We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

7

Design and Implementation WiMAX Transceiver on Multicore Platform

Qing Wang[1], Yonghua Lin[1], Jianwen Chen[1],
Da Fan[1,2] and Ling Shao[1]
1. IBM *China Research Lab, 2. Tsinghua University*
Beijing, P.RChina, 100193

Before we design the WiMAX Base Station (BS), three questions we should answered. The first is what kind of system parameters will be selected? The second is which platform will be used for the BS design. And the third one is which algorithm will be selected for some modules that are not defined in 16e standard, especially for the receiver modules, such as synchronization, channel estimation, and STC (Space Time Coding) decoder, etc. When all the above questions obtain proper answers, we can start the BS design and implementation on specific platform to achieve aimed system performance. In this chapter, we will focus on the PHY (physical Layer) design of WiMAX BS.

1. System Parameters

The setting of system parameters depend on the system requirements, such as the system bandwidth, the coverage radius and the users supported, etc.. WiMAX forum proposes some profiles for system specifications' setting [1]. For the WiMAX BS we designed, the system profile is defined as Table 1. We only set the profile under UL_PUSC zone as an example. That is we implement the BS receiver and MS transmitter to illustrate the design process.

Parameters	Values
System Parameters	
System Frequency Band	3.3GHz
System Channel Bandwidth (BW in MHz)	10
Sampling Frequency (Fs in MHz)	11.2 (n=28/25)
Channel Model	SUI3 and AWGN
OFDMA Symbol Parameters	
FFT Size (NFFT)	1024
Sub-Carrier Frequency Spacing (kHz) (Fs/NFFT)	10.94
Useful Symbol Time (Tb = 1/f)	91.4 µs
Guard Time (Tg =Tb/8)	11.4 µs
OFDMA Symbol Duration (Ts = Tb + Tg)	102.9 µs

UL PUSC	One Slot	1 subchannel 3 OFDMA symbols
	Null Subcarriers	184 (92 left, 91 right, 1 DC)
	Pilot Subcarriers (symbol)	[420; 0; 420]
	Data Subcarriers (symbol)	[420; 840; 420]
	Subchannels	35
	Pilot Subcarriers (subchannel)	[12; 0; 12]
	Data Subcarriers (subchannel)	[12; 24; 12]
	Number of tiles	210
	Subcarriers per tile	4
	Tiles per subchannel	6
Link Parameters		
Link Direction		uplink
Frame duration (T_frame)		5 ms
Number of Frames (per second)		200
Number of OFDMA Symbols in one frame		48
Channel Coding		
FEC		CC: 1/2
Modulation		16 QAM
Sub-Channels		Full
STBC Parameters		
Tx number		1&2, single channel, STC Matrix A and Matrix B
Rx number		1&2, , single channel, STC Matrix A and Matrix B
STC Decoding		SFBC

Table 1. Parameters of WiMAX BS PHY (base band without Ranging, UL_PUSC Zone)

2. Platform Selection

In traditional, the base station (BS) market is dominated by the proprietary hardware platforms, which are composed of DSPs and FPGA, or CPU and DSPs. Although there are ASIC solutions for BS applications, most of them are used in picocell or microcell. For the macrocell application, processing capability is always a key issue that should be considered during the BS platform selection. For the WiMAX base station (BS) design, the traditional platforms can be classified as [2]:

- CPU+DSP array+FPGAs

  The CPU is used for control and system management. It can be a simple processor, such as ARM or MIPS chip. DSP array is responsible for complex calculation and FPGAs for acceleration. Currently, most of BS are based on this kind of platform. The advantages of this kind of platform are low power consumption, good development ecosystem, and rich libraries and tools. For example, TI and ADI can provide high performance DSPs for wireless communication. And Xilinx and Altera have their own solution for WiMAX. The disadvantages of the platform are less scalability and flexibility.

- CPU+ASIC
  The solution based on ASIC is easy for development and implementation. However, cost should be considered and the system is difficult to upgrade due to the appropriative chip.
- Customerized Chipset
  It's proprietary platform, generally with high cost, less flexibility and interoperability.

