AN IMPLEMENTATION OF A SOFTWARE RADIO USING AN FPGA

Yuji IZAWA, David K. ASANO and Daisuke TAKEUCHI

Faculty of Engineering, Shinshu University
500 Wakasato, Nagano 380-8553, Japan
yizawa@cs.shinshu-u.ac.jp

Abstract – In this paper, an implementation of an AM radio receiver using a Field Programmable Gate Array (FPGA) is described. By using this approach, the flexibility of software and the speed of hardware can be combined, i.e., the receiver construction can be modified without new hardware. The receiver consists of a processor core and several logic blocks. Audio signal processing and adaptive control of the RF signals are performed by the processor, while the logic blocks handle high-speed signal processing tasks such as envelope detection. To program the FPGA, all of the circuits are first described using HDL, then the circuit description is downloaded to the FPGA. The experimental board that was built is described and some problems using this approach are reported.

KEYWORDS: FPGA, PROCESSOR CORE, SOFTWARE RADIO, RECONFIGURABLE SYSTEM, HDL

I. INTRODUCTION

Owing to the progress of processors and real-time operating systems, the application domain of real-time signal processing performed by software has expanded greatly. However, the performance of special purpose hardware implemented in VLSI, is still about 10 times higher than that of software. Nowadays, reconfigurable FPGAs (Field Programmable Gate Arrays) are available, so that by downloading a new configuration described by the HDL (Hardware Description Language), the operation of an FPGA circuit can be changed dynamically without any alteration of the hardware [1]. Using this kind of FPGA, we have developed several image transmission system prototypes, which have the flexibility of software and the high-speed operation of hardware [2][3].

Recently, the performance and capability of FPGAs have improved dramatically to the point where many applications can be realized by a system-on-a-chip design. One of the most exciting applications of this technology is Software Defined Radio. Due to the many communications standards being used around the world, research is currently in progress to develop a portable telephone that can be used with any communications standard by simply changing the software in the handset [4]. The advantage of this kind of device is clear — a single handset can be used globally. One of the candidates for the hardware to be used in software radios is FPGA technology [5].

In this paper, as a first step towards a software defined radio, we present an implementation of an FPGA-based software AM radio receiver. The experimental board used to evaluate our design is described and some open problems are reported.

II. AN FPGA-BASED RADIO RECEIVER

Software is suitable for complicated tasks such as adaptive control. On the other hand, hardware has superior performance, i.e., high-speed operation, for simple tasks. In practice, a processor such as a DSP is sometimes combined with an ASIC chip to implement a specific application. However, the constraints on the interface between the processor and the ASIC result in access time overhead. The advantage of FPGA-based systems is high-speed operation because all the components are on a single chip and therefore the complexities

