Software Implementation of WiMAX on the Sandbridge SandBlaster Platform

Daniel Iancu[1], Hua Ye[1], Emanoil Surducan[1], Murugappan Senthilvelan[1], John Glossner[1,2], Vasile Surducan[1], Vladimir Kotlyar[1], Andrei Iancu[1], Gary Nacer[1], and Jarmo Takala

[1] Sandbridge Technologies, One North Lexington Ave., White Plains, NY 10601, USA
[diancu, huaye, esurducan, msenthilvelan, jglossner, vsurducan, vkoltyar, aiancu, gnacer]@sandbridgetech.com

[2] Delft University of Technology, Computer Engineering, EE, Delft, The Netherlands

[3] Tampere University of Technology, Tampere, Finland jarmo.takala@tut.fi

Abstract. This paper describes a Sandbridge Sandblaster system implementation including both hardware and software elements for a WiMax 802.16e system. The system is implemented on the fully functional multithreaded Sandblaster multiprocessor SB3010 SoC chip. The entire communication protocol, physical layer and MAC, has been implemented in software using pure ANSI C programming language and it executes in real time. In this paper, we also present a radio propagation analysis specific to the Samos island at the workshop location, and the DSP execution performance.

1 Introduction

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, hotspot and cellular backhaul, and high-speed enterprise connectivity for busines. 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 (LLC) 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 multi-mode and multi-radios easier implementations and at the same time it also simplifies system management and roaming issues. A multi-mode multi-radio system has historically been implemented using either multiple separate chip sets or specific System on Chip (SoC) solutions with replicated internal hardware. Recently, a more cost effective approach has gained in popularity. A Software Defined Radio (SDR) implements all of the physical layer in software and is capable of dynamically switching waveform execution and thus reusing existing silicon resources. Our WiMAX implementation described in this paper, is an SDR solution.

Table 1. Frequency bands, maximum power at the antenna and EIRP, NA: Not available

Parameters / Country	CE	CE	US	US
Frequency band [MHz]	2400–2483.5	5470–5725	2400–2483.5	5725–5850
Maximal power to antenna [mW]	NA	NA	200	1000
EIRP [dBm]	20 [100mW]	30 [1W]	23 [200mW]	53 [200W]

2 WiMax System Background

The WiMAX 802.16 standard specifies a high throughput non-line-of-site (NLOS) communications link along with connectivity between network endpoints. It specifies an RF spectrum in the 2 to 66 GHz range, including both licensed and unlicensed bands. The maximum bit rate as currently defined is 70 Mbps. The spectrum allocation and the maximum power at the antenna input, for both licensed and unlicensed bands are also specified in [1]. Table 1 lists the maximum power allowed by the standard at the antenna input and the Effective Radiated Power (ERPC) compared to an isotropic radiator, for different geographic areas.

Receiver Sensitivity Calculation. The receiver sensitivity is the measure of the signal strength for a specified modulation mode and bit-error rate (*BER*) that must be present at the receiver input in order to be able to detect the radio frequency signal and to demodulate correctly the transmitted data. The receiver sensitivity (P_{rx}) is a function of the Receiver Noise Floor (*NF*) and the Signal to Noise Ratio (*SNR*). The theoretical receiver sensitivity can be expressed as

$$P_{rx} = SNR + NF \tag{1}$$

where $SNR = (E_b/N_0)(R/B)$, E_b is the energy required per bit of information, N_0 is the thermal noise in 1Hz of bandwidth, R is the system data rate, and B is the system bandwidth.

The *BER* for a BPSK modulation system, with Additive White Gaussian Noise (AWGN) is given by

$$BER = \frac{1}{2}\mathrm{erfc}(E_b/N_0)^{1/2} \tag{2}$$

where $\mathrm{erfc}(\cdot)$ is the *complimentary* error function. The theoretical values of the *BER* as a function of E_b/N_0 are presented in Table 2.

