for PHS mode.

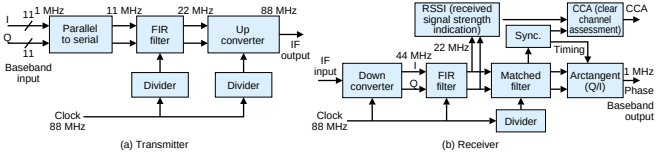

Fig. 5. FR-PPP configuration for wireless LAN mode.

Table 2. Function assignment to processors.

	Common	PHS mode	Wireless LAN mode
CPU	System control, GUI support, OTA download	Call control	Media access control
DSP		Modulation, Demodulation, Voice codec	Modulation, Demodulation, Spectrum spreading
FR-PPP		Channeling, Filtering	Channeling, Filtering, Spectrum de-spreading

supports multi-rate transmission by using the DBPSK and DQPSK modulation schemes [15]. Table 3 shows the major parameters of both systems.

The software architecture of SDRs has been discussed in the SDR forum [7]. Architecture selection strongly depends on the OS, API, and protocol stack, and affects program overhead and system performance. It also influences future expandability. (Note: In this paper, the term API is used in the strict sense defined in the SDR forum.)

Figure 6 shows the program component architecture of the prototype. A system control program, an OTA download program, and communication control programs run on the real-time operating system. The system control program handles the user interface, and system management and control. It executes the communication control programs and the OTA download program. The OTA download program, which is a CPU program, downloads communication control programs from the OTA download server.

The communication control programs are packaged binary files: each consists of CPU, DSP, and FR-PPP programs containing a wireless system protocol. The CPU and DSP programs were written in the C programming language and the FPGA configuration data of the FR-PPP was written in the Verilog hardware

Table 3. Major parameters of wireless systems implemented.

	PHS (RCR STD-28)	WLAN (IEEE 802.11)
Frequency	1.9 GHz	2.45 GHz
Access method	TDMA-TDD	CSMA-CA
Modulation	π/4-QPSK	DSSS
Air bit-rate	384 kbit/s	11 Mchip/s
User data-rate	2 kbit/s	1 Mbit/s, 2 Mbit/s
Omitted functions	Mobility management, authentication, billing, etc.	Power management, etc.

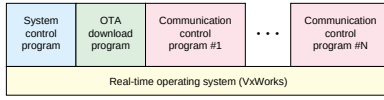

Fig. 6. Program component architecture.

definition language (HDL).

3.2 PHS program

In the PHS communication control program, the CPU program handles call control tasks and a DSP-task-management task. The priority of these tasks is set by the real-time operating system considering the processing time of each task. PHS imposes strict processing limits on the DSP tasks to achieve real-time communication because it uses the TDMA-TDD access scheme. Therefore, the scheduling of DSP tasks should be pre-assigned and the DSP loads of the tasks should be appropriately shared to avoid overloads. Table 4 shows an example of DSP task scheduling. The DSP-task-management task directs the execution of these DSP tasks via an API between the CPU and DSP programs as shown in Fig. 7.

First, the CPU program writes an API command and related parameters into a shared memory. Next it interrupts the DSP program, which is synchronized to TDMA slot timing. The interruption moves the access authority of the shared memory from the CPU to the DSP. The DSP program reads the shared memory to obtain the API command and parameters. After the DSP program has finished executing the task, the DSP program writes the result into the shared memory and interrupts the CPU program. This interruption returns the access authority to the CPU, and the CPU program recognizes the completion of the API command, and the CPU program's state transits. Real-time signal processing can be achieved by performing these controls in each TDMA slot.

3.3 IEEE 802.11 wireless LAN program

In the wireless LAN communication control program, MAC layer controls and physical layer processes are basically assigned to the CPU and DSP programs, respectively. Detailed descriptions of this are shown in [16].

