EVALUATION OF SOFTWARE DEFINED RADIO PROTOTYPE FOR PHS AND WIRELESS LAN

Hiroyuki SHIBA (NTT, Yokosuka-shi, Kanagawa, Japan, shiba@wslab.ntt.co.jp);
Yushi SHIRATO (NTT, Yokosuka-shi, Kanagawa, Japan, yushi@wslab.ntt.co.jp);
Takashi SHONO (NTT, Yokosuka-shi, Kanagawa, Japan, shono@wslab.ntt.co.jp);
Hiroshi YOSHIOKA (NTT, Yokosuka-shi, Kanagawa, Japan,
yoshioka@wslab.ntt.co.jp); Ichihiko TOYODA (NTT, Yokosuka-shi, Kanagawa, Japan,
toyoda@wslab.ntt.co.jp); Kazuhiro UEHARA (NTT, Yokosuka-shi, Kanagawa, Japan,
uehara@wslab.ntt.co.jp);and
Masahiro UMEHIRA (NTT, Yokosuka-shi, Kanagawa, Japan, umehira@wslab.ntt.co.jp)

ABSTRACT

A software defined radio (SDR) prototype based on a multiprocessor architecture (MPA) is developed. Software for the Japanese personal handy phone system (PHS, a 2G mobile system), and IEEE802.11 wireless LAN, which has much wider bandwidth than 2G systems, is successfully implemented. The prototype uses the newly developed flexible-rate pre-/post-processor (FR-PPP) to achieves the flexibility and wideband performance needed. This paper shows the design of the SDR prototype and evaluates its performance by experiments that evaluate PHS processor load and wireless LAN throughput characteristics and processor load.

1. INTRODUCTION

The rapid improvements seen in the performance of programmable signal processing devices such as DSPs and FPGAs have made software defined radio (SDR) possible. An SDR infrastructure can realize various wireless communication systems just by changing the software [1]-[4]. SDR architecture and program download schemes have been discussed in the SDR forum and conferences [5]-[7]. The FCC adopted rule changes to accommodate the authorization and deployment of SDR in September 2001. The new rules allow manufacturers and operators to reconfigure devices after they have been deployed in the field.

In 2000, we reported an SDR prototype that supported only narrow bandwidth systems (several hundred's of kHz) such as the Personal Digital Communication (PDC) System and the Japanese personal handyphone system (PHS) [8]. This paper describes an advanced SDR prototype that uses newly developed flexible-rate pre-/post-processor (FR-PPP) to offer improved bandwidth, more than 20 MHz, and improved 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 wireless LAN (IEEE802.11) operation. The prototype also implements the over-the-air (OTA) download function.

This paper is organized as follows: section 2 describes the hardware architecture of the platform. The software architecture and the actual program design method are shown in Section 3, and the validity of the design method is confirmed by experimental results in Section 4. Section 5 concludes the paper.

2. HARDWARE ARCHITECTURE

The prototype consists of three stages: radio frequency (RF), intermediate frequency (IF), and baseband (BB) stages. The RF and IF stages consist of multiband analog circuits. A/D conversion is performed at the IF. The digital IF and BB stages consist only of programmable devices. Fig. 1 shows a block diagram of the platform. It uses the multiprocessor architecture (MPA) and consists of four DSP's, a CPU, and three pre-/post-processors (PPP's). A 64-bit VME bus connects the DSP, CPU, and external interface modules. The platform has three independent RF/IF branches for future adoption of simultaneous multimode operation and/or smart antennas.

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 wireless LAN operation, the CPU module provides an Ethernet interface at both the access point (AP) and station (STA) platforms. Fig. 2 shows the appearance of the prototype.

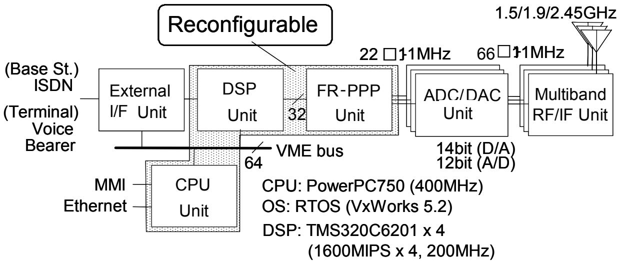


Fig. 1: Block diagram of prototype.