Processing capability, power consumption and cost are the three main factors when considering a BS design. From the traditional view, general IT (information technology) platform, such as blade servers, is not suitable for wireless BS applications. One reason is the processing capability. Compared with DSP or FPGA, the general purpose processor (GPP) has lower performance for complex computation. The other main reason is the power consumption. However, the situation has changed for the occurrence of multicore processor. With the specified architecture, specially designed instruction set, and optimized compiler, the general purpose processor with multicore has powerful processing capability in many applications, including wireless application. For example, Cell Broadband Engine (BE) has 256GFlops processing capability at 3.2GHz [3]. And the Integrated Performance Primitives (IPP) provided by Intel (multicore ready) achieves good performance for multimedia and data processing applications [4]. Considering the total power consumption of the system, the power consumed by baseband is a small portion. Therefore, the general IT platform with multicore or multithread is a good candidate for open wireless architecture (OWA) for its powerful processing capability, flexibility, scalability and interoperability.
In this Chapter, we select the Cell BE as the platform to design and implement the WiMAX BS (base band).

3. System Structure and Functions

3.1 System Structure

The system structure of the proposed BS transceiver (baseband) is described as Fig. 1 [2].
Fig. 2 depicts the uplink subframe with partially used subchannel (PUSC) subcarrier assignment scheme. The uplink subframe is shown in Fig. 2a. Each uplink transmitter is assigned several subchannels to transmit its burst in a timedivision- multiplexing (TDM) manner. Note that the ranging subchanel is not considered in our design. The uplink supports 35 subchannels where each transmission uses 48 data carriers as the minimal block of processing. A slot in the uplink is composed of three symbols and one subchannel, within each slot, there are 48 data subcarriers and 24 fixed-location pilot subcarrier. Multiple subchannels (slots) can be allocated to each user, with one subchannel being the minimum resource that can be allocated to a user. In the frequency domain, a subchannel is constructed from six uplink tiles, each tile has four successive active subcarriers and modulated with a mix of data and pilots over three OFDMA symbols. The configuration of a tile is illustrated in Fig. 2b. Here, we only consider one user and the user occupies the entire system bandwidth.