3.4 Over-the-air download program

To implement OTA downloading, we developed an OTA download protocol and a download server. The protocol should be as independent of the lower layer as possible to allow it to support any wireless communication system, so our protocol is based on TCP/IP. Figure 8 shows its protocol stack. Since this protocol works over TCP/IP, it is not affected by a change in the communication mode. It not only downloads the software, but also authenticates and encrypts the data using SSL (Secure Socket Layer). To ensure secure downloads, the 128-bit next-generation block cipher "Camellia," which was co-developed by NTT and Mitsubishi Electric Corporation, was implemented as the chip algorithm of SSL [17]. Figure 9 shows the OTA download sequence. When the user requests a software download, the prototype attempts to connect to an OTA download server by

Table 4. Example of DSP task scheduling.

TDMA slot#	Transmitter	Receiver
1		
2	Voice coding	
3		(Downlink slot)
4		Clock recovery, detection, AGC, UW detection, and CRC
5		
6	Frame construction, Modulation	Voice decoding
7	(Uplink slot)	
8		

UW: unique word CRC: cyclic redundancy check

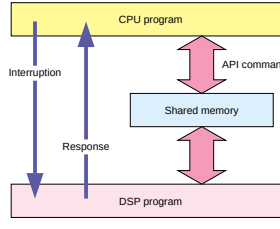

Fig. 7. API between CPU and DSP programs.

SSL, and the server authenticates the user by SSL. If user authentication succeeds, the encrypted data is transmitted and received within the SSL session. The prototype transmits terminal information including CPU specifications, DSP specifications, OS information, and vendor information to the download server using header fields as shown in Fig. 10. The download server lists only those software packages that are suitable for the prototype and the user selects the desired module for downloading. The prototype downloads and verifies the software. If verification succeeds, the prototype notifies the download server of download completion. Finally, the prototype releases the SSL session.

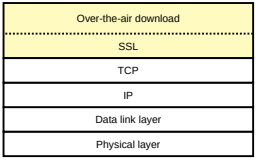

Fig. 8. Over-the-air download protocol stack.

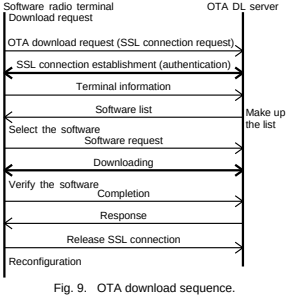

Fig. 9. OTA download sequence.

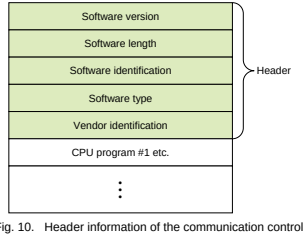

Fig. 10. Header information of the communication control program.

4. Performance evaluation

4.1 FR-PPP performance

To evaluate the basic physical-layer performance, we measured the output signals of the FR-PPP. The spectrum at the FR-PPP transmitter output for wireless LAN mode is shown in Fig. 11. We found that the measured spectrum satisfied the transmission spectrum mask provided in IEEE 802.11 (11 MHz< $|f-f_c|$<22 MHz; less than –30 dBr, $|f-f_c|$>22 MHz; less than –50 dBr) [15]. Figure 12 shows the receiver

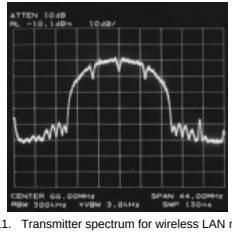

Fig. 11. Transmitter spectrum for wireless LAN mode.

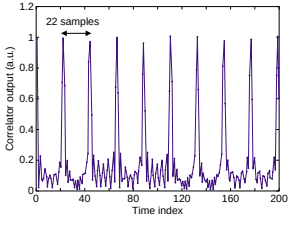

Fig. 12. Correlation output for wireless LAN mode.

output of the FR-PPP for wireless LAN mode. Peaks were monitored every 22 samples for 11 Mchip/s, which were twice-oversampled data.