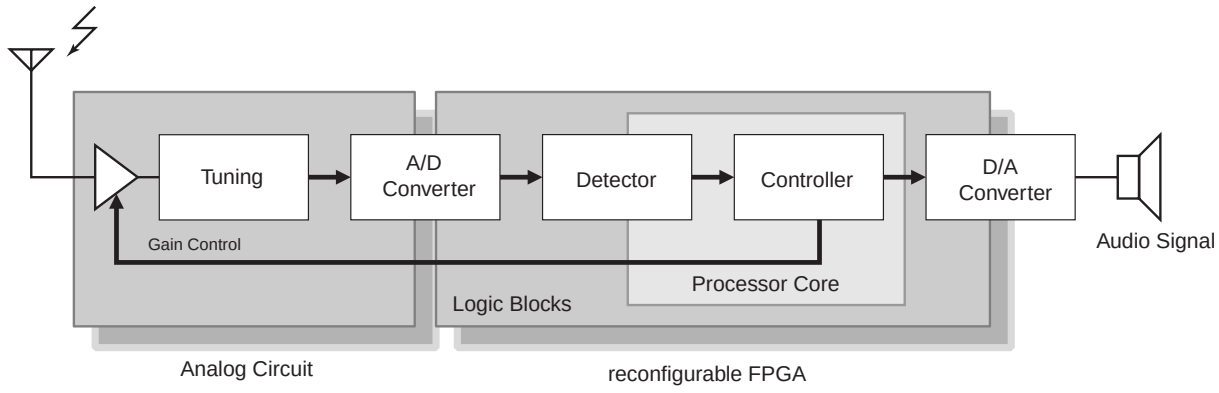

Fig. 1. Block diagram of a software radio receiver

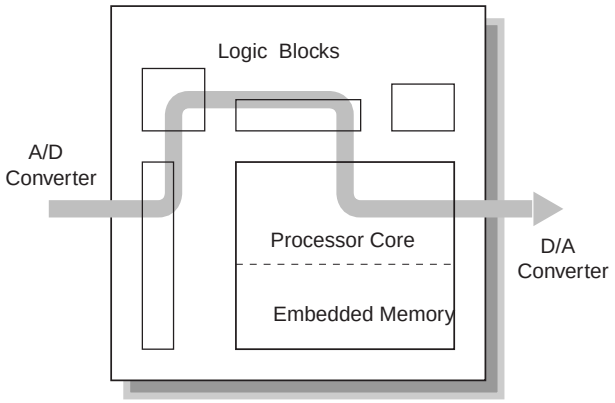

Fig. 2. Block diagram of the FPGA

of the interface between the processor and the external logic blocks can be reduced.

To investigate the problems involved in FPGA implementations, we have developed an AM radio receiver using a reconfigurable FPGA. The overall system is shown in Fig. 1. The system consists of two parts: an analog circuit and a reconfigurable FPGA. The analog circuit, under control of the FPGA, is used to tune the receiver to the desired radio station. Once the desired frequency band is selected, the analog signal is converted to digital data and sent to the FPGA. The FPGA extracts the audio signal using envelope detection and provides control signals for the analog circuit.

The internal architecture of the FPGA is shown in Fig. 2. The data from the A/D converter is processed by logic blocks and a processor core before begin sent to a speaker via a D/A converter.

In this system, partitioning of the circuits is performed to maximize the overall performance. The functions of the processor core are mainly audio signal processing and adaptive gain control of the RF (Radio Frequency) signal. The adaptive gain control allows the precision of the A/D converter to be used fully. If the desired signal is small compared to the dynamic range of the A/D converter, the resulting digital signal would have a large signal to noise ratio. With gain control, this problem is reduced.

III. THE PROCESSOR CORE

To provide software processing and control capabilities, we created a 16-bit RISC processor core inside the FPGA. The specifications of this processor core are shown in Table I. Although the memory size is limited, it is possible to expand the program and data memory size by using a larger FPGA or external memory.

The instruction set consists of the following instructions.

- memory access (load/store)
- arithmetic (add/subtract)
- logic (and, or, xor)
- shift
- comparison

TABLE I
PROCESSOR CORE SPECIFICATIONS

Architecture	16-bit RISC
Clock Frequency	16 MHz
Execution Speed	16 MIPS
Pipeline Stages	3
Number of Registers	8
Number of Instructions	27
Program Memory	>256 Words
Data Memory	>256 Words
FPGA	ALTERA FLEX10K-50
Programming Language	Assembler

- branch
- nop

It is also possible to add additional instructions that handle control of the logic blocks and data I/O. The instruction set, as well as the architecture, of this processor are optimized for real-time audio signal processing. At present, this processor core can only execute instructions in the embedded memory.

To program the processor, there are two methods. The first is to download the instructions when the FPGA is programmed. The second method is to download the instructions from a separate personal computer. As explained in the next section, the FPGA's embedded memory can be accessed from a personal computer. Thus, by using a cross assembler to generate code for the processor core and downloading the code into the embedded memory, the processor core can be programmed.

IV. EXPERIMENTAL EVALUATION

The experimental board that was built to evaluate our design is shown in Fig. 3. The large main board contains the FPGA and interface components, while the analog circuit portion of the system and A/D converter are located on the smaller board attached to the upper left part of the main board. This system also has an ISA-BUS interface (the part on the right of the main board) so that it can be inserted into a personal computer. Via this interface, the embedded memory in the FPGA can be accessed.

