Hand held analog television over WiMAX executed in SW

Daniel Iancu*[a], Hua Ye[a], Murugappan Senthilvelan[a], Vladimir Kotlyar[a], John Glossner[ab], Mayan Moudgill[a], Sitij Agrawal[a], Sanjay Jinturkar[a], Andrei Iancu[a], Vaidyanathan Ramadurai[a], Gary Nacer[a], Stuart Stanley[a], Mihai Sima[c], Jarmo Takala[d]
[a] Sandbridge Technologies, Inc. 1 North Lexington Ave., White Plains, NY 10601 USA;
[b] Delft University of Technology, EEMCS, Delft, The Netherlands;
[c] University of Victoria, P.O.Box 3055, STN CSC, Victoria, B.C. V8W 3P6, Canada;
[d] Tampere University of Technology, Korkeakoulunkatu 1, FIN-33720 Tampere, Finland

ABSTRACT

This paper describes a device capable of performing the following tasks: it samples and decodes the composite video analog TV signal, it encodes the resulting RGB data into a MPEG-4 stream and sends it over a WiMAX link. On the other end of the link a similar device receives the WiMAX signal, in either TDD or FDD mode, decodes the MPEG data and displays it on the LCD display. The device can be a hand held device, such as a mobile phone or a PDA. The algorithms for the analog TV, WiMAX physical layer, WiMAX MAC and the MPEG encoder/decoder are executed entirely in software in real time, using the Sandbridge Technologies' low power SB3011 digital signal processor. The SB3011 multithreaded digital signal processor includes four DSP cores with eight threads each, and one ARM processor. The execution of the algorithms requires the entire four cores for the FDD mode. The WiMAX MAC is executed on the ARM processor.

Keywords: software defined radio, multithreading, digital signal processor

1. INTRODUCTION

Although digital TV and HDTV deployment is underway, analog TV broadcasting (NTSC, PAL, SECAM) is still the most common TV system. It has become increasingly attractive to provide mobile platforms that offer the capability to view analog TV broadcasts while relaying them over the internet using one of the existing wireless communication protocols, which in our case, is WiMAX. WiMAX[1] is a long range, fixed, portable, or mobile wireless technology specified in the IEEE 802.16 standard. It provides high-throughput broadband connections similar to 802.11 wireless LAN systems but with much larger range. Possible applications for WiMAX include "last mile" broadband connections, hotspots, cellular backhauls, and high-speed enterprise connectivity for business. Since the IEEE 802.16 standard defines a Media Access Control (MAC) layer that supports different physical layers and also defines the same Logical Layer Control level l for different Local and Wide Area Networks (LAN and WAN), it opens up the possibility of bridging different communication networks together. A common MAC allows easier implementation of multi-mode multi-radios. At the same time, it also simplifies system management and roaming issues. Multi-mode multi-radio systems have historically been implemented using either multiple separate chip sets or specific system-on-chip solutions with replicated internal hardware. Recently, a more cost effective approach which has gained in popularity is to implement a Software Defined Radio (SDR) whereby the entire physical layer in executed in software. A SDR is also capable of dynamically switching waveform execution and thus reusing existing silicon resources. Our analog television and WiMAX implementations described in this paper are true SDR solutions, executed entirely on the same platform using the Sandbridge Technologies multithreaded multiprocessor SB3011 DSP[2,3].

2. ANALOG TV SYSTEM

In NTSC systems[4,5], each video frame consists of two color fields or 525 lines. As shown in Figure 1, the first 20 lines of color field I, constitute the field-blanking period that carries the vertical synchronization pulses. The vertical synchronization pulses indicate the start of a new video frame. The color field II starts from the middle of line 263,

*DIancu@sandbridgetech.com; phone 914 2878510; fax 914 287 8501; sandbridgetech.com

Multimedia on Mobile Devices 2007, edited by Reiner Creutzburg, Jarmo Takala, Jianfei Cai,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6507, 650709,