The transmitter output spectrum for PHS mode is shown in Fig. 13. Figure 14 shows the constellation for PHS mode. The measured error vector magnitude was 1.9%. These results confirm that the proposed FR-PPP functioned correctly for both wireless LAN and PHS modes.

The utilization of the FPGA in each mode was less than 33% (Table 5), though a megagate-scale FPGA was utilized to allow for possible expansion of the FR-PPP design.

Table 5. Actual utilization of FPGA.

Mode	Utilization of FPGA [%]
Wireless LAN transmitter	4
Wireless LAN receiver	33
PHS transmitter	7
PHS receiver	10

4.2 System performance

Figure 15 shows the experimental setup used. The non-compressed packaged PHS and wireless LAN communication control programs were about 2 and 3 Mbytes, respectively. First, by loading the wireless LAN program and establishing STA-AP communication, we confirmed the data communication between two PCs connected to different SDR prototypes. Performance evaluation results of the wireless LAN mode are described in detail in the next paper in this issue [16]. Next, the PHS PS communication control program was download over the air from the OTA download server. PS software allocation (1.8 Mbytes) is shown in Fig. 16, which confirms that the PS software of PHS was successfully downloaded and software integrity was maintained.

After the OTA download had finished successfully,

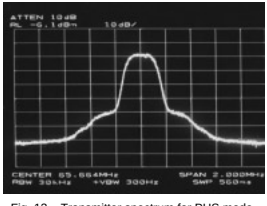

Fig. 13. Transmitter spectrum for PHS mode.

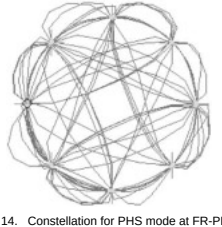

Fig. 14. Constellation for PHS mode at FR-PPP.

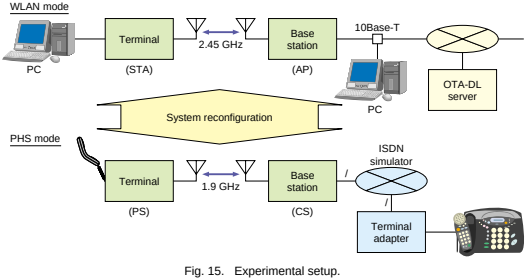

Fig. 15. Experimental setup.

the prototype was automatically reconfigured by the system control program, as shown in Fig. 17. First, the FPGA configuration control program for FR-PPP configuration was loaded into a DSP and started by the system control program. The system control program stored FPGA configuration data in the shared memory, which can be accessed by both the CPU and DSP programs. The FPGA configuration control program transferred FPGA configuration data to the FIFO (first in first out) buffer from the shared memory. The FR-PPP controller read data from the FIFO and installed the data to reconfigure the FPGA. After the FR-PPP configuration was complete, DSP programs for the transmitter and receiver were loaded and executed by the system control program. Finally, the CPU program for PHS communication protocol was started. This concluded the system reconfiguration process in the prototype. For the 1.8-MB program examined here, reconfiguration took about 10 s. The components of the measured re-configuration time are shown in Fig. 18. This time is dominated by the time taken to install the FPGA software (about 8 s). The reason for this is that each FPGA has about one million gates, which must all be rewritten. After the platform switched to PHS mode, we established CS-PS communication to confirm that the communication sequence was successfully completed and that clear voice communication was possible. Figure 19 shows the measured average DSP loads of the PS prototype in each TDMA slot. Average DSP load is defined as the processing time normalized by the TDMA slot time. The CS prototype showed almost the same results. This confirms that all tasks were performed as designed and that DSP loads never became excessive. This means that real-time communication was successfully achieved. In addition, we note that 4-TDMA processing is possible by using the currently idle slots.

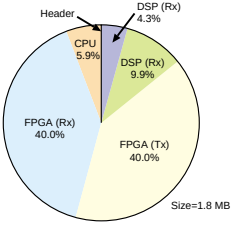

Fig. 16. PHS PS software allocation.