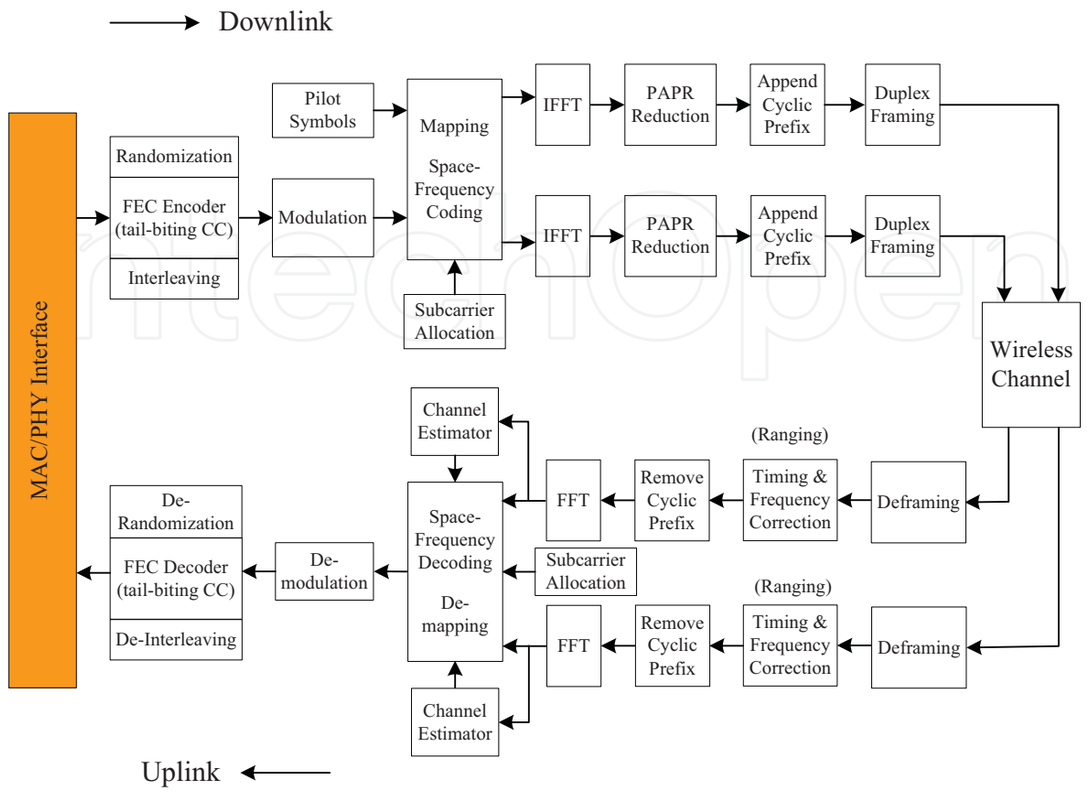

Fig. 1. WiMAX BS PHY System Structure (baseband)

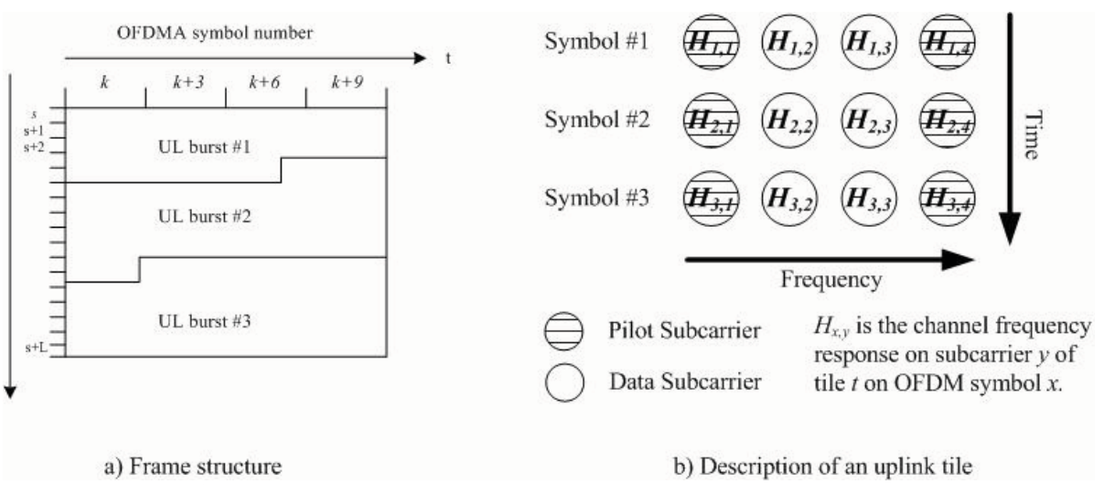

Fig. 2. Frame and Tile Structure

3.2 Signal Model

In the downlink shown in Fig.1, after FEC (Forward Error Control) coding, modulation, zone permutation, OFDMA modulation and cyclic prefix (CP) insertion, the time-domain samples of an OFDM symbol can be obtained from frequency-domain symbols as

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j2\pi nk/N} \qquad -N_{CP} \le n \le N-1 \tag{1}$$

where X (k) is the modulated data on the kth subcarrier of one OFDM symbol, N is the number of subcarriers and N_{CP} is the length of cyclic prefix.
The impulse response of multi-path channel can be approximately denoted as:

$$h(\tau,t) = \sum_{l=1}^{L} \alpha_l(t)\delta(\tau - \eta) \tag{2}$$

where L is the total number of paths, and α_l and τ_l are the complex gain and time delay of the lth path. It supposes that the signals are transmitted over a quasi-static multipath fading channel, that is to say, the channel varies much slowly and the fading coefficients can be assumed to be constant during the OFDM block [5].
Assuming perfect time and frequency synchronization, the model of received signal at the BS after removal of the CP can be written as

$$y(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} H(k)X(k)e^{j2\pi nk/N} + z(n) \qquad 0 \le n \le N-1 \tag{3}$$

where H(k) is the channel frequency response at the kth subcarrier and z(n) is the additive white complex Gaussian noise (AWCGN).

