Software AM Radio Implementation

Daniel Iancu, John Glossner, Hua Ye, Youssef Abdelila, Stuart Stanley
Sandbridge Technologies Inc.
1 N Lexington Ave. 10th fl.
White Plains, NY 10601
daniel.iancu@sanbridgetech.com

Abstract

In multimedia multiprotocol communications convergence devices it is desirable to reduce the number of components required. Traditionally, each communication protocol requires a separate baseband processor implemented by discrete components. For a system that would like to support WCDMA, GSM/GPRS, GPS, Bluetooth, AM, FM, and 802.11b, at least five separate chips may be required. The space required on the board by these multiple devices makes small form factor implementations difficult. A desirable alternative to multiple chip implementations is a Software Defined Radio (SDR) approach. In SDR, a single baseband chip is reconfigured such that multiple baseband protocols may execute on the same device. Sandbridge Technologies has designed a platform for reconfigurable software defined radio communications. As part of this project we have implemented an efficient software AM receiver. In our implementation, all functions associated with the AM receiver, including most of the filtering and demodulation are executed in SW using only two threads of the Sandbridge Sandblaster multithreaded processor core.

1. Introduction

Traditional communications systems have typically been implemented using custom hardware solutions. The advantages of reconfigurable Software Defined Radio (SDR) solutions versus hardware solutions are significant. First, reconfigurable solutions are more flexible allowing multiple communication protocols to dynamically execute on the same transistors thereby reducing hardware costs. Specific functions such as filters, modulation schemes, encoders/decoders etc., can be reconfigured adaptively at run time. Second, several communication protocols can be efficiently stored in memory and coexist or execute concurrently. This significantly reduces the cost of the system for both the end user and the service provider. Third, remotely reconfigurable protocols provide simple and inexpensive software version control and feature upgrades. This allows service providers to differentiate products after the product is deployed. Fourth, the development time of new and existing communications protocols is significantly reduced providing an accelerated time to market. Development cycles are not limited by long and laborious hardware design cycles. With SDR, new protocols are quickly added as soon as the software is available for deployment. Fifth, SDR provides an attractive method of dealing with new standards releases while assuring backward compatibility with existing standards.

Sandbridge Technologies has designed a multi-threaded processor capable of executing DSP, Control, and Java code in a single compound instruction set architecture optimized for handset radio applications [1][2]. The Sandbridge design overcomes the deficiencies of previous approaches by providing substantial parallelism and throughput for high-performance DSP applications while maintaining fast interrupt response, high-level language programmability, and very low power dissipation [3].

As shown in Figure 1, the design includes a unique combination of modern techniques such as a SIMD Vector/DSP unit, a parallel reduction unit, and a RISC-based integer unit. Each processor core provides hardware support for concurrent execution for up to eight threads. All state may be saved from each individual thread and no special software support is required for interrupt processing. The machine is partitioned into a RISC-based control unit that fetches instructions from a set-associative instruction cache and a Vector/DSP unit that performs non-associative arithmetic in parallel. Instruction space is conserved through the use of compounded instructions that are grouped into packets for execution. The machine also contains instruction set support for Java execution.

The platform is programmed in a high-level language such as C, C++, or Java. The program is then translated using an internally developed supercomputer class vectorizing, parallelizing, multithreaded compiler [4]. The tools are driven by a parameterized resource model of the architecture that may be programmatically generated for a variety of implementations and organizations. The source input to the tools, called the Sandbridge architecture Description Language (SaDL), is a collection of python source files that guide the generation and optimization of the input program and simulator. The compiler is retargetable in the sense that it is able to handle multiple possible implementations specified in SaDL and produce an object file for each implementation. The platform also supports many standard libraries (e.g. libc, math, etc.) that may be referenced by the C program. The compiler generates an object file optimized for the Sandblaster architecture.