Table 2. Theoretical values of BER as a function of E_b/N_0

BER	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
E_b/N_0 [dB]	4.3	6.8	8.4	9.6	10.6	11.3

Table 3. Receiver sensitivity for BPSK modulation at 10^{-6}

Modulation	R/B	BER	E_b/N_0	N_0 [dBm]	N [dB]	SNR	P_{rx} [dBm]
BPSK	1/2	10^{-6}	10.6	-113	7.5	7.6	-85.4

The receiver Noise Floor (N_F) is the sum of thermal noise (N_0) and the noise figure (N) of the receiver as follows

$$N_F = N + N_0 \tag{3}$$

where $N_0 = kTB$, is the thermal noise power measured in Watts, N is the noise figure of the receiver, k is the Boltzman constant, T is the system absolute temperature usually assumed to be 290 K, and B the channel bandwidth measured in Hz. All these entries are summarized in Table 3 and they are in accordance with the standard recommendations.

Link Budget Calculation. The link analysis provides the estimation of the required transmitted power level in order to cover for a desired range [2]. The sum of *EIRP* (transmitted power plus antenna gain) and receiver absolute sensitivity $|P_{rx}|$ must be equal to the sum of link loss (LL) and Fade Margin (FM) [3, 4]. The link loss includes the Path Loss (PL), at frequency F over the range D, and the external Microwave Circuit Loss (MCL) (switch, antenna cables, connectors) and is shown in the following:

$$EIRP + |P_{rx}| = PL(D,F) + FM + MCL. \tag{4}$$

To estimate the maximum range with a given *EIRP* and receiver sensibility P_{rx} it is necessary to estimate the fading loss, the RF front-end external circuit loss and to calculate *PL*. Table 4 illustrates the path loss versus distance D for the most popular propagation models. In Table 4, the columns refer to the following:

Table 4. Link budget for different channel models, path loss (PL) given in dB at 2.45 GHz

D [km]	CCIR	Hata-l city	Hata-s city	Hata suburb	Hata open	ITU	WI-LOS	WI-NLOS	SPLM
2	120.4	142.4	125.4	112.4	91.7	105.3	118.3	129.7	141.6
2.5	123.9	146.0	128.9	116.0	95.2	106.2	120.8	133.4	146.6
3	126.8	148.8	131.8	118.9	98.1	107.0	122.8	136.4	150.6
3.5	129.2	151.3	134.2	121.3	100.5	107.9	124.6	139.0	154.0
4	131.3	153.4	136.3	123.4	102.6	108.6	126.1	141.2	157.0
5.5	136.4	158.4	141.4	128.4	107.6	110.4	129.7	146.4	164.0
6.5	139.0	161.1	144.0	131.1	110.3	111.4	131.6	149.2	167.7
11	147.3	169.4	152.3	139.4	118.6	115.0	137.5	157.9	179.3
12	148.7	170.7	153.7	140.8	120.0	115.6	138.5	159.3	181.2

Table 5. Maximum range for the unlicensed frequency bands: * calculated with HATA Open model, 0 dB antenna gain and 12 dB loss and ** calculated with HATA Suburban model, 0 dB antenna gain and 12 dB loss

Frequency band [MHz]	2400–2483.5	5470–5725	2400–2483.5	5725–5850	5725–5795 5815–5850
EIRP [dBm]	20 (100mW)	30 (1W)	23 (200mW)	53 (200W) 30dBm in the antenna	36 (4W)
*Max LOS [km]	3	5.5	3.58	23.8	8
**Max NLOS range [km]	0.8	1.23	1	5.3	1.8

- CCIR: [4] Empirical model for the combined effect of free-space path loss and terrain-induced path loss published by CCIR-Committee Consultative International des radio Communication, now ITU-R.
- HATA: [4] Based on Okamura *et al.* (Empirical curves).
- Hata-l.city: Large City model (building height greater than 15m).
- Hata-s.city: Medium to Small City model.
- Hata-suburb: Suburban model.
- Hata-open: Free space model.
- ITU: Line of sight (LOS), experimentally tested for D larger than 3km as follows

$$PL(\mathrm{dB}) = 92.45 + 20\log(D + F) \tag{5}$$

  where D is measured in km and F in GHz.
- WI: "Walfish-Ikegami" is an empirical and semi deterministic model for mobile radio propagation (COST-231 project). WI has a good fit for the frequencies in the range of 800 to 2000 MHz and the range of 0.02 to 5 km.
- WI-LOS: [4] No obstruction in direct path (LOS) (base station antenna height 30m)
- WI-NLOS: [4] No-line-of-sight (NLOS). For the path loss calculation the following values have been used:
  - Base station antenna height (hb) : 4–50m,
  - Terminal antenna height (hm): 1–3m,
  - Building separation (b): 20–50m,
  - Width of street (if not specified, $b/2$ is recommended), and
  - Angle of the incident wave to streets (assumed 90 degrees).
- SPLM: [5] Suburban Path Loss Model it is a modified Hata-Okamura model.

