AN IMPLEMENTATION OF NOVEL MULTI-BAND/MODE SDR PLATFORM FOR SIMULTANEOUS MULTIPLE RADIO COMMUNICATION SYSTEM

Kosuke YAMAZAKI, Toru KITAYABU, Yuji IKEDA, Issei KANNO and Hiroyasu ISHIKAWA
KDDI R&D Laboratories Inc., 2-1-15 OHARA, FUJIMINO-SHI, SAITAMA, JAPAN
{ko-yamazaki, to-kitayabu, yj-ikeda, is-kanno, ishikawa}@kddilabs.jp

ABSTRACT

Toward the actualization of flexible multi-band/mode handheld terminals, we have newly developed a Software Defined Radio (SDR) platform in order to verify its low power consumption feature. The developed multi-mode/band SDR platform has only one single baseband processor and is equipped with newly developed control software which handles the limited computational power effectively. As a result, the developed prototype SDR platform can handle multiple radio communication systems (RCS) simultaneously even in the limited computational resource availability.

1. INTRODUCTION

Recently, various Radio Communication systems (RCS) have been developed. Users can use a suitable RCS depending on their own environment and receive great benefit from them. Against such a background, a multi-band/mode terminal which is equipped with various RCS is attracting great attention. In order to use each RCS effectively, it is very important to monitor and evaluate each signal quality in advance[11]. Therefore, simultaneous multiple RCS operation becomes a significant function for a terminal to avoid disconnection while monitoring. However, high power consumption makes it difficult for a terminal to perform these operations.
To overcome this problem, we chose the Software Defined Radio (SDR) approach. RCS implemented by the SDR technique can be replaced easily without any hardware change and then there is no need to add new hardware for additional RCS. However, because the required computational power for SDR software becomes very high, a solution is required to use limited computational power. Our prototype uses only one baseband processor that can execute SDR software. Various RCS can run on this processor by software. Naturally, its computational power is limited and the newly developed control software can handle this effectively.
To confirm SDR feasibility, we have already developed multiple SDR prototypes. In the first prototype, CDMA2000 1xEV-DO [4] was implemented on a low-power consumption baseband processor, SB3010 SoC produced by Sandbridge Inc. [8]. We achieved air connection between this prototype and a commercial base station [1, 2, 3]. In the second prototype, EV-DO software was ported to another baseband processor, S5530 SoC produced by Stretch Inc. On this porting, we developed an automatic source code optimization tool, SCOT (Source Code Optimization Tool) [10]. This tool treats tags embedded in the source code and outputs optimized source code for each baseband processor. Developed EV-DO software for SB3010 has been translated for S5530 using SCOT at very low cost. In the third prototype, EV-DO software was ported to another baseband processor, PC102 produced by picoChip [9]. The PC102 processor has higher computational power. In this prototype, IEEE802.16D was implemented too. Even if simultaneous operation is not implemented, it is confirmed that different SDR software can run on the same platform by running software [7]. However, one of the main drawbacks of this prototype is higher power consumption and it is very difficult for the terminal to adopt it. Through these activities, the feasibility of the SDR technique has been validated.
Now, we have developed a novel SDR platform, called the “SDR core module”, whose baseband processor is SB3500 SoC, as the next version SoC of SB3010. The newly developed control software (CTRL S/W) on this module realizes many functions related to power saving. The RCS selection function is a very important one that should be performed based on the monitored signal quality. This real-time monitoring can be realized only by simultaneous multiple RCS operation. In order to realize it on a single baseband processor, we focus on the relationship between processing load and expected function. The amount of signal processing load for monitoring is much less than that for handling traffic data. Therefore, the monitoring function can be pooled in the remaining small computational resources even while the other RCS is running.
The rest of this paper is organized as follows. In Section 2, we present details of our newly-developed SDR platform, the “SDR core module”. In Section 3, we discuss the