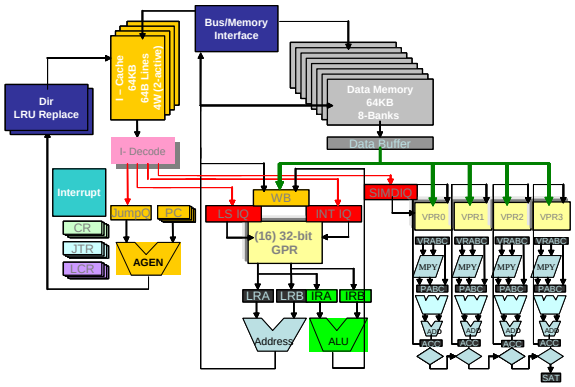

Figure 1. Sandblaster Processor Core

In this paper we discuss an Amplitude Modulated (AM) radio implementation written in C that is compiled to the Sandblaster processor.

2. AM Radio Theoretical approach

An AM composite signal can be viewed as a superposition of N in band equally spaced carriers, each modulated by a modulation signal $\varphi_k(t)$. If multi-path effects are ignored, the AM composite function can be written as:

$$s(t) = \mathrm{Re}\left[\sum_{i}^{N} \varphi_i(t) e^{-jw_i t}\right] + n(t) \qquad (1)$$

Where: $n(t)$ is thermal noise, mostly coming from the receiver front end.

Using a rectangular windowing function, centered on one of the carriers, equation (1) can be rewritten as:

$$s(t) = \sum_{m=-\infty}^{+\infty} g(t - mT_c)\,\mathrm{Re}\left[\sum_{i}^{N} \varphi_i(t) e^{-jw_i t}\right] + n(t) \qquad (2)$$

where: $g(t-mT_c)=\sigma(t-mT_c)\ \sigma[(m+1)T_c-t)]$, T_c is the period of the carrier, and $f_c=1/T_c$ is the carrier frequency,

$$\sigma(t) = \begin{cases} 1\ for & t \ge 0 \\ 0\ for & t < 0 \end{cases}$$ is the unit step function.

Since the AM band contains multiple carriers, in the following, by 'carrier frequency' we mean the carrier we intend to demodulate.

First, the AM composite signal is filtered using a band-pass filter centered on the carrier frequency f_c. For the mathematical approach, without loosing generality, we consider a rectangular band-pass filter centered at f_c, with out-of-band attenuation $1/\alpha$, and a pass-band attenuation of zero. Next, in the demodulation process, the filtered signal is multiplied by the demodulation function [5] at the carrier frequency f_c to obtain:

$$d(t) = s(t)\,\mathrm{Re}\left[e^{-jw_k t}\right] =$$

$$= \sum_{m=-\infty}^{\infty} g(t - mT_c) \cdot \qquad (3)$$

$$\cdot \left[\varphi_k(t)\cos^2 w_k t + \alpha\left(\sum_{i\neq k}^{N} \varphi_i(t)\cos w_i t \cdot \cos w_k t + n(t)\cos w_k t\right)\right]$$

Multiplying equation (3) by $g(t-lT_c)$ and integrating over a carrier cycle, with the assumption that $\varphi_k(t)$ is constant over a cycle period T_c, we obtain one sample of the modulation function for the l^{th} period of the carrier:

$$\int_{-mT_c}^{(m+1)T_c} d(t)g(t - lT_c)dt =$$

$$= \int_{-mT_c}^{(m+1)T_c} \left\{\sum_{m=-\infty}^{+\infty} g(t - mT_c)g(t - lT_c) \cdot \right. \qquad (4)$$

$$\left. \cdot \left[\varphi_k(t)\cos^2 w_k t + \alpha\left(\sum_{i\neq k}^{N} \varphi_i(t)\cos w_i t \cdot \cos w_k t + n(t)\cos w_k t\right)\right] dt \right\}$$

After calculation, the right hand side integral in (4) can be rewritten as:

$$\begin{cases} 0 \quad for \quad m \neq l \\ \varphi_k^{(l)} \int_{-lT_c}^{(l+1)T_c} \cos^2 w_k t \cdot dt + \\ \frac{\alpha\varphi_k^{(l)}}{2} \int_{-lT_c}^{(l+1)T_c} \sum_{i\neq k}^{N} [\cos(w_i - w_k)t + \cos(w_i + w_k)t]dt + \\ + \alpha \int_{-lT_c}^{(l+1)T_c} n(t)\cos w_k t \cdot dt \\ \qquad\qquad for \quad m = l \end{cases} \qquad (5)$$

In the right hand side of expression (5), the three integrals have the following meaning: the first represents the modulator, the second is the in-band interference due to the other AM carriers, and finally, the third one represents noise. The spectrum of intermodulation components (w_i-w_k) resulting from in-band interference terms is spaced above (at higher frequencies) the information spectrum, and therefore can be easily filtered out using a low pass filter. The remaining noise after multiple stages of filtering and

integration over a carrier period is negligible and can be ignored for simplicity.

