

DESIGN AND PERFORMANCE OF IEEE 802.11A SDR SOFTWARE IMPLEMENTED ON A RECONFIGURABLE PROCESSOR

Hiroyuki SHIBA (NTT, Yokosuka-shi, Kanagawa, Japan, shiba.hiroyuki@lab.ntt.co.jp); Kazunori AKABANE (NTT, Yokosuka-shi, Kanagawa, Japan, akabane.kazunori@lab.ntt.co.jp); Munehiro MATSUI (NTT, Yokosuka-shi, Kanagawa, Japan, munehiro.matsui.munehiro@lab.ntt.co.jp); Kiyoshi KOBAYASHI (NTT, Yokosuka-shi, Kanagawa, Japan, kobayashi.kiyoshi@wslab.ntt.co.jp) and Katsuhiko ARAKI (NTT, Yokosuka-shi, Kanagawa, Japan, araki.katsuhiko@lab.ntt.co.jp)

ABSTRACT

Many software defined radio (SDR) prototypes have been developed, and SDR technology has been already applied to base stations and military radio equipment. However, it is difficult to realize a SDR terminal because of its stringent power consumption requirement. To solve this problem, NTT focused on a reconfigurable processor for an SDR terminal. We developed IEEE 802.11a software running on a reconfigurable processor and evaluated its performance by employing a simulator and an evaluation board with a real chip. The results of the evaluation confirmed that the developed software performed as designed.

1. INTRODUCTION

High-performance general-purpose processors and DSP-based and multiprocessor-based architectures have been used in software defined radio (SDR) prototypes, base stations, and military radio equipment [1]-[7]. However, since a mobile terminal has a stringent requirement as to its power consumption, it is difficult to realize such a terminal with these architectures. To solve this problem, NTT focused on a reconfigurable processor for an SDR terminal.

The reconfigurable processor, which has flexibility like software and high processing power with low power consumption, makes it possible to reconfigure circuits and data paths. In this regard, however, software needs to be optimized for it. Therefore, we developed IEEE 802.11a Physical (PHY) layer software and Media Access Control (MAC) layer software running on a reconfigurable processor. Since IEEE 802.11a requires a stringent time restriction and a large number of calculations, it is one of the hardest standards to confirm feasibility.

Section 2 of this paper describes the software implementation, and section 3 describes performance evaluation of developed software. Section 4 is a brief conclusion.

2. SOFTWARE IMPLEMENTATION

2.1. Reconfigurable Processor

We applied an Adaptive Computing Machine (ACM) processor [8]-[10] to an SDR terminal prototype. An ACM processor is a kind of a heterogeneous processor, which consists of many different nodes. The ACM chip has three kinds of node: Programmable Scalar Node (PSN), Domain Bit Manipulation Node (DBN) and Adaptive X Node (AXN). The PSN is a kind of a RISC processor. The DBN can efficiently perform bit-intensive algorithms. The AXN is a kind of parallel DSP. Figure 1 shows the ACM node layout of one ACM chip that we used in our prototype. Two PSNs, four DBNs and four AXNs are available with one ACM chip. Each node is connected via a Matrix Interconnect Network (MIN). The MIN interconnects the nodes through a 32-bit bidirectional packet switched network. Each node can support a multiplicity of unrelated tasks managed by the Hardware Task Manager (HTM).

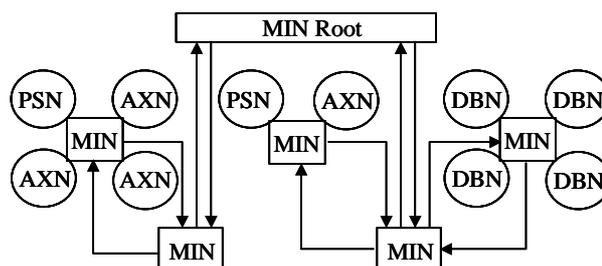


Fig. 1 ACM node layout

2.2. Software Design

Our software design policies are as follows: 1) Efficient task partition, 2) Efficient node usage, and 3) Compliance with the IEEE 802.11a standard. IEEE 802.11 PHY software and IEEE 802.11 MAC software are

implemented in accordance with our software design policies.

2.2.1. PHY Layer Software

IEEE 802.11a PHY transmitter and receiver block diagrams are shown in Fig. 2 and Fig. 3 respectively. There are many signal processing functions, such as Scrambler, Convolutional coder, Puncturer, Interleaver, Mapper and IFFT, etc., in the IEEE 802.11a PHY transmitter. The IEEE 802.11a PHY receiver consists of FFT, Depuncturer, Deinterleaver, Demapper and Viterbi decoder, etc. [11].