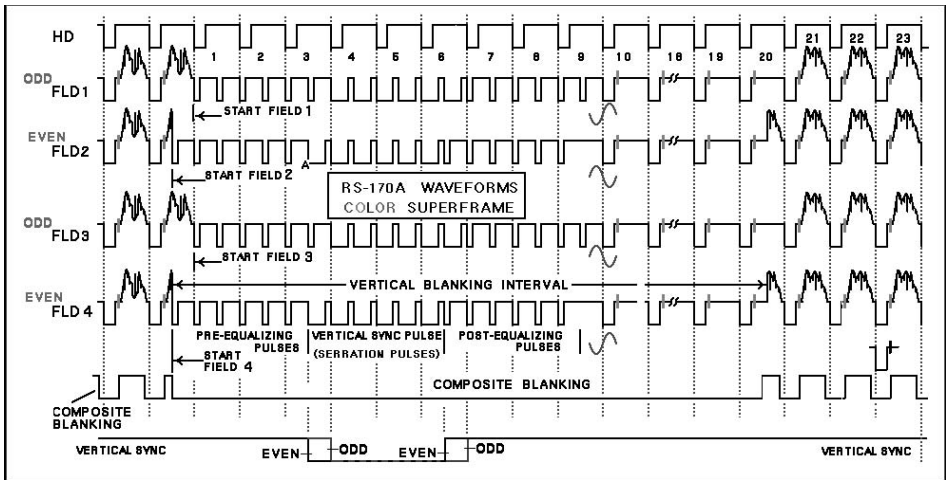

Fig. 1. Color field blanking sequence for NTC analog television system.

carrying another 20 lines for the second field-blanking period indicating the middle of the video frame. The rest of the lines in color fields I and II are the displayable video lines. The analog TV receiver digital signal processing blocks implemented in the SB3011 DSP are illustrated in Figure 2. The horizontal synchronization and vertical synchronization sequences are detected and tracked by a delay lock loop. The input video composite signal is then adjusted to achieve proper DC and IRE (Institute of Radio Engineers) scaling levels. The active video information is extracted line-by-line as follows: the Y/C, or luminance and color, are separated first, followed by the color I/Q demodulation and the RGB signals reconstruction. The RGB signals for the current video frame are finally converted to pixels and displayed on the LCD screen. For the interested reader more detailed information on the signal processing algorithms is described in our earlier paper[6].

2.1 Real time implementation

Details of the real time implementation are described in our previous paper[7]. The analog video composite signal (RGB or S-video output of a DVD player, video recorder or analog TVset) is sampled at 14.318 MHz and copied continuously into a circular buffer in the SB3011 on chip memory. There are three tasks: horizontal synchronization, vertical synchronization, and video decoding. The synchronization tasks detect the beginning of each video line and video frame.

The video decoding task demodulates the composite signal into the chrominance and luminance components and then converts the components into displayable RGB stream. Video decoding is most computationally intensive. It requires about a billion arithmetic operations per second. A real-time implementation requires effective parallelization of the video decoding phase. There are a number of choices on how the work can be partitioned. Video decoding consists of a embarrassingly few parallel steps, such as FIR filtering and color-space conversion. First, each video line can be decoded independently. For partitioning onto *N* threads, the choices range from having all *N* threads cooperate on every video line to process *N* video lines on separate threads. It turns out that the former choice requires too much

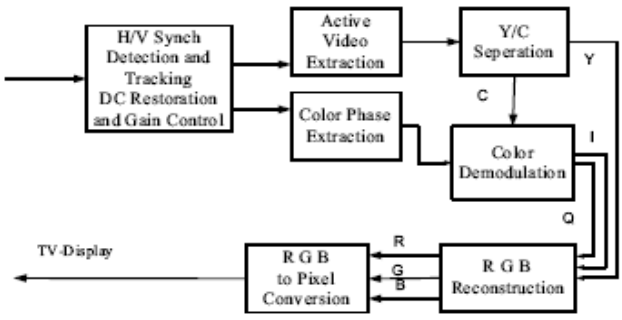

Fig. 2. Signal processing diagram of the NTSC receiver.