For the Samos island case, we have chosen the values for Hata-open and Hata-suburban models. The maximal distance possible to be covered within the maximum range of the allowable EIRP values, specified in the standard, are presented in Table 5. For the theoretical analysis we have considered a 0dB gain antenna, $FM = 10$dB fading loss and $MCL = 2$dB loss.

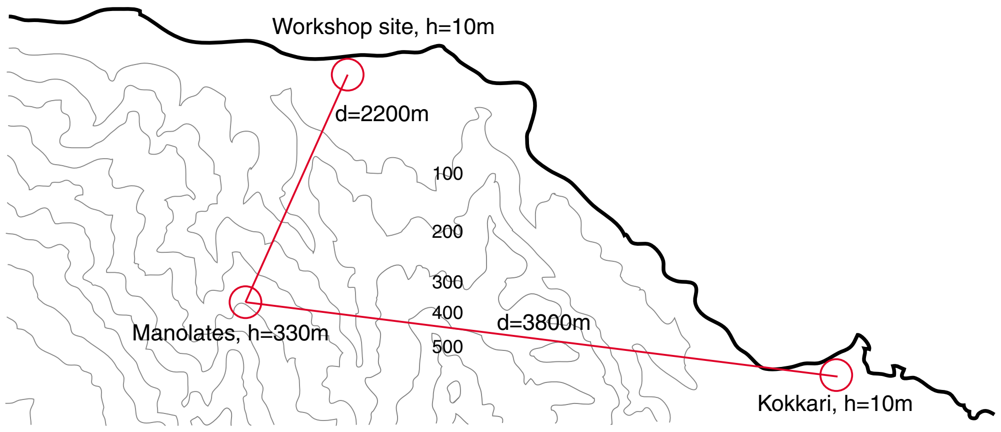


Fig. 1. Satellite map of the demo place

3 System Description

Figure 1 shows a satellite map of Samos Island. Our goal is to connect Agios Konstantinos to Kokkari through a WiMax link. There is 5.5 km between Agios Konstantinos and Kokkari but there is no direct LOS path. In order to meet the link budget for the unlicensed band a repeater is required. Based on the receiver sensitivity calculations and availability, for our demonstrations we used a standard off-the-shelf 802.11 WiFi transceiver which supports 7MHz bandwidth operation mode and meets our estimated sensitivity requirements. Using the 802.11 front-end also gives us the option of executing IEEE 802.11 a/b/g standard on the same platform.

A repeater must be able to support the Full Duplex Mode (FDD) mode on two different bands, for instance we can receive on the 2.4GHz band and transmit on the 5.6GHz band or vice-versa. Since the WiFi front-end supports only a TDD mode, there is need for two transceiver chips for each system. We note that we can also make use of the additional WiFi transceiver for Multi Input Multi Output (MIMO) communication modes. To summarize, the end to end system consists of: (a) TDD mode platforms at both ends on either 2.4 or 5.6 GHz and (b) a repeater in between, in FDD mode with LOS to both ends.

A hardware block diagram is illustrated in Fig. 2. The hardware components of both the end unit and repeater are identical. The RF front-end consists of two RF transceiver chips and one high rate sampling AD/DA (Analog-to-Digital/Digital-to-Analog converter) directly connected to the SB3010 Sandblaster evaluation board through a high speed parallel interface. Power amplifiers are connected to each transmitter and bandpass filters are placed between the antennas and the receivers. The system can operate with a single antenna employing Rx/Tx switches or two separate Tx and Rx antennas. We describe results for the second case. All serial controls for the various chips are generated by software executing on the SB3010.