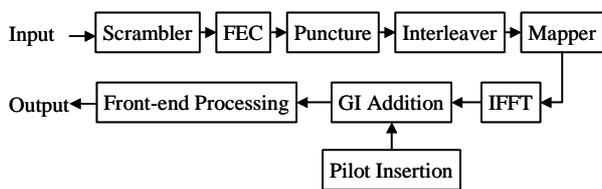


Fig. 2 PHY transmitter block diagram

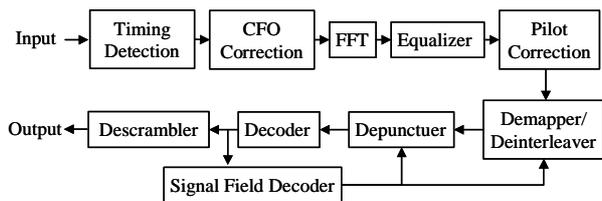


Fig. 3 PHY receiver block diagram

The following criteria and assumptions of PHY software design are applied:

- Compliance with IEEE 802.11a physical layer specifications.
- Minimization of ACM node number for all tasks.
- Sampling rate of 80 MHz for the AD and DA converters.

Additionally, there are two constraints on the ACM platform:

- The nodal memory size is 16 KB.
- The startup and teardown delays of tasks running on the DBN are several hundred cycles.

To design the tasks efficiently, each signal processing function was classified according to its features. The functions were then organized into a task. All tasks were assigned to nodes based on the node characteristics.

Because of the limited size of nodal memory, we had to pay close attention to total memory consumed by the task. Therefore, we estimated the total memory required by the tasks before assigning a node with them.

The task assignment of the IEEE 802.11a PHY transmitter software and IEEE 802.11a PHY receiver software are shown in Table 1 and Table 2. The DBN is suitable for bit operations, while the AXN is suitable for multiply-accumulate intensive signal processing. Therefore, the FFT and IFFT are mapped onto the AXN, respectively. Bit operation processings such as convolution coding, Viterbi decoding and interleaving, etc., are mapped onto the DBN. To process the signals effectively with regard to DBN characteristics, the TX Operation task is composed of several functions.

Table 1 Task assignment of PHY transmitter software

Node	Task	Function
AXN1	Pilot Insertion	Pilot Insertion
AXN2	IFFT, etc.	IFFT, etc.
AXN3	Tx Post processing	Preamble insertion
AXN4	Tx PHY Controller	MAC-PHY Interface
DBN1 DBN2	TX Operation	Scrambler, Convolutional coder, Puncturer, Interleaver, Mapper
DBN3	TX Signal	Signal field processing

Table 2 Task assignment of PHY receiver software

Node	Task	Function
AXN1	Pre FFT, etc.	Interpolation, GI removal, etc.
AXN2	Phase rotator	Phase rotator
AXN3	Demodulator, AGC	Deinterleaver, Demapper, AGC
AXN4	FFT, etc.	FFT, etc.
AXN5	FO handler, Post FFT, etc.	Channel estimation, Equalization, Frequency offset correction etc.
DBN1	Depuncturer	Depuncturing
DBN2	Viterbi	Viterbi decoder
DBN3	Signal decoder, Descrambler	Signal field decoding, Descrambling

2.2.2. MAC Layer Software

The following are the criteria and assumptions of MAC software design.

- Only station mode is considered.
- Mandatory features are implemented.
- Optional features such as Point Coordination Function (PCF) and IBSS power management are not implemented.
- The host interface is a PCI bus.
- The timers on the PSN can be used by MAC.
- Direct Memory Access (DMA) is available for transferring data.

Additionally, there are two constraints of the ACM platform:

- It does not support an Ethernet interface.
- The nodal memory size is 16 KB.

IEEE 802.11a MAC layer has a lot of functions such as Cyclic Redundancy Check (CRC), which is a signal processing function, fragmentation, Management Information Base (MIB) management, timers and Short Interframe Space (SIFS) response [12].

Function partitioning and task assignment were carried out in the same way as with the PHY layer software. Table 3 shows the task assignment of the MAC layer software. Since the PSN is suitable for control applications, most tasks of the MAC layer software were mapped onto the PSN.