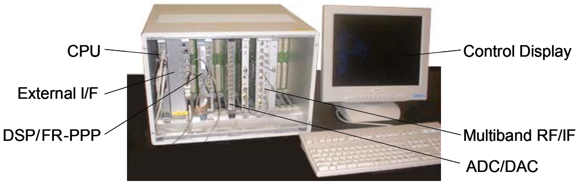


Fig. 2: Appearance of the prototype.

The RF signal received by the antenna is down-converted by analog circuits to an IF signal, and then analog-to-digital 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 realized with a single amplifier by switching bandpass filters, to meet the PDC, PHS, and IEEE802.11 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 ADC with 12-bit resolution and 88 MSPS sampling rate. The AGC circuit is set before the ADC. D/A conversion is performed using a DAC with 14-bit resolution after up-sampling the BB signal with center frequency of 22MHz and bandwidth of 22MHz. The imaging of the center frequency of 66 MHz whose bandwidth is 22 MHz is used as the IF signal.

The IF has too high a clock-rate to be processed by BB processors, so pre-/post-processors (PPP's) are used to realize the high-speed real-time digital processes required; these include filtering, waveform-shaping, and spectrum de-spreading.

The newly proposed flexible-rate pre-/post-processor (FR-PPP) consists of field programmable gate arrays (FPGA's) and direct digital synthesizer (DDS). Each branch uses two 1 MGate FPGA's for pre- and post-processing. Fig. 3 shows the configurations of the receiving blocks of a conventional PPP and the FR-PPP. Conventional PPP's (commercially available digital up/down converter) 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-PPP is much smaller because the FPGA can flexibly act as the filters needed for each system. In addition, while conventional PPP's 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 clock-rates demanded in an arbitrary manner. This also reduces the circuit scale and offers high-speed operation. These breakthroughs realize a small circuit scale (75% smaller), a wide-bandwidth (more than 20 MHz) and very flexible SDR that can support wireless LANs as well as 2G systems such as PHS. Figures 4 and 5 show detailed configurations of FR-PPP for PHS mode and wireless LAN mode, respectively.

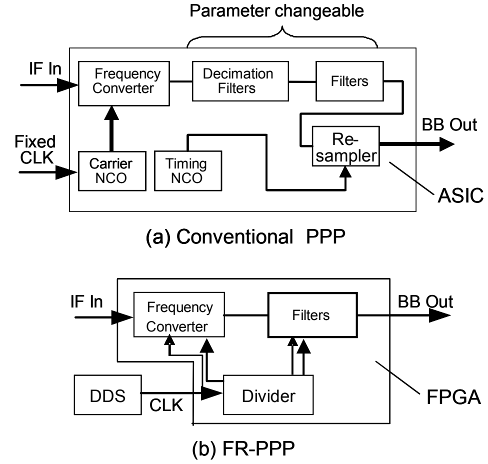
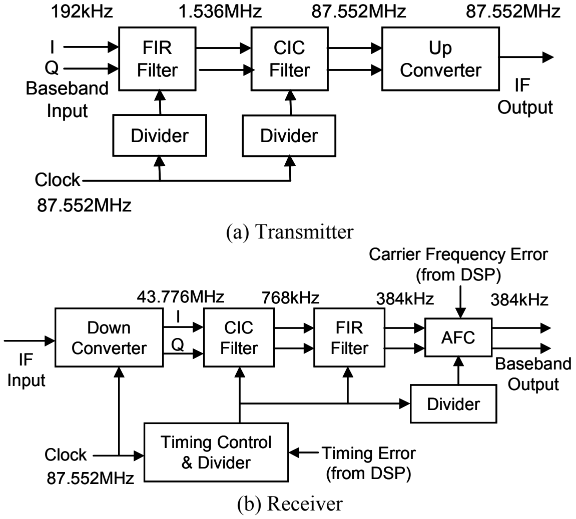


Fig. 3: Configurations of conventional PPP and FR-PPP (receiving blocks).