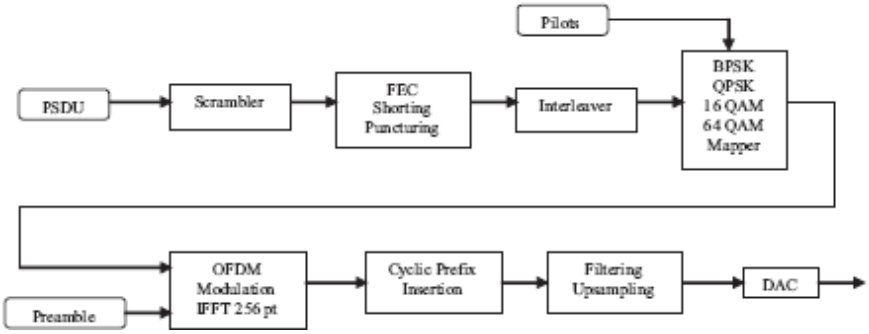

Fig. 3. WiMAX transmitter block diagram specified by the IEEE standard.

synchronization, and the latter choice consumes too much L1 memory. Leaving data in L2 memory slows down processing and negates much of the benefit of parallelization. Our eventual implementation used four teams of four threads each. Each team cooperates on its own video line. Such an arrangement minimizes data sharing and synchronization while still using only L1 memory. Our implementation is also able to adapt to changing computational capabilities. The quality of output can be gracefully degraded by shutting down one or more teams and interpolating or skipping the missing lines.

The parallel implementation is described in detail in a previous paper[7] with the emphasis on algorithm time/space partitioning as required by the features of the platform.

3. WIMAX SYSTEM

The physical layer transmitter pipeline for the OFDM PHY as specified in IEEE 802.16 standard is shown in Figure 3. The OFDM signaling format was selected in preference to competing formats such as single-carrier CDMA due to its superior multipath performance, permitting significant equalizer design simplification to support operation in non line of sight fading environments. Figure 4 shows the 802.16 OFDM physical layer signal processing block diagram for the receiver. The inputs to the A/D converter are the I and Q baseband signals coming from the RF front end. The I/Q signals are first 2:1 decimated and filtered to the FFT sampling frequency. The Rx processing includes the reversed Tx blocks plus the receiver synchronization, timing and frequency tracking. All of the processing blocks are executed entirely in software. Detailed description of the implementation is available in an earlier paper[8].

3.1 Real time implementation

The WiMAX transmit and receive algorithms are implemented as concurrent multithreaded pipelines [8]. The pipelines consist of all the processing steps such as FFT, filtering, scrambling, etc. To implement a pipeline on the Sandblaster processor we have (a) aggregated steps into stages, and (b) decided how to assign threads to the computations within a stage.

The WiMAX transmitter is a simpler algorithm and we use it to illustrate our partitioning methodology. There are four steps: (a) OFDM data symbol/preamble generation, (b) FFT, (c) filtering, and (d) data transfer to the D/A converter. Based on profiling of the sequential ANSI C implementation, we allocate two processor threads for symbol generation, three threads for FFT, two threads for filtering and one thread for copying data to the D/A. The total number of threads is eight and thus the WiMAX transmitter may be implemented in a single Sandblaster processor core.

The pipeline implementation is shown in Figure 5(a). Symbol generation and filtering are partitioned naturally across two threads. Each thread works either on the I or on the Q channel. To avoid the overhead of partitioning the FFT, we replicate FFT processing across three threads. Each thread works on a different symbol. Our implementation illustrates two methods for partitioning work among threads: either we partition a unit of work (an OFDM symbol in this case) across multiple threads, or we process multiple units of work concurrently. In general, we might have multiple units processed concurrently, with each unit being partitioned across a team of threads.