After integration and filtering, the final expression for equation (5) becomes:

$$\begin{cases} 0 & for \quad m \neq l \\ \frac{T_c}{2}\varphi_k^{(l)} & for \quad m = l \end{cases}$$

Now, we can perform a summation over all carrier periods indexed $l \in (-\infty,+\infty)$ and scale with $2/T_c$. Finally, the sampled version of the modulation function will have the following expresssion:

$$\varphi_k(\tau) = \varphi_k[nT_c] \cong \sum_{l=-\infty}^{+\infty} \varphi_k^{(l)} \Delta[(l-m)T_c]$$

where: $\Delta[(l-m)T_c] = g(t-lT_c)\, g(t-mT_c)$.

3. Implementation

The system implemented is a coherent AM receiver 0[5][6], illustrated in block diagram form in Figure 2. The highlighted HW blocks may be shared between multiple communication protocols.

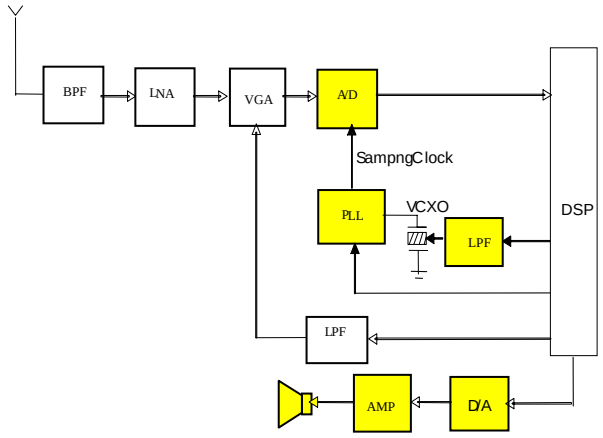

Figure 2. HW implementation of the AM receiver

The signal from the antenna is first filtered using an AM band pass filter, then amplified and sampled by the A/D. The PLL with additional prescaler – divider logic provides the sampling clock for the A/D and is configured by the processor. The sampling frequency may be fine tuned by voltage tuning the reference crystal oscillator. The control signals for both the AGC and AFC are entirely generated by the DSP.

Software Implementation

The SW blocks are illustrated in Figure 3. The SW implementation closely follows the mathematical description from above. The oversampled signal from the A/D at eight times the carrier is first filtered using a second order IIR band pass filter, centered at the carrier frequency, with 3 dB attenuation bandwidth of 5 KHz. The filtered signal is then multiplied with the cosine of the sampled signal (LO) and integrated over eight samples. After integration, the data goes through: 1:16 decimation, filtering using a 96 tap 80dB FIR low pass filter, rescaling, and DC removal. Finally, the data is sent to the D/A. The AFC and the AGC functions are also implemented in SW. The coefficients for the filters are pre-computed and stored in nonvolatile memory for each carrier in the AM frequency band.

System Validation

The algorithms have been designed and simulated in Matlab, using a simulated wave form consisting of five consecutive carriers at 900, 1000, 1010, 1020, 1030, 1600 KHz with AWGN added to the composite signal. The normalized power for each carrier is one. Each of the carriers is modulated with single tones at different modulation frequencies. The spectrograms after demodulation and after the last stage of filtering are shown in Figure 4 and Figure 5 respectively. The demodulated carrier is at 1010 KHz, modulated at 3.3 KHz.

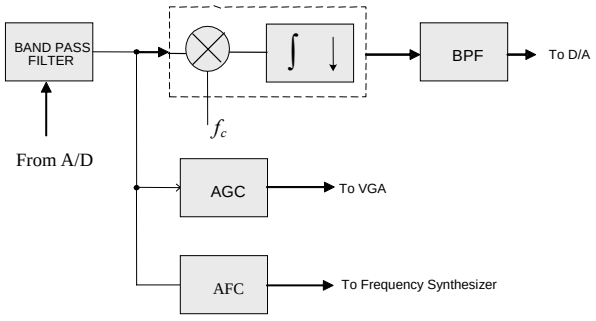

Figure 3. SW blocks for the AM receiver.