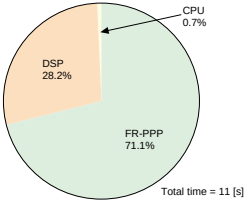

Fig. 18. Measured re-configuration time of PS.

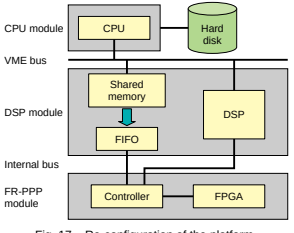

Fig. 17. Re-configuration of the platform.

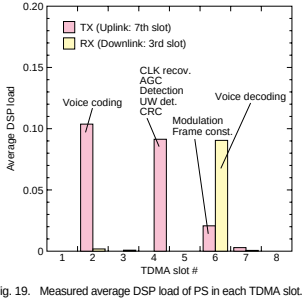

Fig. 19. Measured average DSP load of PS in each TDMA slot.

5. Conclusion

We made a wideband (>20 MHz) highly flexible SDR platform by developing flexible-rate pre-/post-processors (FR-PPPs). They make it possible to support wireless LAN systems that use the direct sequence spread spectrum (DSSS) scheme. We confirmed that the FR-PPP functioned as intended in both wireless LAN and PHS modes. Programs for both modes were written and tested in an experimental system. For the PHS mode, measured average DSP loads confirmed that stable full-duplex real-time communication was achieved. An over-the-air download protocol based on TCP/IP was designed and implemented on the prototype, and its operation was confirmed. The measured system reconfiguration was 11 s.

References

[1] N. Nakajima, R. Kohno, and S. Kubota, "Research and Developments of Software-Defined Radio Technologies in Japan," IEEE Commun. Mag., Vol. 39, No. 8, pp. 146-155, Aug. 2001.

[2] W.H.W. Tuttlebee, "Software Radio Technology: A European Perspective," IEEE Commun. Mag., Vol. 37, No. 2, pp. 118-123, Feb. 1999.

[3] C. Taylor, "A DSP System for a Software Radio Testbed," ACTS Mobile Commun. Summit'98, Rhodes, Greece, No. B1.2, June 1998.

[4] S. Ellingson and M. Fitz, "A Software Radio-Based System for Experimentation in Wireless Communications," Proc. IEEE VTC'98, Ottawa, Canada, Vol. 3, pp. 2348-2352, May 1998.

[5] H. Ishii, S. Kawamura, T. Suzuki, M. Kuroda, H. Hosoyama and H. Fujishima, "An Adaptive Receiver based on Software Defined Radio Techniques," Proc. IEEE PIMRC, San Diego, USA , Vol. 2, pp. 120-124, Sep. 2001.

[6] M. Mehta, N. Drew, and C. Niedermeier, "Reconfigurable Terminals: An Overview of Architectural Solutions," IEEE Commun. Mag., Vol. 39, No. 8, pp. 82-88, 2001.

[7] SDR Forum Technical Report, ver. 2.1, Nov. 1999.

[8] Y. Suzuki, K. Uehara, M. Nakatsugawa, Y. Shirato, and S. Kubota, "Software radio base and personal station prototypes," IEICE Trans. Commun., Vol. E83-B, No. 6, pp. 1261-1268, June 2000.

[9] intersil, "HSP50214B Data sheet," May 2000.

[10] intersil, "HSP50215 Data sheet," Jan. 1999.

[11] Texas Instruments, "GC4016 Data sheet," Aug. 2001.

[12] Texas Instruments, "GC4116 Data sheet," Apr. 2001.

[13] Y. Shirato, M. Nakatsugawa, H. Yoshioka, M. Matsui, S. Kubota, and M. Umehira, "Software Defined Radio Prototype for PHS and Wireless LAN Systems (II) –Signal Processing Platform Using Flexible Rate Pre/Post Processor–," IEICE Technical Report SR01-014, pp. 47-53, 2001.