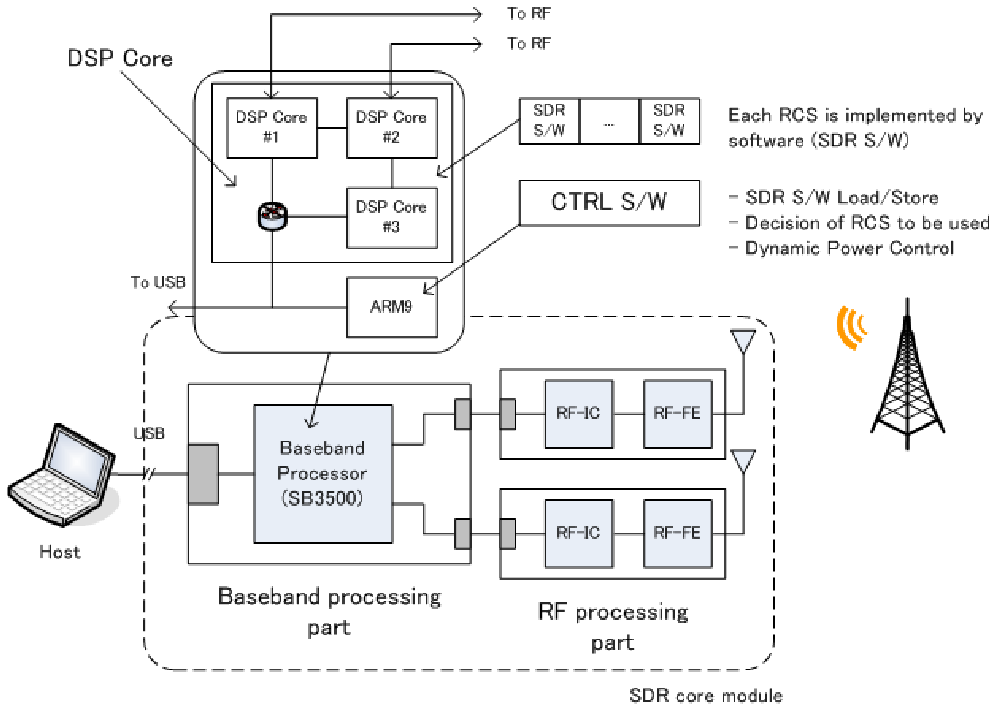


Figure 1: An overview of SDR core module. This module can be divided into two parts, the "Baseband processing part" and "RF processing part". A single DSP, SB3500 produced by Sandbridge Inc., is applied to the baseband processing part. This processor consists of three DSP cores and one ARM core. Control software implemented on the ARM core has the responsibility to load/store each SDR software, decide the RCS used, Dynamic Power Control etc. The SDR software implemented on the DSP core has the responsibility to perform baseband processing for each RCS.

feasibility of simultaneous multiple RCS operation under the limited computational resources based on our implementation results. In Section 4, we conclude this paper and discuss future works.

2. SDR CORE MODULE

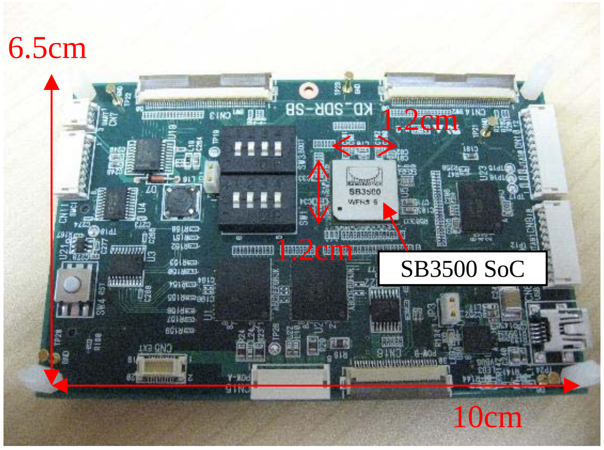
We developed a new SDR platform called the "SDR core module". This module is connected with host computers and behaves as a modem. The SDR core module always monitors the current environment and selects the best RCS. Figure 1 shows an overview of the SDR core module.

The SDR core module consists of a "baseband processing part" and two "RF processing parts." The baseband processor for the SDR core module is SB3500 SoC produced by Sandbridge Inc., which consists of three DSP (digital signal processing) cores that have four threads per core. The DSP core runs at 600 MHz and each thread on it runs at 150 MHz. The software for SB3500 can be developed by the ANSI C standard and its portability is very high. In fact, our EV-DO software is written in the ANSI C and has been ported to SB3500 at low cost. As a general purpose processor, the ARM 926EJ-S core is applied. Some control procedures run on it. An integrated device power management unit can perform flexible power control that ups/slows the clock frequency of the DSP and ARM core according to the processing load for each core. An abstract of SB3500 is summarized in Table 1. A photograph of the baseband processing part is shown in Figure 2.

Table1: SB3500 features.