In Table 3, all time-critical functions were organized with the TxRx Protocol Accelerator and mapped onto PSN1. The remaining functions were grouped together and mapped onto PSN2. The CRC32 task is a bit operation, so it was assigned to a DBN. PSN1 must be deployed closer to the node for realizing the MAC-PHY interface to minimize the response delay via MIN due to the time-critical task on PSN1.

Table 3 Task assignment of MAC layer software

Node	Task	Function
PSN1	Receive, Transmit, Management, Timer1, Timer2	Management messaging, Synchronization, MIB management, Duplicate Detection, Fragmentation, Auto Rate, Infrastructure station power management, Host interface management, NAV Timer, TSF Timer, etc.

PSN2	TxRx Protocol Accelerator	SIFS response generation, RTS generation, CCA, Backoff procedure, Beacon generation, etc.
DBN4	CEC32	CRC 32 Generation and FCS verification

3. PERFORMANCE EVALUATION

3.1. Memory Consumption

We evaluated memory usage of each task by using an ACM simulator. The simulator measured the program size and data area of each task. After measuring all tasks of each node, the memory consumption of each node was computed as shown in Table 4. Memory consumption is the ratio of the used memory to the nodal memory size. The results in the table show that all tasks were performed as designed, and all nodal memory usages were less than the nodes' limits. Moreover, we confirmed that task assignment to the node was performed efficiently because most of the nodal memory was used.

Table 4 Memory consumption of IEEE 802.11a software

Node	Memory consumption	Node	Memory consumption
PSN1	75 %	PSN2	84 %
DBN1	79 %	DBN2	80 %
DBN3	47 %	DBN4	60 %
AXN1	95 %	AXN2	95 %
AXN3	95 %	AXN4	95 %
AXN5	90 %		

3.2. Processing Cycle Count

We evaluated the processing time of each task running on the ACM simulator, which can generate a detailed log file containing the time stamp of each task.

Table 5 shows the measured processing cycle counts of tasks of the IEEE 802.11a PHY receive software (1 OFDM symbol at 54 Mbit/s mode was processed).

If each task was completed within 4 us, the processing delay did not accumulate. These results indicate that the chip clock must be more than 206 MHz. Figure 4 shows the processing cycle counts from the Pre FFT task through Viterbi task. These tasks were scheduled and parallelized.

An approximate formula of the relationship between data length and processing cycle count is,

$$\text{Processing cycle count} = 823 * (\text{Number of OFDM symbols} - 1) + 3091$$

This equation shows that each task was scheduled and parallelized effectively.

Table 2 Processing cycle count of IEEE 802.11a PHY receiver software (1 OFDM symbol, 54 [Mbit/s] Transmission rate)

Task	Cycle count	Task	Cycle count
Pre FFT	823	Demodulator	823
FFT	818	Depuncturer	777
Post FFT	443	Viterbi	806

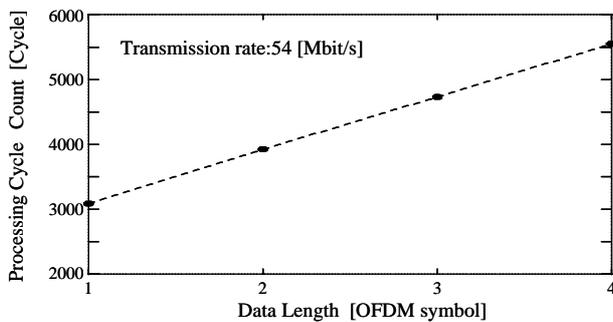


Fig. 4 Data length vs. processing cycle count

3.3. Power Consumption

We evaluated the power consumption of each task executing on the ACM chip by using the power consumption measurement system shown in Fig. 5.

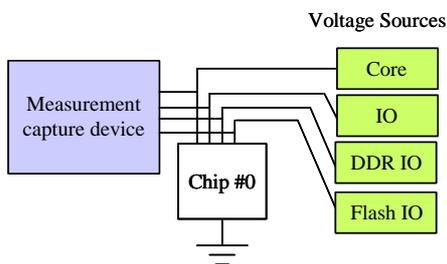


Fig. 5 Power consumption measurement system

Figure 6 and Figure 7 respectively show the measured power consumption of the Viterbi task and the IFFT task running on the ACM chip at 166 MHz. The plots show

that task processes consume about 100 mW, but the entire chip consumes more than 700 mW. The chip is slightly power hungry even when it is in the idle state of an SDR terminal. This problem stems from the fact that the current version of the ACM chip isn't designed to reduce power consumption. However, an ACM chip should be able to be designed as an SDR terminal by applying low power consumption design strategies such as time clock management.