[14] RCR Standard, "Personal Handy Phone System," ARIB STD-28, ver. 1, rev. 1, 1995.

[15] ISO/IEC 8802-11, ANSI/IEEE Std 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications."

[16] T. Shono and M. Matsui, "Software Defined Radio Prototype (II) –Implementation and Evaluation of IEEE 802.11 Wireless LAN," NTT Technical Review, Vol. 1, No. 4, pp. 24-30, 2003.

[17] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, "Camellia: a 128-bit bock cipher suitable for multiple platforms–design and analysis–," in Proc. Workshop on Selected Areas in Cryptography, SAC2000, Waterloo, Canada, Aug. 2000.

Hiroyuki Shiba

Wireless Systems Innovation Laboratory, NTT Network Innovation Laboratories.

He received the B.E. and M.E. degrees from Gunma University, Gunma, in 1995 and 1997, respectively. In 1997, he joined NTT Wireless Systems Laboratories, where he has been engaged in research of data communication and software defined radio systems. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE), Japan. He received the Young Investigators Award from IEICE in 2001 and the 18th Telecom System Technology Award from the Telecommunications Advancement Foundation.

Yushi Shirato

Research Engineer, Wireless Access Systems Project, NTT Access Network Service Systems Laboratories.

He received the B.E. and M.E. degrees in electrical engineering from the Science University of Tokyo, in 1990, and 1992, respectively. In 1992, he joined NTT Wireless Systems Laboratories. Since then, he has been engaged in research and development of digital signal processing techniques on adaptive equalizers and automatic gain controlled amplifiers, and diversity combining. He has also worked on software defined radio. His current interest is modems for fixed wireless access systems. He is a member of IEICE. He received the Young Investigators Award from IEICE in 2000 and the 18th Telecom System Technology Award from the Telecommunications Advancement Foundation.

Hiroshi Yoshioka

Wireless Access Project, NTT Access Service Systems Laboratories.

He received the B.E. and M.E. degrees in computer engineering from Niigata University, Niigata, in 1996 and 1998, respectively. In 1998, he joined NTT Wireless Systems Laboratories. In 1999, he joined the Network Innovation Laboratories. He has been engaged in research of wireless data communication and software radio systems. In 2003, he joined the Wireless Access Project, NTT Access Service Systems Laboratories. His research focuses on fixed wireless access systems. He is a member of IEICE.

Ichihiko Toyoda

Senior Research Engineer, Wireless Systems Innovation Laboratory, NTT Network Innovation Laboratories.

He received the B.E., M.E., and Dr.Eng. degrees in communication engineering from Osaka University, Osaka, in 1985, 1987, and 1990, respectively. In 1990, he joined NTT Radio Communication Systems Laboratories, Kanagawa, Japan, where he was engaged in developmental research based on electromagnetic analysis for three-dimensional (3-D) and uniplanar MMICs. From 1994 to 1996, he was with NTT Electronics Technology Corporation, Kanagawa, Japan, where he was engaged in development of wireless communication equipment and MMICs. From 1996 to 1997, he was with NTT Wireless Systems Laboratories, Kanagawa, Japan, where he did research and development on highly integrated multifunctional MMICs, high-frequency Si MMICs, and MMIC design software based on the 3-D masterslice MMIC technology. From 1997 to 2001, he was engaged in development of ultra-high-speed digital ICs and MMICs for optical and wireless communication systems as an Engineering Director of IC Design at NTT Electronics Corporation, Kanagawa, Japan. His current interests concern high-speed wireless access systems and software defined radio and their applications. He is a member of IEICE and IEEE. He received the 1993 Young Investigators Award presented by the IEICE, the Japan Microwave Prize presented at the 1994 Asia-Pacific Microwave Conference in Tokyo, and the 18th Telecom System Technology Award from the Telecommunications Advancement Foundation.