DSP core architecture	Three DSP cores that have four hardware threads each
DSP core clock frequency	600 MHz
CPU	ARM926EJ-S
Option	ANSI C available
	Integrated Device Power Management Unit

The RF signal comes from the RF processing part, which is converted to a digital I/Q baseband signal. These signals go into the DSP core of SB3500 to be processed by SDR S/W on the DSP core and passed to the USB interface via the ARM core.
SDR software (SDR S/W) implemented on the DSP core performs signal processing of RCS. Most of the signal processing is performed on the DSP core and some management processing is performed on the ARM core.



These two types of core collaborate to perform each RCS.
Figure 2: The baseband processing part of the SDR core module. SB3500 SoC is located in the center of the board. The module size is 65mm * 100mm.

3. SOFTWARE OPTIMIZATION

In our implementation, two RCS are selected for implementation, CDMA2000 1xEV-DO (EV-DO) and WiMAX wave2 (WiMAX). When an approach other than SDR is adopted, two baseband chips have to be implemented; however, our prototype can realize this type of dual-mode service with only a single baseband processor. This leads to low power consumption and lower hardware cost. In addition, if another new RCS, such as LTE (Long Term Evolution), is released, we can just install it using software. In this section, we show the implementation results of EV-DO and WiMAX and a feasibility study of simultaneous multiple RCS operation is shown.

3.1. CDMA2000 1xEV-DO

SDR S/W of CDMA2000 1xEV-DO meets the specifications published by 3GPP2 (Third Generation Partnership Project 2) [6]. EV-DO is an FDD system and its center frequency is 875MHz for downlink and 830MHz for uplink, its bandwidth is 1.48MHz. In the EV-DO system, users have to synchronize with a base station at first. This is performed by the PN despreading function. After that, various configuration parameters are set up between the terminal and the base station and data traffic goes through the flow according to these parameters. We have implemented this flow, measured the required cycle counts and evaluated the required computational resources. To realize EV-DO real-time operation, one slot data arriving each terminal has to be processed within one time slot (1.6ms). Or each data is overwritten by the next data. Table 2 shows the implementation results on the SDR core module.

Table 2: The EV-DO implementation results on the SDR core module. These results are measured on our hardware and some unpredictable redundant cycles are already included.

Procedures	# of cycles	Time(ms)
Initial acquisition	296,648,070	2,373.1846
Track acquisition	30,683	0.2455
Filter calculation	178,651	1.4292
Rx (Forward Ch.)	385,281	3.0822
Tx(Reverse Ch.)	1,013,379	8.1070
Tx(Access Ch.)	3,172,946	25.3836

All of these procedures listed on Table 2 have been implemented on the DSP core in SB3500 SoC. “Initial acquisition” is performed to synchronize with the base station before any procedure of EV-DO begins and then it is acceptable that its processing time exceeds 1.6 ms. In fact, even though the execution time vastly exceeds this as shown in Table 2, the QPSK constellation has been found clearly as shown in Figure 3, where the initial acquisition process has been completed on our prototype..

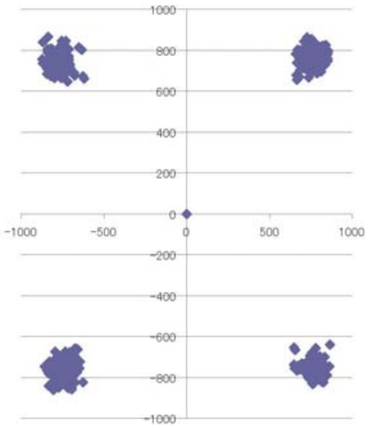


Figure 3: EV-DO received constellation.

Therefore, if the required task for EV-DO is only initial or track acquisition to monitor signal quality like CINR (Carrier to Interference Noise Ratio), we can use a much longer time than 1.6ms and then the required computational resources can be reduced. These reduced computational resources can be allocated for WiMAX processing.

3.2. WiMAX Wave2

The implemented WiMAX meets the WiMAX forum mobile system profile wave1 and wave2 [5, 6]. In our implementation, the bandwidth is 10 MHz and the number of FFT points is 1024. As the modulation scheme, 16 QAM and QPSK have been implemented. Tx spectrum from our prototype is shown in Figure 4.

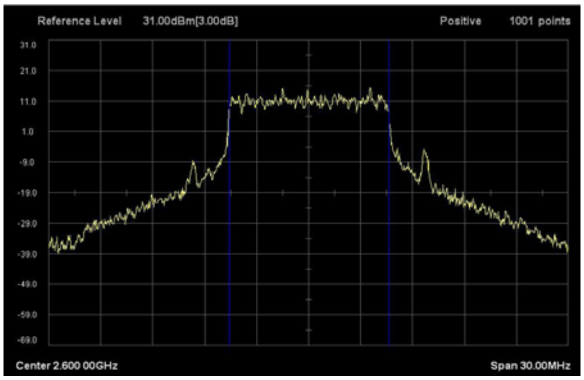


Figure 4. WiMAX Tx spectrum. The center frequency is 2.6GHz and OBW is 9.21MHz.