3.3 Algorithm Selections

For the transmitter of WiMAX PHY, the algorithm of each module, such as FEC coding, modulation, map constellation, etc., is mature relatively. Thus we focus on the discussion of algorithm selection for receiver block in this section.

3.3.1. Synchronization

Timing and frequency synchronization are two important tasks needed to be performed by the receiver. Through the timing and frequency offset estimation and correction, the effects of ISI (inter symbol interference) and ICI (inter-carrier interference) can be reduced.
In the presence of symbol timing offset (STO) and carrier frequency offset (CFO), equation (3) should be modified as follows:

$$y(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} H(k)X(k)e^{\frac{j2\pi k(n-\theta)}{N}} e^{\frac{j2\pi \varepsilon n}{N}} + z(n) \qquad 0 \le n \le N-1$$

where θ is the normalized STO with respect to the sample duration, and ε denotes the normalized CFO as a fraction of the intercarrier spacing.
A number of approaches to estimate timing and frequency offset in OFDM systems have been presented in the literature. Some operate in the time domain [6][7], while others use the cycle prefix or the cyclostationarity of OFDM transmissions (e.g., Van de Beek algorithm [8])

to gain information about the symbol timing and frequency offset. As theWiMAX standard, the preamble in OFDMA-mode does not have the repeating pattern similar to that in OFDM-mode. And only uplink subframe is considered in our design. Therefore, in this paper, ML algorithm based on the CP [8] is chosen to achieve the symbol timing and carrier frequency synchronization.

Through the algorithm introduced in [8], we can obtain the estimation of μ and " according to the following two equations:

$$\hat{\theta}_{ML} = arg\max_{\theta}\{|\gamma(\theta)| - \phi(\theta)\} \tag{4}$$

$$\hat{\varepsilon}_{ML} = -\frac{1}{2\pi}\gamma(\hat{\theta}_{ML}) \tag{5}$$

where $\gamma(m)$ is a sum of L consecutive correlations between pairs of samples spaced N samples apart. The term $\phi(m)$ is an energy term, independent of the frequency offset ε .

Once the STO and CFO are estimated, the received time samples can be corrected as follows:

$$y(n)_{corrected} = y(n + \hat{\theta}_{ML})e^{\frac{-j2\pi n\hat{\varepsilon}_{ML}}{N}} \tag{6}$$

3.3.2. Channel Estimation

It is well known that it is necessary to remove the amplitude and phase shift caused by the channel.

Based on the uplink tile structure, shown as Fig.2b, the pilot-aided channel estimation methods can be employed, which consist of algorithms to estimate the channel at pilot frequencies and to interpolate the channel. The estimation of the channel at the pilot frequencies can be based on least square (LS), minimum mean-square (MMSE) or least mean-square (LMS). Though MMSE has been shown to perform much better than LS, it needs knowledge of the channel statistics and the operating SNR [9]. The interpolation of the channel can depend on linear interpolation, second order interpolation, low-pass interpolation, spline cubic interpolation, and time domain interpolation. Considering the tradeoff between feasibility of implementation and system performance, we choose linear interpolation in time and frequency on a tile-by- tile basis for each subchannel.

When the data and pilot information has been assembled as shown in Fig. 2b, it is possible to calculate H_{11}, H_{14}, H_{31} and H_{34} using the equation:

$$\hat{H}_p(t,m) = \frac{Y_p(t,m)}{S_p(t,m)} \tag{7}$$

for the mth OFDMA symbol of the tth tile where:

$Y_p(t,m)$ is the pth received pilot subcarrier

$S_p(t,m)$ is the pth transmitted pilot subcarrier.