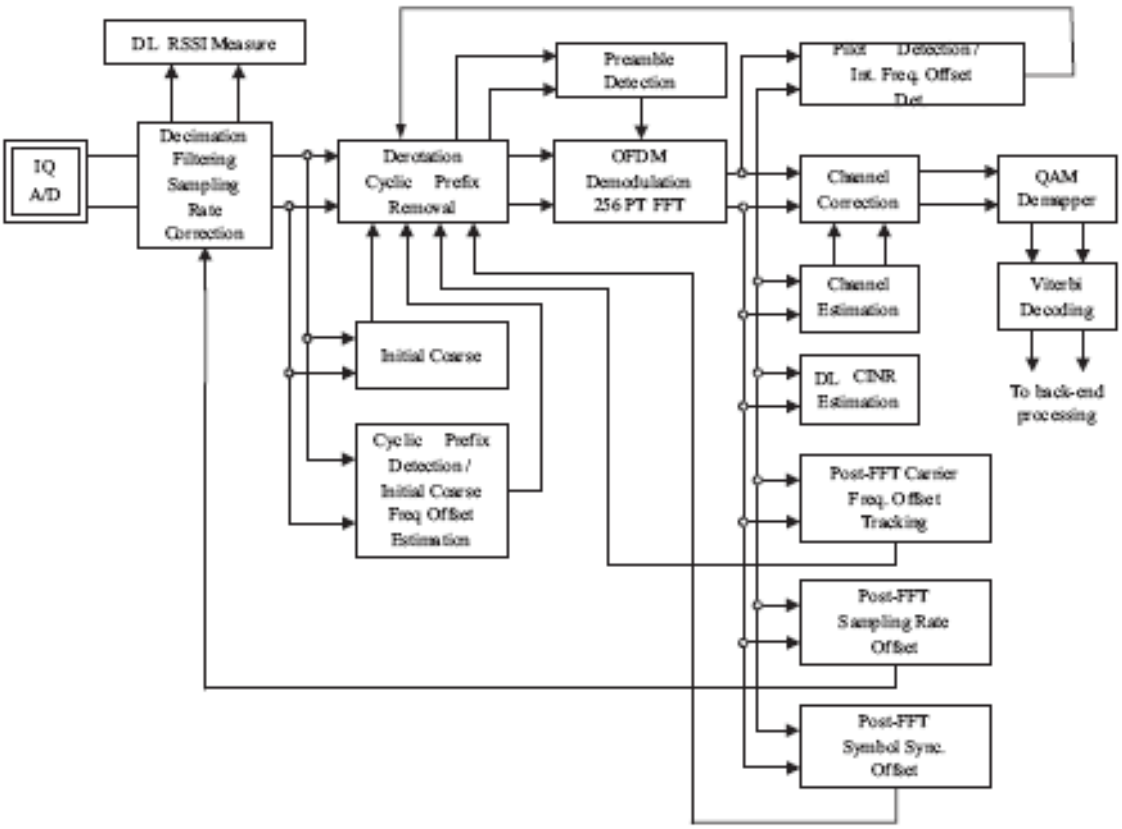

Fig. 4. WiMAX receiver signal processing blocks.

Therefore, for each stage we have to specify (a) the number of concurrent teams and (b) the number of threads in each team. The partitioning of work within each team is dependent on the particular computation. Using this strategy, the FFT stage is assigned to three teams. Each of the teams has a single thread. Symbol generation is assigned to one team of two threads, same as filtering. The D/A transfer is assigned to a single team of one thread. We use double buffering to communicate between stages. When data is communicated between a stage with one team and a stage with multiple teams (e.g., symbol generation to FFT, FFT to filtering), round-robin scheduling is used to decide which team is communicated with.

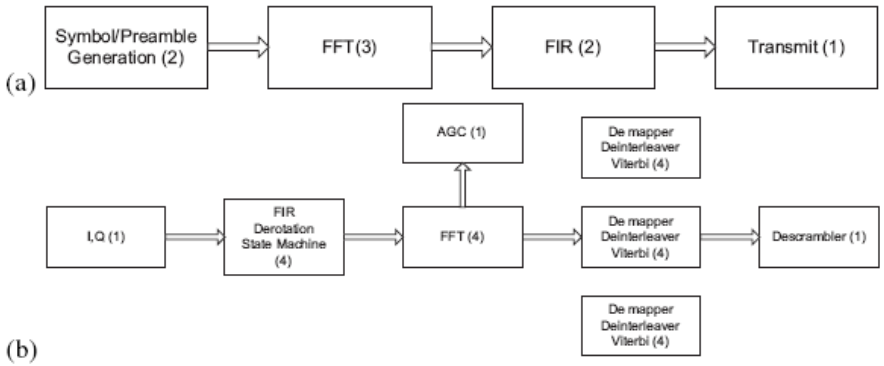

Fig. 5. (a) Transmitter and (b) receiver pipeline; each box is a team, the size of each team is in parenthesis.