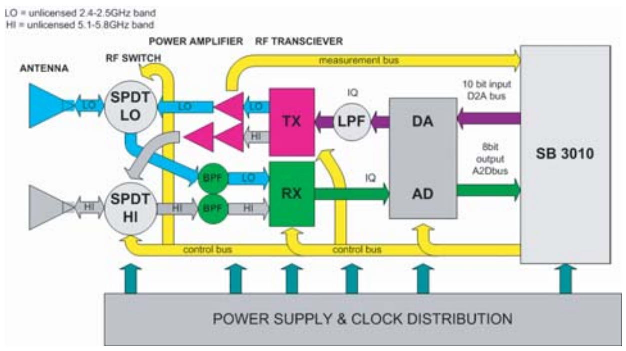


Fig. 2. Hardware block diagram of the WiMAX system

4 Sandblaster Platform

The SB3010 chip [6] consists of four Sandblaster DSP cores connected by a unidirectional deterministic and opportunistic ring network. The SB3010 chip is fabricated in 90nm and each DSP core runs at 600MHz. Each DSP core has a branch unit, a scalar Arithmetic Logic Unit (ALU), a Single Instruction Multiple Data (SIMD) vector unit and a load/store unit. These execution resources are time multiplexed equally among 8 threads per core. Each thread has its own set of scalar and SIMD vector registers.

Instruction Set Architecture. Each thread executes 64-bit compound instructions. A compound instruction can contain up to 3 concurrently executed compound operations. For example a load can be issued in parallel with an arithmetic operation and a branch. The following instruction computes the inner product of a vector with itself:

```
Label:
    vmulred %ac3, %vr7, %vr7, %ac3 ||
    lvu %vr7, %r8, 8 ||
    loop 0, %lc0, Label
```

The ”vmulred” operation multiplies each of four 16-bit elements contained in the vector register `%vr7` with itself and accumulates the products into an accumulator register `%ac3`. At the same time, the `lvu` operation increments the scalar register `%r8` by 8 and loads the next 4 values from the resulting address. The `loop` instruction decrements the loop count register `%lc0` and repeats the instruction if the register is non-zero.

Each Sandblaster core is capable of completing an instruction from a thread on every 600MHz cycle provided there are no stalls due to memory access. In particular, each core is capable of completing a 4-way multiply-accumulate (MAC) instruction at every 600MHz cycle. *Across four cores this adds up to 4×600×4=9600 million MACs per second.*

Since core execution resources (ALU, branch, etc.) are shared equally among the 8 threads - we can view a core as an 8-way multiprocessor with each processor running at 600MHz/8 = 75 MHz. We denote this performance as a ”thread cycle”. *In the rest of the paper, we report memory latencies and algorithm complexity using thread cycles.*

Memory Structure. Each core has a 32kB *instruction cache*. Data memory is not cached and is divided between a 64kB Level 1 (L1) and 256 kB Level 2 (L2) memory. A load from L2 memory incurs a pipeline stall. Stores into L2 are issued through a FIFO and do not block unless the FIFO is full. In practice up to four threads can simultaneously store into L2 without blocking. The L1 memory is divided into 8 banks of 8kB each. A particular implementation detail is that there is no penalty if the parity (odd/even) of the thread is the same as the parity of the bank. There is a single cycle penalty both for loads and for stores if the parities of the thread and the bank do not match. The instruction in the inner product example will complete within a single thread cycle, if it is executed on an even thread and `%r8` points into an even bank. The compiler tries to ensure memory affinities and the processor tools can automatically generate linker scripts that optimize memory access.

Programming Environment. The Sandblaster programming tools include: a supercomputer-class vectorizing compiler, a fast simulator, and real-time operating system (RTOS) that implements POSIX threads standard [7]. *Our WiMax implementation is written entirely in ANSI C using POSIX API for thread management.* We rely on the optimizing compiler to produce highly efficient machine code from straight-forward C source. The compiler automatically vectorizes most of the loops that occur in signal processing and media applications. It performs semantic analysis of input programs and automatically recognizes saturating arithmetic in ANSI C [8]. For example the single-instruction loop for an inner product is generated automatically from the following source:

```
int i, s;
for (I = 0, s = 0; I < N; i++) {
    s += A[k]*A[k];
}
```

The Sandblaster simulator is capable of executing over 100 million instructions per second on a 3GHz x86 computer [9]. The Sandblaster RTOS is capable of multiplexing an arbitrary number of software threads onto hardware threads. Software threads can be designated as pinned or non-pinned. Pinned threads are removed from the general thread scheduler and by convention, their stacks are allocated to L1 memory. Non-pinned threads can be rescheduled any time the operating system chooses and can be allocated based on the scheduling policy implemented in pthreads.