In WiMAX, users have to synchronize with the base station using the preamble. Next, some parameters for downlink are found in the DCD (Downlink Channel Descriptor) message and those for uplink are found in the UCD (Uplink Channel Descriptor) message. After getting these parameters, the ranging scheme starts to negotiate the bandwidth, MAC address, connection ID and so forth. Basic information exchanges for PHY, authentication, and the registration scheme follow. After all processes are complete, the IP connection can be setup. The implementation results on the PHY signal processing are shown in Table 3.

Symbol processing includes Receive I/Q filtering and decimation, Frequency offset removal, FFT and Cyclic Prefix (CP) removal. The input to symbol processing is baseband I/Q samples for one complete OFDMA symbol. Preamble processing includes preamble acquisition, frame boundary detection, symbol boundary detection and frequency offset estimation. But, this has to be performed before all of the processing starts and then the lower one in Table 3 does not include it. The channel decoder includes burst processing and block processing. The former means a regular data decoder and the latter means a CTC/CC decoder. As shown in Table 3, only one computational resource is required for symbol processing, slot processing,

Table 3: An example of simultaneous RCS operation. The upper table shows the case when all traffic flows through EV-DO connection and the signal quality of WiMAX is monitored. Just one thread is required and then EV-DO can use the remaining threads. The lower table shows the opposite case when the signal quality of EV-DO is monitored and all the traffic flows through the WiMAX connection.

Core #	Thread #	EV-DO : Data Transfer, WiMAX : Monitoring DL	
1	1	Main Routine	
	2	WiMAX Preamble Acquisition	
	3		EV-DO TxDMA & RCH & ACH
	4		
	SBX Mem	Tx, Thread#3,#4 stack -> 130KByte	
2	1		EV-DO Turno Dec.
	2		EV-DO Turno Dec.
	3		EV-DO Turno Dec.
	4		EV-DO Turno Dec.
	SBX Mem	Turbo Decode, Thread#1,#2,#3,#4 stack -> 210KByte	
3	1	EV-DO RxDMA & Initial Acquisition	EV-DO RxDMA & Track Acquisition
	2	EV-DO Initial Acquisition	EV-DO Filter Calc.
	3	EV-DO Initial Acquisition	EV-DO FCH
	4	EV-DO Initial Acquisition	EV-DO FCH
	SBX Mem	Initial acquisition, tracking acquisition, adaptive filter, Rx processing, Thread#0,#1,#2,#3 stack -> 256KByte	
ARM		CTRL-S/W, EV-DO MAC, WiMAX Dummy MAC	

Core #	Thread #	EV-DO : Monitoring DL, WiMAX : Data Transfer		
1	1	Rx symbol processing	Tx symbol processing	Rx symbol processing
	2	Rx Slot Group processing	TX Block Processing - 1	Rx Slot Group processing
	3	Rx Demodulator, scrambler	TX Block Processing - 2	Rx Demodulator, scrambler
	4	RX Burst Processor, RX Interleaver, Partial RX	TX Block Processing - 3	RX Burst Processor, RX Interleaver, Partial RX
	SBX Mem			
2	1	Channel Decoder(Viterbi, Turbo) - 1	Channel Encoder - 1 AES - 1	Channel Decoder(Viterbi, Turbo) - 1
	2	Channel Decoder(Viterbi, Turbo) - 2	Channel Encoder - 2 AES - 2	Channel Decoder(Viterbi, Turbo) - 2
	3	Channel Decoder(Viterbi, Turbo) - 3	Channel Encoder - 3 AES - 3	Channel Decoder(Viterbi, Turbo) - 3
	4	Channel Decoder(Viterbi, Turbo) - 4	Channel Encoder - 4 AES - 4	Channel Decoder(Viterbi, Turbo) - 4
	SBX Mem			
3	1	EV-DO RxDMA & Initial Acquisition	EV-DO RxDMA & Track Acquisition	
	2	EV-DO Initial Acquisition		
	3	EV-DO Initial Acquisition		
	4	EV-DO Initial Acquisition		
	SBX Mem	Initial acquisition, tracking acquisition, adaptive filter, Rx processing, Thread#0,#1,#2,#3 stack -> 256KByte		
ARM		CTRL-S/W, WiMAX MAC Mgnt, WiMAX MAC, a part of EV-DO MAC		

demodulator and scrambler and burst processing. And the CTC/CC decoder requires four threads. Based on this result, it is clear that WiMAX's full function can be implemented in SB3500 SoC.