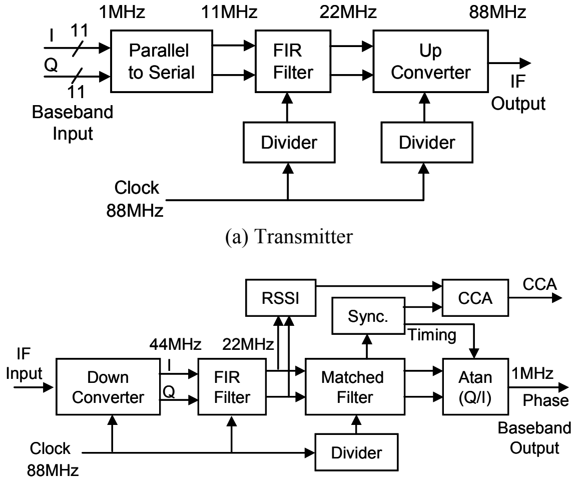


(a) Transmitter

(b) Receiver

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

The prototype uses a Central Processing Unit (CPU) and digital signal processors (DSP's) to perform BB processing and control. The CPU, a 400 MHz PowerPC, handles high-layer protocols including PHS call control and medium access control (MAC) for the wireless LAN. Physical layer processes such as modulation and demodulation, voice coding and decoding etc., are handled by the DSP's. Each DSP offers the computational power of 1,600 MIPS, and transmitting and receiving processes are each performed by one DSP. Table 1 summarizes the function assignment to the processors.



(a) Transmitter

(b) Receiver

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

Table 1: Function assignment to processors.

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

3. SOFTWARE ARCHITECTURE

The SDR prototype operates as the terminal of a specific wireless system after loading the software of the system into its processors. The programs written for the prototype reproduce almost all of the key operating functions of PHS [13] and the direct-sequence spread spectrum (DSSS) scheme of IEEE 802.11 wireless LAN [14] (infrastructure mode) offered by regular commercial terminals. Table 2 shows the major parameters of both systems.

The software architecture of SDR's has been discussed in the SDR forum [5]. Architecture selection strongly depends on the OS, API, and protocol stack, and impacts 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.)

Fig. 6 shows the program component architecture of the prototype. A system control program, an over-the-air (OTA) download program, and communication control programs run on the real time OS (RTOS). 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 programs, and FR-PPP configuration data containing a wireless system protocol. The CPU and DSP programs were written in the C programming language while the FPGA configuration data of the FR-PPP was written in the Verilog hardware definition language (HDL).

Table 2: Major parameters of wireless systems implemented.

	PHS•RCR STD-28•	WLAN•IEEE802.11•
Frequency	1.9GHz	2.45GHz
Access method	TDMA-TDD	CSMA-CA
Modulation	•/4-QPSK	DSSS
Air bit-rate	384kbit/s	11Mchip/s
User data-rate	32kbit/s	1Mbit/s, 2Mbit/s
Omitted functions	Mobility management, Authentication, Billing, etc	Power management, etc

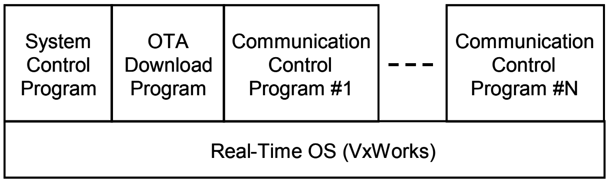


Fig. 6: Program component architecture of the prototype.

3.1. 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 RTOS 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 3 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 the CPU program interrupts the DSP program, which is synchronized to TDMA slot timing. The DSP program reads the shared memory to obtain the API command and parameters. After the DSP program executes the task, the DSP program writes the result into the shared memory and interrupts the CPU program. 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.

Table 3: Example of DSP task scheduling.

TDMA slot#	TX	RX
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		

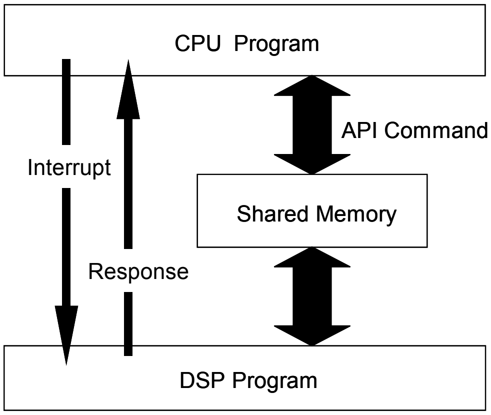


Fig. 7: API between CPU and DSP programs.