5 WiMAX Algorithms

The physical layer transmitter pipeline for the OFDM PHY as specified in IEEE 802.16 [1] is shown in Fig. 3(a). The OFDM signaling format was selected in preference to competing formats such as single-carrier (SC) CDMA due to its superior multipath performance, permitting significant equalizer design simplification to support operation in NLOS fading environments.

Figure 3(b) shows the 802.16 OFDM PHY receiver block diagram. The back-end signal processing block is the reverse of the transmitter pipeline shown in Fig. 3(a). Note that a Reed-Solomon (RS) decoder is not required for the 2.9 Mbps BPSK mode.

Figure 4 shows the 802.16 OFDM PHY front-end signal processing block diagram. The inputs to the A/D converter are the I and Q baseband signals coming from the RF

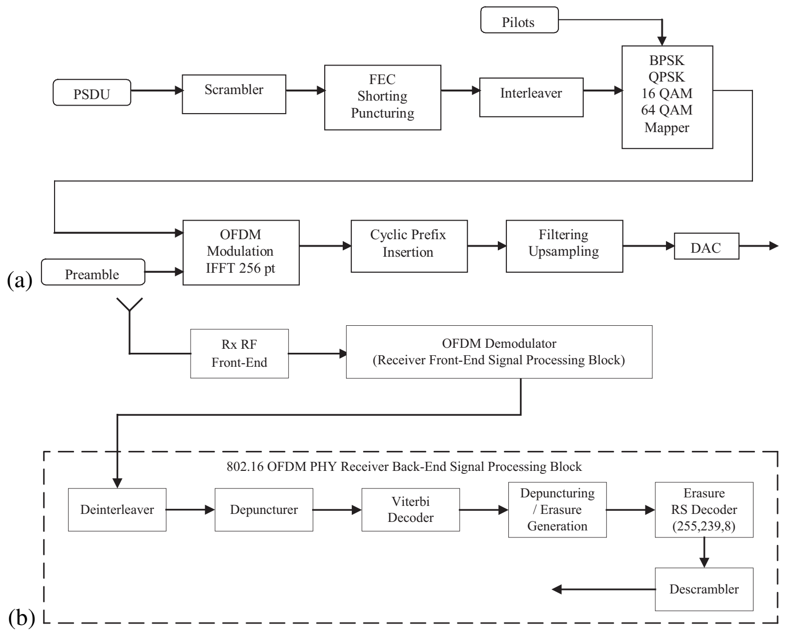

Pilots
PSDU
Scrambler
FEC Shorting Puncturing
Interleaver
BPSK QPSK 16 QAM 64 QAM Mapper
OFDM Modulation IFFT 256 pt
Cyclic Prefix Insertion
Filtering Upsampling
DAC
Preamble
(a)
Rx RF Front-End
OFDM Demodulator (Receiver Front-End Signal Processing Block)
802.16 OFDM PHY Receiver Back-End Signal Processing Block
Deinterleaver
Depuncturer
Viterbi Decoder
Depuncturing / Erasure Generation
Erasure RS Decoder (255,239,8)
Descrambler
(b)

Fig. 3. Block diagrams of 802.16 OFDM PHY (a) transmitter and (b) receiver

chip. The I/Q signals are first 2:1 decimated and filtered to the FFT sampling frequency F_s. The FFT sampling frequency is proportional to the channel bandwidth (B) as shown in the following:

$$F_s = \lfloor nB/8000 \rfloor 8000 \tag{6}$$

where $\lfloor \cdot \rfloor$ is floor function. In our implementation, $B = 7$ MHz, $n = 8/7$, $F_s = 8$ MHz, and, therefore, the ADC sampling frequency will be at $2F_s = 16$ MHz.