We omit the index of receive antenna here, since channel estimation for each receive antenna is performed independently. Subsequently, frequency domain linear interpolation is performed to calculate channel estimates using the following equations:

$$\hat{H}_{1,2}=\frac{1}{3}(\hat{H}_{1,4}-\hat{H}_{1,1})+\hat{H}_{1,1} \qquad \hat{H}_{1,3}=\frac{2}{3}(\hat{H}_{1,4}-\hat{H}_{1,1})+\hat{H}_{1,1}$$
$$\hat{H}_{3,2}=\frac{1}{3}(\hat{H}_{3,4}-\hat{H}_{3,1})+\hat{H}_{3,1} \qquad \hat{H}_{3,3}=\frac{2}{3}(\hat{H}_{3,4}-\hat{H}_{3,1})+\hat{H}_{3,1} \tag{8}$$

where $H_{m,k}$ is the channel frequency response at the kth subcarrier of the mth OFDM symbol and $\hat{H}_{m,k}$ Hm k is the estimation of $H_{m,k}$.
Finally, time domain linear interpolation is achieved as follows:

$$\hat{H}_{2,1}=\frac{1}{2}(\hat{H}_{1,1}-\hat{H}_{3,1}) \qquad \hat{H}_{2,2}=\frac{1}{2}(\hat{H}_{1,2}-\hat{H}_{3,2})$$
$$\hat{H}_{2,3}=\frac{1}{2}(\hat{H}_{1,3}-\hat{H}_{3,3}) \qquad \hat{H}_{2,4}=\frac{1}{2}(\hat{H}_{1,4}-\hat{H}_{3,4}) \tag{9}$$

When all of the channel estimates have been formed, these estimated values are transmitted to the space-frequency decoding module for the data detection using ML method.

3.3.3. SFBC

A user-supporting transmission using transmit diversity configuration in the uplink, shall use a modified uplink tile. The pilots in each tile shall be split between the two antennas and the data subcarriers shall be encoded in pairs after constellation mapping, as depicted in Fig. 3. Because this is applied in the frequency domain (OFDM carriers) rather than in the time domain (OFDM symbols), we note it as space-frequency block coding (SFBC) [10].
Defined $H_{m,k}^{(i,j)}$ as the channel frequency response at the k th subcarrier of the m th OFDM symbol corresponding to the i th transmit and the j th receive antenna pairs, and $Z_{m,k}^{j}$ as the frequency response of the AWCGN on the k th subcarrier of the m th OFDM symbol at antenna j respectively, on the assumption that the neighboring subcarriers have the same frequency response, the estimation of X_1 and X_2 are:

$$\hat{X}_1=\arg\min_{\hat{X}_1\in X}\left[\left(\sum_{i=1}^{2}\sum_{j=1}^{2}\left|\hat{H}_{1,2(3)}^{(i,j)}\right|^2-1\right)\left|\hat{X}_1\right|^2+d^2(\tilde{X}_1,\hat{X}_1)\right]$$
$$\hat{X}_2=\arg\min_{\hat{X}_2\in X}\left[\left(\sum_{i=1}^{2}\sum_{j=1}^{2}\left|\hat{H}_{1,2(3)}^{(i,j)}\right|^2-1\right)\left|\hat{X}_2\right|^2+d^2(\tilde{X}_2,\hat{X}_2)\right] \tag{10}$$

where

$$\tilde{X}_1=\sum_{i=1}^{2}\sum_{j=1}^{2}\left|\hat{H}_{1,2(3)}^{(i,j)}\right|^2X_1+\sum_{j=1}^{2}\left(\hat{H}_{1,2}^{(1,j)}\right)^*Z_{1,2}^{(j)}+\sum_{j=1}^{2}\hat{H}_{1,3}^{(2,j)}\left(Z_{1,3}^{(j)}\right)^*$$
$$\tilde{X}_2=\sum_{i=1}^{2}\sum_{j=1}^{2}\left|\hat{H}_{1,2(3)}^{(i,j)}\right|^2X_2+\sum_{j=1}^{2}\hat{H}_{1,2}^{(2,j)}\left(Z_{1,2}^{(j)}\right)^*+\sum_{j=1}^{2}\left(\hat{H}_{1,3}^{(1,j)}\right)^*Z_{1,3}^{(j)} \tag{11}$$