3.2. 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. With a few exceptions, the spectrum despreading process is handled by the FR-PPP because it requires real-time correlation detection given the very high-speed signals; the chip-rate is 11 Mchip/s. The load of the spectrum spreading process is small, so it can normally be handled by the DSP. In addition, the 32-bit frame check sequence (FCS) process is assigned to the DSP because it consists of product-sum operations which are very suitable for DSP execution. The API between the CPU and DSP programs is the same as that of PHS, but the interrupt timing is based on MAC layer packet data units (MPDU's).

Since the MPA of the platform uses a VME bus to link the CPU and DSPs, there is a significant processing delay. For instance, the data transfer time was estimated to be about 40μs, and the interrupt response time to be about 3μs [15]. They make it difficult for the CSMA-CA protocol to realize μs-order response times. IEEE 802.11 specifies the short interframe space (SIFS) of 10μs [14]. Therefore, we parameterized the IFS times as follows in order to evaluate system response performance•

$SIFS' = M \times SIFS$,

$DIFS' = M \times DIFS = M \times (SIFS + 2 \times Slot\text{-}Time)$,

where, M is defined as the IFS factor.

Fig.8 shows the relation between IFS factor and throughput under the condition that packet error and congestion are negligible. The maximum transmission rate of MPDU is defined as throughput in Fig. 8. When M is 10, throughput becomes about 60-70 %.

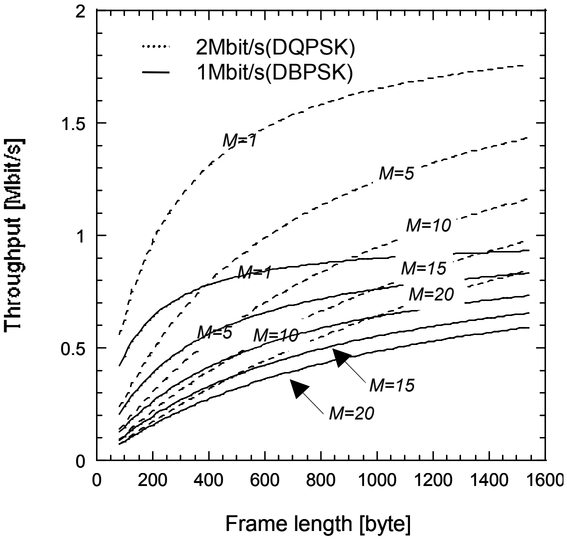


Fig. 8: The relation between Throughput and IFS factor.

3.3. Over-the-Air Download Program

The OTA download program is designed to be very general, compact, and secure. The OTA download protocol works over TCP/IP; even if the communication mode is changed, the OTA download protocol is not affected by the lower layer protocols. The OTA download protocol not only downloads the software, but also authenticates and encrypts the data using SSL (Secure Socket Layer). To ensure the security of downloads, the 128-bit next-generation block cipher "Camellia," which was co-developed by NTT and Mitsubishi Electric Corporation, was implemented as the chipper algorithm of SSL.

4. PERFORMANCE EVALUATION

Fig. 9 shows the experimental setup used. The program sizes of the non-compressed packaged PHS and wireless LAN communication control programs were about 2 MBytes and 3 MBytes, respectively.

First, by loading the wireless LAN program into the platforms and establishing communication between the STA and AP, we confirmed successful data communication between two PC's connected to the two SDRs. The spectrum at the FR-PPP transmitter output for wireless LAN mode is shown in Fig. 10. It was confirmed that the measured spectrum satisfied the transmit spectrum mask provided in IEEE 802.11 (11MHz < $|f\text{-}f_c|$ < 22MHz; less than -30dBr, $|f\text{-}f_c|$ > 22MHz; less than -50dBr) [14]. Fig. 11 shows measured average DSP loads of each symbol in wireless LAN mode. The ratio of the measured average DSP loads for each transmitting and receiving signal process, such as CRC, modulation, descrambling, and demodulation, is also shown. Two DSP chips are used, one for transmission, the other for reception. The measured average DSP load was about 60%, which confirms that signal processing in the DSP was executed without dropping. Fig. 12 shows the throughput characteristics for the 1 Mbit/s and 2 Mbit/s modes. Measured data agreed very well with the simulated data. This confirmed that the programs were performing as intended. The measured minimum IFS factor M was 10. This value can be further minimized by using on-board or on-chip processors, for instance.