The Automatic Gain Control (AGC) block calculates the new value required to establish the appropriate control bits used to set the gain level for the two gain stages in the RF chip based on the signal energy measurements as follows

$$E = \sum_{i=0}^{N-1} \left[r_I(i)^2 + r_Q(i)^2 \right] \tag{7}$$

where $r_I(i)$ and $r_Q(i)$ are the decimated I/Q signals and N is the number of samples in a symbol including the guard period.

The AGC algorithm runs under coarse setting and fine setting. In the coarse setting mode, the AGC monitors the input energy E and once the incoming signal is detected, an initial AGC setting is calculated by comparing the measured energy level E with a preset target energy level. The AGC coarse setting will allow the Voltage controlled Gain Amplifier VGA to pull the input signal within the ADC's dynamic range. Once the coarse setting is complete, the AGC gain will be kept constant while the receiver goes through a training process to achieve synchronization with the transmitter.

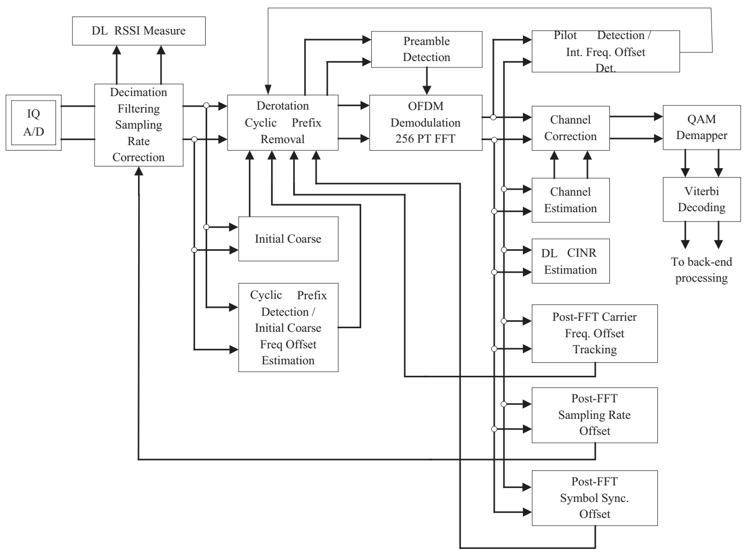

DL RSSI Measure
IQ A/D
Decimation Filtering Sampling Rate Correction
Derotation Cyclic Prefix Removal
Preamble Detection
OFDM Demodulation 256 PT FFT
Pilot Detection / Int. Freq. Offset Det.
Channel Correction
QAM Demapper
Channel Estimation
Viterbi Decoding
To back-end processing
DL CINR Estimation
Initial Coarse
Cyclic Prefix Detection / Initial Coarse Freq Offset Estimation
Post-FFT Carrier Freq. Offset Tracking
Post-FFT Sampling Rate Offset
Post-FFT Symbol Sync. Offset

Fig. 4. 802.16 OFDM PHY front-end signal processing block diagram

The AGC then enters a fine setting mode where the energy E measured during the preamble symbol duration will be compared with the preset energy target. Based on this measurement the LSB of the VGA control bits are adjusted accordingly.

The derotation operation is performed in time domain as follows

$$r'_I(i) + jr'_Q(i) = (r_I(i) + jr_Q(i))\, e^{\frac{-j2\pi\Delta f}{F_s}}, \;\; i = 0 \ldots N-1 \tag{8}$$

where j is the imaginary unit. The purpose of the derotation is to correct for the frequency offset Δf that is detected by the initial coarse estimation and fine tracking. The cyclic prefix is then removed and the remaining I and Q samples are further used in the OFDM demodulation.

Both short and long preambles are defined to assist in channel estimation, timing, and carrier frequency estimation. The time domain periodicity properties of the preamble can be exploited to detect the preamble sequence and symbol boundary. The following equations are used to detect the preamble sequences:

$$c(j+n) = \sum_{i=0}^{127} r(i+n)r(i+j+n); \tag{9}$$

$$n_{max} = \arg\left(\max_n \sum_{j=0}^{L-1} \sqrt{\Re\left[c(j+n)\right]^2 + \Im\left[c(j+n)\right]^2}\right) \tag{10}$$

where $r(k)$ is a complex signal sample after decimation and L is the number of samples in the guard period. The autocorrelation peak at position n_{max} indicates the presence of preamble sequence and its starting sampling position.