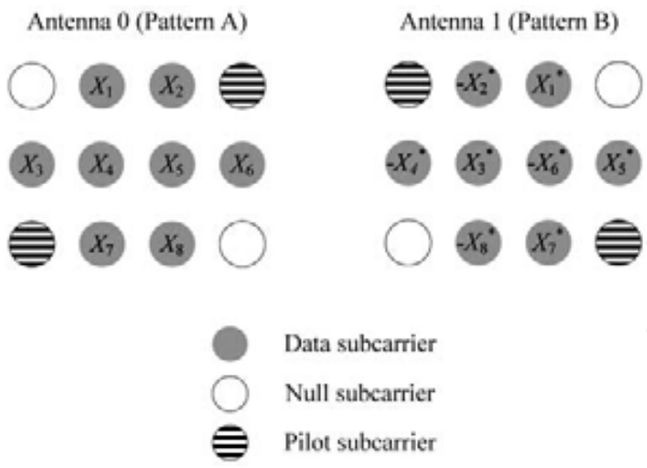

Fig. 3. Pilots and Data Subcarriers in SFBC Mode

4. Implementation on Cell BE

4.1 Cell Processor

Cell processor is proposed and designed as the engine of the PlayStation 3 of Sony initially. But as a powerful, all-purpose multiprocessor, Cell can be expect to be much potential in other areas. A single chip Cell processor contains one PowerPC Processor Element (PPE) and eight Synergistic Processor Elements (SPE). The PPE unit on Cell is a general purpose 64-bit RISC core with 2-way hardware multithreading, used for operating systems and system control, and 8 SPE cores are optimized for compute-intensive, single-precision, floating-point workloads. These units are interconnected with a coherent on-chip element interconnect bus (EIB). The system frequency of Cell is 3.2GHz and the computation capability is 256GFlops [3][11].

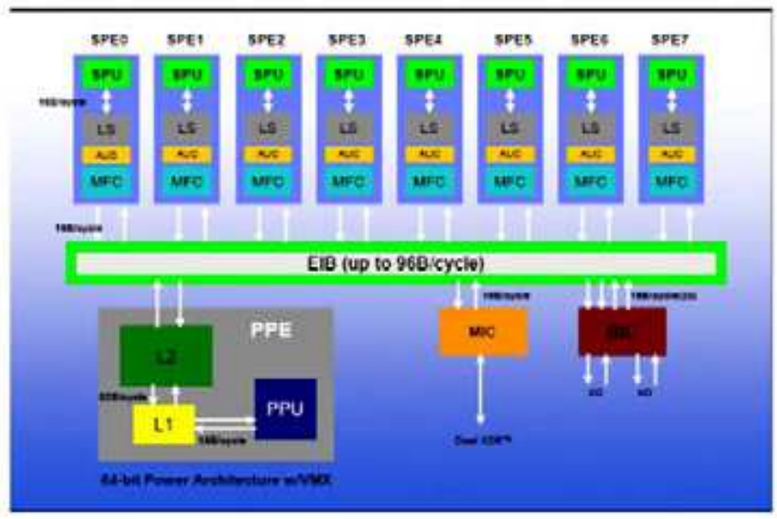

Fig. 4. Block Diagram of Cell Processor and Workload Partition (source: cell diagram from [11])

4.2 Programming on Cell

Cell processor is a kind of heterogeneous multicore processor. Its programming model is novel, see [12] in details. In summary, programming on Cell includes two main points. One is the programming on SPE, especially optimization on SPU since SPE acts as computation accelerator. It has special chip architecture and instruction sets to support such acceleration. The other is the communication between PPE and SPE, and communications among multiple SPEs. In this section, we will focus on the discussion about optimization on SPE (SPU). The communication mechanism of PPE and SPE will be contained in the introduction about software framework design.

For the optimization on Cell, it includes two aspects. One is the processing speed, evaluated by the number of cycle. The other is the local store consuming since each SPU only have 256KB local store. We should make balance between these two factors during optimization. If the computation capability is critical for one component while the buffer and code size are small, we can scarify some local store for achieving high computation performance and vice versa. In our case, for most components, limited local store is more troubled than computation capability. In general, it can solved by good coding design, optimization and local store overlay. Some general optimization techniques on Cell are listed as follows[17][18]:

- Reduce Branch

  Branch can significantly influence the efficiency of the SPU since SPU is an in-order processor with no branch prediction, any judgment will result in the SPU stall. Using the compare-select function instead of short judgment function is a good optimization method for most branches.