3.3. Simultaneous operation of EV-DO and WiMAX

The computational resource required for EV-DO and WiMAX is analyzed in previous sub-sections. Our main target is the simultaneous multiple RCS operation, EV-DO and WiMAX. The amount of required computational resources for EV-DO is two cores and two threads. That for WiMAX is two cores. The available amount is just three cores and then it is impossible to implement both of them simultaneously. However, in both cases, some computational resources still remain, two threads are available in EV-DO and four threads (one core) are available in WiMAX, which is sufficient to perform each monitoring function.

In Table 3, core/thread assignment is shown for the simultaneous multiple RCS operation. The upper one shows the case where EV-DO runs with full function and WiMAX runs with only monitoring function. The lower one shows the opposite case. Here, the monitoring function means the measurement of CINR (Carrier to Interference and Noise Ratio) value.

The CINR for EV-DO can be calculated only after the initial acquisition. Therefore, the amount of computational resources to calculate CINR can be reduced as small as required as mentioned in 3.1. The CINR for WiMAX is reported from WiMAX BS in the periodical message, DCD message. To receive this message, preamble synchronization, DL-MAP decoding, and burst processing are required. Among them, the time-sensitive function is the preamble synchronization only. To measure CINR alone, other functions can be operated for a longer time. And then the amount of required computational resources is only one thread. It takes about three times longer to measure the CINR value. These analyses are illustrated in Table 3.

In the upper one in Table 3, "WiMAX Preamble Acquisition" resides at the remaining computational resource in core 1. In the lower one, "EV-DO initial acquisition" and "EV-DO track acquisition" reside in the remaining computational resource in core 3. The results of this study confirm the feasibility of simultaneous multiple RCS operation.

4. CONCLUSION AND FUTURE WORKS

In this paper, we present a newly-developed SDR prototype called the SDR core module. Various RCS can run on this module including simultaneous operation. As one of the promising applications, this module will be very suitable for monitoring functions in cognitive radio systems. Through some implementation results, the feasibility is confirmed.

To confirm this feasibility more deeply, a performance evaluation of the SDR core module with an RF processing part is essential. Moreover, it has to be confirmed that our newly-developed CTRL S/W can manage this module in power-saving fashion. Related to monitoring, the monitoring interval becomes longer in our current implementation. Our current study will solve this problem by adaptive change according to signal quality.

REFERENCES

[1] KUNISAWA Yoshio, KAMISAKA Daisuke, WATANABE Shingo and TAKEUCHI Yoshio "An initial acquisition scheme for software defined cdma2000 1xEV-DO radio terminal," Personal, Indoor and Mobile Radio Communications (PIMRC) 2007

[2] KAMISAKA Daisuke, WATANABE Shingo, KUNISAWA Yoshio, INOUE Takashi and TAKEUCHI Yoshio, "Study of software optimization for software defined radio using multiple-core dsp," GSPx 2006

[3] WATANABE Shingo, KUNISAWA Yoshio, KAMISAKA Daisuke, INOUE Takashi and TAKEUCHI Yoshio, "A software defined radio implementation of cdma2000 1xEV-DO on a single dsp chip designed for mobile handset terminal," VTC Fall 2006

[4] 3rd generation partnership project 2, 3GPP2 C.S0024 "cdma2000 High Rate Packet Data Air Interface Specification"

[5] IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems (802.16-2004)

[6] IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1 (802.16e-2005)

[7] Kosuke YAMAZAKI, Toru KITAYABU, Hiroyasu ISHIKAWA and Yoshio TAKEUCHI, "A study of effective SDR software porting for various hardware platforms," IEICE Technical Report, SDR2008-22

[8] Sandbridge technologies http://sandbridgetech.com

[9] picoChip –flexible wireless- http://www.picochip.com

[10] Yuji IKEDA, Kosuke YAMAZAKI, Toshiyuki MAEYAMA and Yoshio TAKEUCHI, "Proposal for an efficient software optimization method for software-defined radio," SDR Forum 2008

[11] Koji ENDA and Ryuji KOHNO, "A study on the estimation of traffic quantity and MAC protocol in radiowave environmental monitoring," SDR Forum 2007