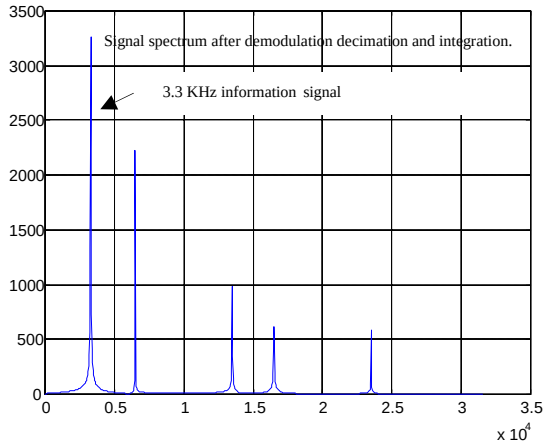

Figure 4. Signal spectrum after demodulation, integration and decimation.

4. Results

The processor architecture and tools for the Sandbridge Sandblaster core are described in Section 1. The AM receiver is implemented using standard ANSI C code with automatic assembly language generation using our vectorizing multithreading compiler. The platform is able to implement the entire AM radio processing chain is under 160MHz. This is about 5% of the processing capacity of the SB9600 platform.

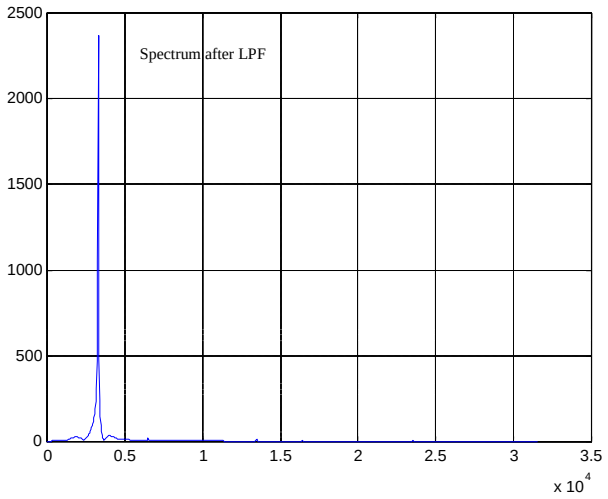

Figure 5. Information signal after the last stage of filtering.

5. Conclusions

In this paper we have presented a new algorithm for implementing a software defined AM radio. Using control signals generated by software, we are able to control the demodulation of AM signals using significantly less complex operations than otherwise achievable. This has been incorporated into the SB9600 product offering and validated on prototype silicon.

References

[1] J. Glossner, T. Raja, E. Hokenek, and M. Moudgill, **"A Multithreaded Processor Architecture for SDR"**, *The Proceedings of the Korean Institute of Communication Sciences*, Vol. 19, No. 11, pp. 70-84, November, 2002.

[2] J. Glossner, E. Hokenek, and M. Moudgill, **"Multithreaded Processor for Software Defined Radio"**, *Proceedings of the 2002 Software Defined Radio Technical Conference*, Volume I, pp. 195-199, November 11-12, 2002, San Diego, California.

[3] J. Glossner, D. Iancu, J. Lu, E. Hokenek, and M. Moudgill, **"A Software Defined Communications Baseband Design"**, *IEEE Communications Magazine*, Vol. 41, No. 1, pages 120-128, January, 2003.

[4] J. Glossner, S. Dorward, S. Jinturkar, M. Moudgill, E. Hokenek, M. Schulte, and S. Vassiliadis, **"Sandbridge Software Tools"**, *Accepted for publication in the 3rd annual Systems, Architectures, Modeling, and Simulation (SAMOS) Conference*, Samos, Greece, July 21-24, 2003.

[5] J. P. Costas, **"Synchronous Communication"**, *Proc. PRE* vol.44 pp. 1713-1718, Dec. 1956

[6] S. Haykin, *Communication Systems*, John Wiley and Sons. Inc. 1983.