Next, the PHS PS communication control program was download from the OTA download server by the OTA download function. It was confirmed that the PS software of PHS was successfully downloaded and software integrity was retained. After the OTA download finished successfully, prototype reconfiguration was executed by the system control program automatically. The reconfiguration time to install the PS software of PHS was about 10 seconds. This time is dominated by the time taken to configure the FR-PPP, about 8 seconds. The reason for this is that all configuration data of both FPGAs (1 MGate for transmitting and 1 MGate for receiving) must be rewritten.

After the platform was changed to PHS mode, communication between the CS and PS was established to confirm that the communication sequence was successfully realized and also that clear voice communication was possible. The constellation for PHS mode whose modulation is π/4-QPSK is shown in Fig. 13. The measured error vector magnitude was 1.9%. Fig. 14 shows measured average DSP loads for each TDMA slot of the PS. The average DSP load was defined as the processing time normalized by the TDMA slot time. The CS showed almost the same results. The results confirm that all tasks were performed as designed and that DSP load never exceeded one. 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.

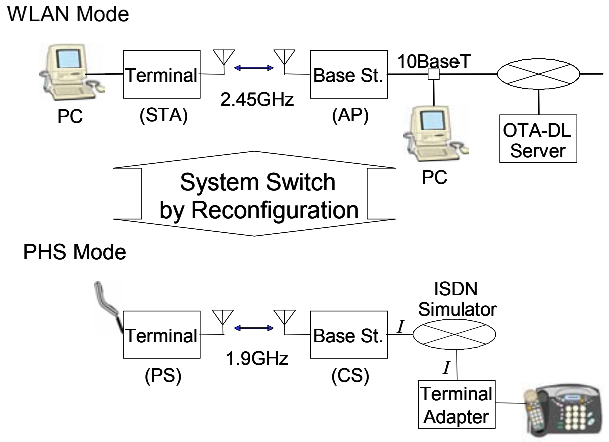


Fig. 9: Experimental setup.

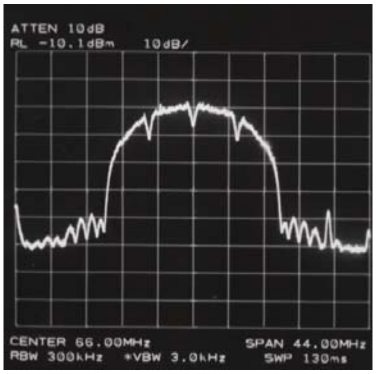


Fig.10: The transmitter spectrum for wireless LAN mode.

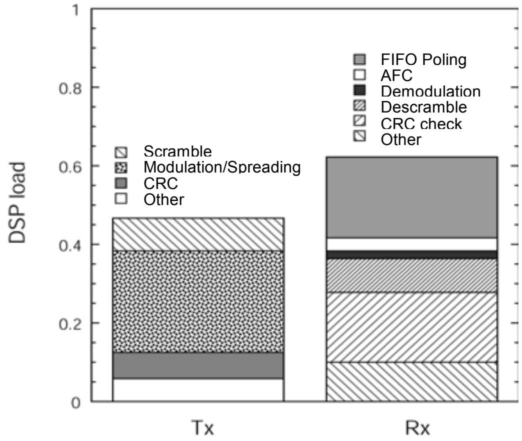


Fig. 11: Measured average DSP loads of each symbol for wireless LAN mode.

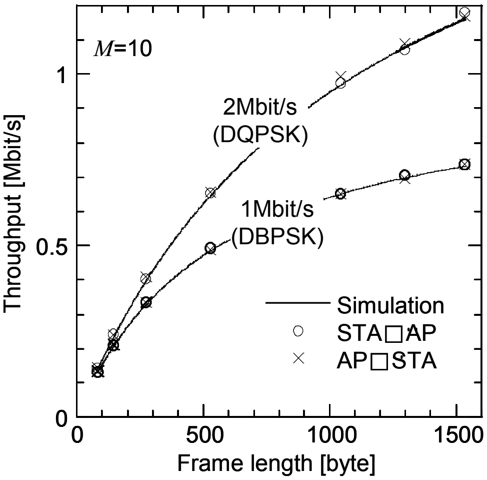


Fig. 12: Throughput characteristics in wireless LAN mode.