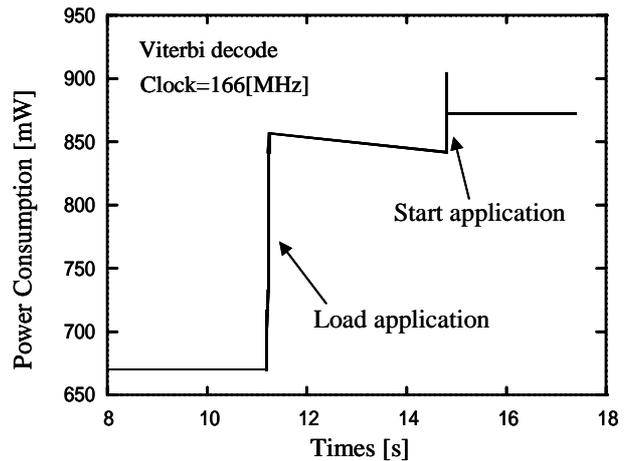


Fig. 6 Power consumption of Viterbi task

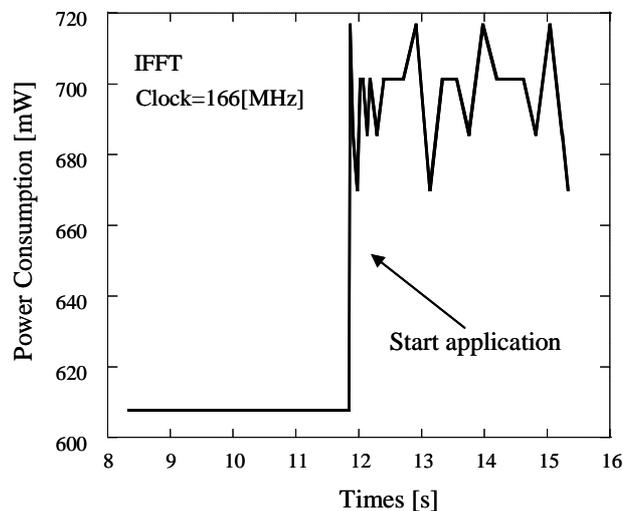


Fig. 7 Power consumption of IFFT task

4. CONCLUSION

We have designed new IEEE 802.11a PHY layer and MAC layer software for an SDR terminal employing a

reconfigurable processor and also evaluated its performance using an ACM simulator and an evaluation board with ACM chips. This paper described the task assignment to the ACM chip of IEEE 802.11 PHY layer and MAC layer software. The evaluation results confirmed that all tasks were mapped onto each node effectively and were performed as designed. In order to process the task without a delay, the clock speed of the chip has to be more than 206 MHz. The power consumption of the entire chip is not small enough for realizing an SDR terminal. Further power reduction is desired.

REFERENCES

- [1] 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.
- [2] P. W. Fox, "Software Defined Radios - Motorola's Wireless Information Transfer System(WITS)," *EUROCOMM*, pp. 43-46, 2000.
- [3] Pucker, L, "Applicability of the JTRS software communications architecture in advanced MILSATCOM terminals," *MILCOM*, pp. 533 – 537, vol.1, October 2003.
- [4] J.Mitola.III, "SDR architecture refinement for JTRS," *MILCOM*, pp. 214 – 218, vol.1, October 2000.
- [5] D. murotake, L. Oafes, A. Fuchs, "Real-time implementation of a reconfigurable IMT-2000 base station channel modem," *IEEE Comm. Mag.*, pp. 148 – 152, Vol.38, February 2000.
- [6] D. Efstathiou, "Base station transceiver trends for 3G and beyond wireless systems," *PIMRC*, pp. 117 – 121, vol.1, October 2001.
- [7] <http://www.vanu.com/>
- [8] <http://www.qstech.com/default.htm>
- [9] J. Watson, "AdAPTIVE COMPUTING IC TECHNOLOGY ENABLES SDR AND MMULTIFUNCTIONALITY IN NEXT-GENERATION WIRELESS DEVICES," *Software Defined Radio Conference*, pp. 185-188, November 2002.
- [10] B. Plunkett, D. Chou "Computational Efficiency: Adaptive Computing vs. ASICs," *Electronics, Circuits and Systems*, pp. 819-822, vol.2, September 2002.
- [11] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, High-speed physical Layer (PHY) specifications, IEEE std 802.11a, 1999.
- [12] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE std 802.11, 1999.