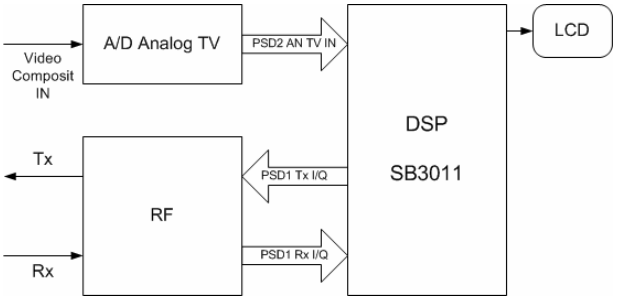

Fig. 6. Device block diagram.

4. SANDBLASTER PLATFORM

The platform includes the SB3011 DSP and the corresponding software tools, including a real-time multi-threaded POSIX-compliant operating system, an optimizing ANSI C compiler, and a fast, cycle-accurate simulator. The SB3011 is a multi-core multi-threaded DSP. There are four cores and each core executes eight thread units. A thread unit supports a mix of integer arithmetic, branch instructions and 4-way SIMD vector instructions. Each core is capable of issuing four multiply-accumulate operations per cycle. Thus at 600MHz, the SB3011 DSP is capable of 9.6 billion multiply-accumulate operations per second. Each core contains 64K of fast L1 memory and 256K of L2 memory. There is no penalty for L1 reads and writes. The memories are globally addressable across cores. Remote reads/writes are communicated via a unidirectional 64-bit wide ring.

Two DMA channels per core are available for bulk transfers. The SB3011 digital signal processor also contains two reconfigurable high speed bi-directional RF interfaces, PSD1 and PSD2. Each interface is capable of accommodating Rx/Tx IQ signals. In our case, PSD1 is allocated to WiMAX while the other one is dedicated to the analog television as illustrated in Figure 6.

5. CONCLUSIONS

We have presented a device capable of executing both WiMAX (in FDD or TDD mode) and analog TV on the Sandblaster platform. It is entirely programmed in ANCI C and all the baseband function is entirely executed in SW, resulting in a true SDR based platform. The WiMAX bit rate is in accordance with the WiMAX Forum Wave1 recommendation with a maximum bit rate of 2.9Mbps in FDD mode. The processor usage for 2.9 Mbps is as follows: WiMAX Rx and Tx 16 threads and eight threads, respectively, analog TV three threads, MPEG decoding (QCIF) one thread, and MPEG encoding four threads (QCIF).

REFERENCES

1. Institute of Electrical and Electronics Engineers, *IEEE standard for local and metropolitan area networks Part 16. Air interface for fixed broadband and wireless access system*, Std. 802.16, 2004.
2. J. Glossner, T. Raja, E. Hokenek, and M. Moudgill, "A Multithreaded Processor Architecture for SDR," *Proc. Korean Institute of Communication Sciences* 19(11), 70-84 (2002).
3. J. Glossner, M. Schulte, M. Moudgill, D. Iancu, S. Jinturkar, T. Raja, G. Nacer, and S. Vassiliadis, "Sandblaster low-power multithreaded SDR baseband processor", *Proc. Workshop on Applications Specific Processors*, 53-58 (2004).
4. National Television System Committee, http://www.ntsc-tv.com/
5. International Telecommunication Union, *Television Systems*, recommendation ITU – R BT.470-4, 1995.
6. H. Ye, D. Iancu, J. Glossner, V. Kotlyar, and A. Iancu, "Signal processing algorithms for DSP implementation of analog TV receivers," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, 1168-1171 (2006).
7. V. Kotlyar, D. Iancu, J. Glossner, Y. He, and A. Iancu, "Real-time software implementation of NTSC analog TV on Sandblaster SDR platform," *Proc. Karlsruhe Workshop on Software Radios*, 171-176 (2006).
8. D. Iancu, H. Ye, E. Surducan, M. Senthilvelan, J. Glossner, V. Surducan, V. Kotlyar, A. Iancu, G. Nacer, and J. Takala, "Software implementation of WiMAX on the Sandbridge Sandblaster platform," *Proc. Workshop on Embedded Computer Systems: Architectures, Modeling, and Simulation*, LNCS 4017, 435-446 (2006).