The coarse fractional carrier frequency offset can be estimated as

$$\Delta f = \frac{F_s}{2\pi} \tan^{-1}\left(\frac{\Im\left[c(n_{max})\right]}{\Re\left[c(n_{max})\right]}\right). \tag{11}$$

After the initial coarse symbol timing and frequency offset estimation, fine estimation and adjusting algorithms are required in the frequency domain. There are 8 pilot signals inserted in each data-bearing OFDM symbol. These are used to perform post FFT carrier frequency offset tracking, symbol synchronization tracking, and sampling rate offset tracking.

Channel estimation is performed when receiving the long preamble symbol that has 100 pilots spaced two subcarriers apart (excluding the DC subcarrier). The transmitted pilots can be represented as X_s, $s = 0, 1, \ldots, 99$. The corresponding received subcarriers at the pilot locations can be represented as Y_s. The channel frequency response at the pilot subcarrier locations can be represented as H_s. The least-square estimate of the channel frequency response at the pilot subcarrier location s, $\tilde{H}_s$, is given by the following equation:

$$\tilde{H}_s = \frac{Y_s}{X_s}, \;\; s = 0, 1, \ldots, 99. \tag{12}$$

The channel frequency response at the remaining 100 non-pilot subcarriers can be readily estimated using linear interpolation.

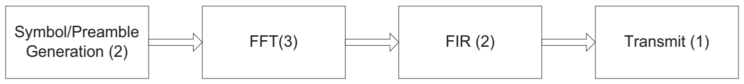
It is mandatory that the Down Link (DL) receiver measures and reports the mean and standard deviation of the ratio of the Received Signal Strength Information and the Carrier to Interference and Noise Ratio (RSSI / CINR) to the Base-Station (BS) within a strict time requirement. Both RSSI and CINR measurements are performed using preamble sequences. The QAM demapping is Gray coded and the implementation supports up to four soft bit demapping.

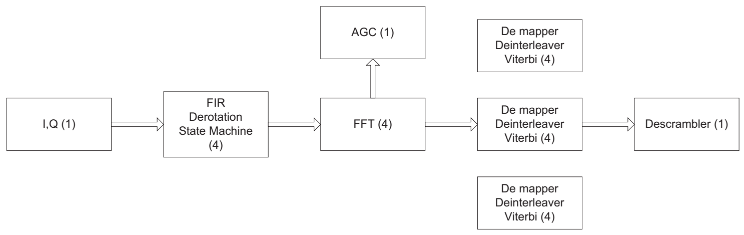
6 Multithreaded Multiprocessor Implementation

The WiMAX transmit and receive algorithms are implemented as concurrent multi-threaded pipelines. The pipelines consist of all the processing steps such as FFT, filtering, scrambling, etc. To implement a pipeline on Sandblaster processor we have (a) aggregate 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 copy to D/A converter. Based on profiling of the sequential ANSI C implementation, we allocate 2 processor threads for symbol generation, 3 threads for FFT, 2 threads for filtering and 1 thread for copying data to the D/A. The total number of threads is 8 and thus the WiMAX transmitter may be implemented in a single Sandblaster processor core. The pipeline implementation is shown in Fig. 5(a). Symbol generation and filtering are partitioned naturally across two threads. Each thread works on either the I channel or 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.

(a)




AGC (1)
De mapper Deinterleaver Viterbi (4)
I,Q (1)
FIR Derotation State Machine (4)
FFT (4)
De mapper Deinterleaver Viterbi (4)
Descrambler (1)
De mapper Deinterleaver Viterbi (4)

(b)

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

Our implementation illustrates two methods for partitioning work to 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. 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 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 copy 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.

The WiMAX receiver has two major modes of operation: startup and steady-state. During the startup process the receiver goes through several states of a state machine until reaches the steady state. The receiver runs through a startup process to achieve synchronization with the transmitter as follows: State 1: Initial energy detection and initial AGC setting, State 2: Coarse carrier frequency offset estimation and correction, State 3: OFDM symbol synchronization via preamble sequence, State 4: Integer frequency offset detection and correction, and State 5: Steady-state processing.