In RF band signal processing, an A/D converter with high precision and high-speed operation is required. In our system, we use an A/D converter with a precision of 10-bits and a sampling rate of 40 Msamples/sec. The input to the A/D converter is controlled by the processor core so that the full dynamic range of the A/D converter can be exploited.

Tuning of the RF signal is performed by a conventional analog circuit in our present implementation. After the desired radio signal is selected, we use envelope detection to extract the audio signal. The envelope of

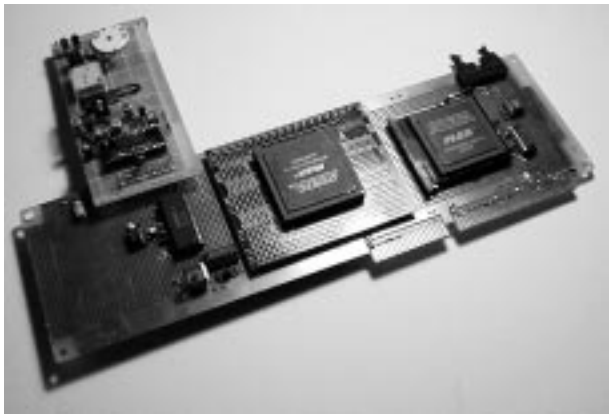

Fig. 3. The experimental board

the AM carrier signal is detected using the FPGA logic blocks and processor core. The resulting digital signal is transformed to an audio signal by a D/A converter with a precision of 12 bits. From our test results, we verified that the receiver functioned properly by listening to the audio signal using a speaker.

At present, the data necessary to program the FPGA is downloaded from a ROM, but it is also possible to transmit the entire demodulation scheme from a remote location. This flexibility is one of the requirements of a true software defined radio system.

V. FUTURE WORK

Digitization of the tuning process is our next goal. In order to tune the radio digitally, high speed digital filters, which involve multipliers, are necessary. It is possible to realize non-recursive filters using deep pipelines, but for recursive filters, a high-speed FPGA must be used.

If a purely digital implementation is not possible, it will be necessary to develop a hybrid tuning scheme using an FPGA and conventional analog circuits.

It is also necessary to develop a method to improve the precision of the A/D conversion process using, for example, multiple A/D converters.

Before the A/D conversion process, some sort of software controllable bandpass filter is necessary. This will allow the desired frequency band to be selected from software as well as to remove unwanted interference.

To improve the efficiency of the digital processing after A/D conversion, control of the A/D converter's clock signal is important. This should also be controlled using software in the FPGA.

Methods to reduce the power consumption should also be examined.

VI. CONCLUSIONS

As a first step in developing a software defined radio, a prototype model of a software AM radio receiver was implemented using a reconfigurable FPGA. Using an experimental board, AM audio signals were successfully demodulated and output to a speaker. Tuning of the desired radio station was done using an analog circuit, but a software tuning algorithm is currently under development.

REFERENCES

[1] Cook et al., "A simulation tool for dynamically reconfigurable field programmable gate arrays," *ASIC Conference & Exhibit '95*, pp. 167–170, Sept. 1995.

[2] Y. Imai, Y. Izawa, and Y. Kinoshita, "A proposal of programmable accelerators for image signal processing," *IEICE Student Research Meeting*, p. 100, Mar. 1994.

[3] Y. Imai, J. Kawano, Y. Izawa, and Y. Miyashita, "Programmable accelerators for image signal processing," *Picture Coding Symposium Japan (PCSJ94)*, pp. 41–42, Oct. 1994.

[4] J. Mitola III, "Technical challenges in the globalization of software radio," *IEEE Commun. Mag.*, vol. 37, pp. 84–89, Feb. 1999.

[5] M. Cummings, "FPGA in the software radio," *IEEE Commun. Mag.*, vol. 37, pp. 108–112, Feb. 1999.