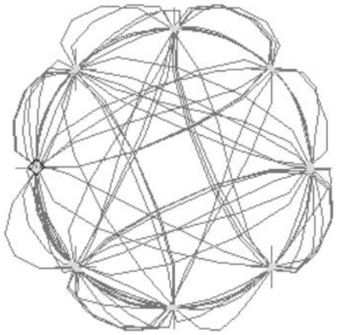


Fig. 13: The constellation for PHS mode.

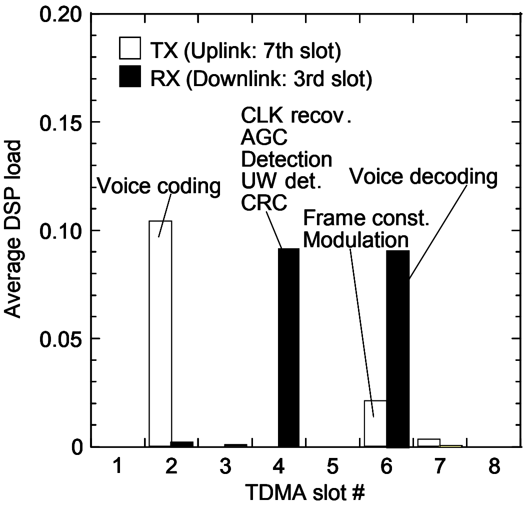


Fig. 14: Measured average DSP load of personal station in each TDMA slot for PHS mode. (The average DSP load is defined as the processing time normalized by the TDMA slot time.)

5. CONCLUSION

We realized a wideband (more than 20 MHz) and highly-flexible SDR platform by developing flexible-rate pre-/post-processors (FR-PPP's). They make it possible to support wireless LAN systems that use the direct sequence spread spectrum (DSSS) scheme. It was confirmed that the FR-PPP functioned as intended in both PHS mode and wireless LAN mode. Programs for PHS and the IEEE802.11 wireless LAN were written and tested in an experimental system. For the PHS mode, measured average DSP loads confirmed the achievement of stable full-duplex real-time communication. For the wireless LAN mode, throughput measurements and the measured average DSP loads confirmed that the prototype operated as designed. An over-the-air download protocol based on TCP/IP was designed and implemented on the prototype, and its operation was confirmed. Moreover, system reconfiguration was executed; the measured reconfiguration time was about 10 seconds.

10. REFERENCES

[1] J. Mitola, “The software radio architecture,” IEEE Commun. Mag., vol. 33, no. 5, pp. 26-38, May 1995.

[2] J. Mitola,′′Software Radio Architecture: A Mathematical Perspective,′′IEEE J. Sel. Areas Commun., vol. 17, no. 4, pp. 514-538, April 1999.

[3] P. W. Fox, ′′Software Defined Radios – Motorola’s Wireless Information Transfer System(WITS), ′′ EUROCOMM, pp. 43-46, 2000.

[4] N. Nakajima, R. Kohno, and S. Kubota, “Research and Developments of Software-Defined Radio Technologies in Japan,” IEEE Commun. Mag., pp. 146-155, August 2001.

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

[6] M. Cummings and S. Heath, “Mode switching and software download for software defined radio: The SDR Forum Approach,” IEEE Commun. Mag., vol. 37, no. 8, pp. 104-106, Aug. 1999.

[7] H. Shiba, T. Shono, K. Uehara, and S. Kubota, “A Study on an Over-the-Air Download Scheme for Software Radio,” Proc. 7th Int. Workshop on MoMuC, pp. P-3-1-P-3-6, Oct. 2000.

[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,” January 1999.

[11] TEXAS INSTRUMENTS, “GC4016 Data sheet,” Augest 2001.

[12] TEXAS INSTRUMENTS, “GC4116 Data sheet,” April 2001.

[13] RCR Standard, “Personal Handy Phone System,” ARIB STD-28, ver. 1, rev. 1, 1995.

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

[15] K. Uehara, H. Tanaka, H. Shiba, Y. Suzuki, M. Nakatsugawa, Y. Shirato, and S. Kubota, “Design of software radio for cellular communication systems and wireless LANs,” in Proc. PIMRC 2000, vol. 1, pp. 474-478, Sep. 2000.