In the steady-state mode, the following functions are performed: (a) I/Q signal decimation and filtering, (b) energy monitoring and AGC fine tuning, (c) I/Q signal derotation, (d) OFDM demodulation via 256 point FFT per OFDM symbol, (e) post-FFT 4*64 preamble detection, (f) symbol timing offset tracking via 4*64 preambles, (g) carrier frequency offset tracking via data symbol pilots, (h) channel estimation via 2*128 preambles, and (i) data symbol processing: channel correction, demapping, deinterleaving, Viterbi decoding, and descrambling. In the implementation, we view steady-state processing as a pipeline. We combine the initial state machine onto one of the stages. The assignment of stages to threads is shown in Fig. 5(b). Overall, the receiver uses 24 threads (3 cores). The state machine is run within one of the threads along with the

FIR/derotation team. Depending on the state transition, data is either passed to the FFT stage (in State 5) or to the thread responsible for the four initial states.

The receiver performance for 2.9 Mbps has been tested according to IEEE802.16 specifications. The targeted receiver SNR was 3.0 dB when using BPSK modulation with 1/2-rate convolutional coding. The measured receiver SNR was 1.59 dB when using 4-bit soft decoding. The simulation has been performed in the Sandblaster simulator. The Sandblaster SB3010 chip is sufficient for a complete ANSI C implementation of the entire physical layer processing. All results have been validated on the hardware development board including complete RF and baseband processing.

7 Conclusion

We have presented a real-time implementation of 2.9Mbps WiMAX on the Sandblaster SDR platform. Our work demonstrates that a software implementation of WiMAX, suitable for mobile applications can be achieved on the same platform along with other communication protocols [10, 11].

References

1. IEEE: IEEE standard for local and metropolitan area networks Part 16: Air interface for fixed broadband wireless access systems. Std. 802.16 (2004)
2. Intersil: Tutorial on basic link budget analysis. App. Note AN9804.1 (1998)
3. Miller, L.E.: LinkCalc: NIST link budget calculator. Technical report, National Institute of Standards and Technology, (Gaithersburg, MA)
4. Miller, L.E.: General purpose propagation loss calculator propagation models: CCIR-Hata Walfisch-Ikegami (WIM). Technical report, Wireless Communication Technologies Group, National Institute of Standards and Technology, (Gaithersburg, MA)
5. Erceg, V., Hari, K.V.S., Smith, M.S., Baum, D.S., Sheikh, K.P., Tappenden, C., Costa, J.M., Bushue, C., Sarajedini, A., Schwartz, R., Branlund, D., Kaitz, S., Trinkwon, D.: Channel models for fixed wireless applications. IEEE 802.16a WG document 802.16.3c-01/29r4, IEEE (2001)
6. Glossner, J., Mougdill, M., Iancu, D., Nacer, G., Jintukar, S., Stanley, S., Samori, M., Raja, T., Schulte, M.: The Sandbridge Sandblaster convergence platform (2005)
7. Nichols, B., Buttlar, D., Proulx-Farrell, J.: Pthreads Programming: A POSIX Standard for Better Multiprocessing. 1st edn. O'Reilly & Associates, Sebastopol, CA (1996)
8. Kotlyar, V., Moudgill, M.: Detecting overflow detection. In: Proc. IEEE/ACM/IFIP Int. Conf. Hardware/Software Codesign and System Synthesis, Stockholm, Sweden (2004) 36–41
9. Glossner, J., Dorward, S., Jinturkar, S., Moudgill, M., Hokenek, E., Schulte, M., Vassiliadis, S.: Sandbridge software tools. In: Proc. Int. Workshop Systems, Architectures, Modeling, and Simulation, Samos, Greece (2003) 142–148
10. Glossner, J., Iancu, D., Lu, J., Hokenek, E., , Moudgill, M.: Software-defined communications baseband design. IEEE Communications Magazine **41** (2003) 120–128
11. Glossner, J., Iancu, D., Nacer, G., Stanley, S., Hokenek, E., Moudgill, M.: Multiple communication protocols for software defined radio. In: Proc. IEE Colloquium on DSP Enable Radio, Livingston, Scotland (2003) 227–236