- Access Local Store pattern

  The best assess pattern for SPU is data and structure aligned with vector operation. The Scalar and unaligned access will result in many additional instructions for data aligned and scalars extracted from vectors. In some case, we can operate the scalar as the vector. This method solves the data access problem of the SPU which can not be made as SIMD pattern.

- SIMD Accelerating

  SIMD (single instruction multiple data) is a very useful accelerating technique for SPU. It generally has 4—8 times speed-up rate.

- Pipeline and Dual-issue

  Each instruction has its latency and Stall cycles which will influence the efficiency of the SPU due to the dependency. If two conjoint instructions can be placed in the different pipeline with no dependency, the two instructions can be dual-issue.

4.3 Workload Analysis and Optimization

4.3.1 Workload Analysis

From the theoretical analysis, we know the modules of uplink, such as channel decoding, channel estimation and SFBC, consume most of the computation resource. They are the modules with heavy workload. This conclusion is also verified by workload test on Cell. Table 2 shows the workload of each module of uplink for processing 3 OFDMA symbols. The test runs on Cell BE simulator-Mambo with Cell SDK2.1. The cycle numbers of "CP remove" module and "channel estimation" module are for one antenna. The "viterbi" module

is 1/2 data rate and the constraint length is 7. We note that the Viterbi, deinterleave and SFBC are the top three modules with heavy workload. And the other modules, such as channel estimation, derandomize and demodulation modules, do not match the throughput requirement without optimization. Thus we need to optimize those modules to meet the targeted 20Mbps throughput.

Module	Original Cycles	Optimized Cycles	Speed-up Rate
CP remove	93261	10919	8.54
Channel est.	260812	19420	13.43
SFBC	1413688	162179	8.72
Demodulation	267175	41159	6.49
Deinterleave	3622165	154506	23.44
Viterbi	4728180	343227	13.78
Derandomize	278893	8031	34.73

Table 2. Workload for Modules of Receiver

For the modules of downlink, the result of workload testing depicts as Table 3. The test environment and data length are the same as that of uplink. The initial length of data is 3354 bits, containing 3 OFDMA symbols.

Module	Original Cycles	Optimized Cycles	Speed-up Rate
Randomize	278893	6139	45.43
Convolutional coding	190816	79979	2.39
Interleave	3729682	146975	25.38
Modulation (16QAM)	144543	76137	1.90
Zone Permutation	372386	187054	1.99
CP Insert	209997	24551	8.55

Table 3. Workload for Modules of Transmitter

We use convolutional code (data rate =1/2, constraint length = 7) for channel coding and the modulation is 16QAM. Except the interleave module, the other modules of downlink have the same level workload before optimization. Compared with the workload of uplink, the modules of downlink consume less computation resource. For the FFT and IFFT used for system, we will use the library provided by Cell SDK. There is no optimization work on these two modules. Hence we did not list their workloads here.

4.3.2 Workload Optimization

Based on the workload analysis, we optimize each module to meet the throughput requirement we pre-set. That is 20Mbps processing capability for both downlink and uplink. In our application, each technique mentioned above is used and the speed-up rate of each module is shown in Table 2 and Table 3 for uplink and downlink respectively.
During the optimization, we should tradeoff between computation performance (cycles) and

local store consumption. For the computation critical module, such as Viterbi decoding, we will scarify the local store to obtain the smaller cycles; While for the local store critical module, we will try to save buffers instead of achieving highest performance. Therefore, when we refer to the performance of each module, we should consider both the number of cycles and the consuming of local store, which is very important for workload partition on different SPEs. The optimized results shown in Table 2 and Table 3 are not the best one. We just optimize them till they can meet our design requirements. They still have potential optimization space.

Based on the optimization results and local store consumptions, the workload can be partitioned to five SPEs, in which two SPEs for downlink and three SPEs for uplink. PPE is responsible for SPE control and management. So one Cell BE chip can process both uplink and downlink with 20Mbps throughput in theory. Figure 5 depicts the workload partition of Cell.

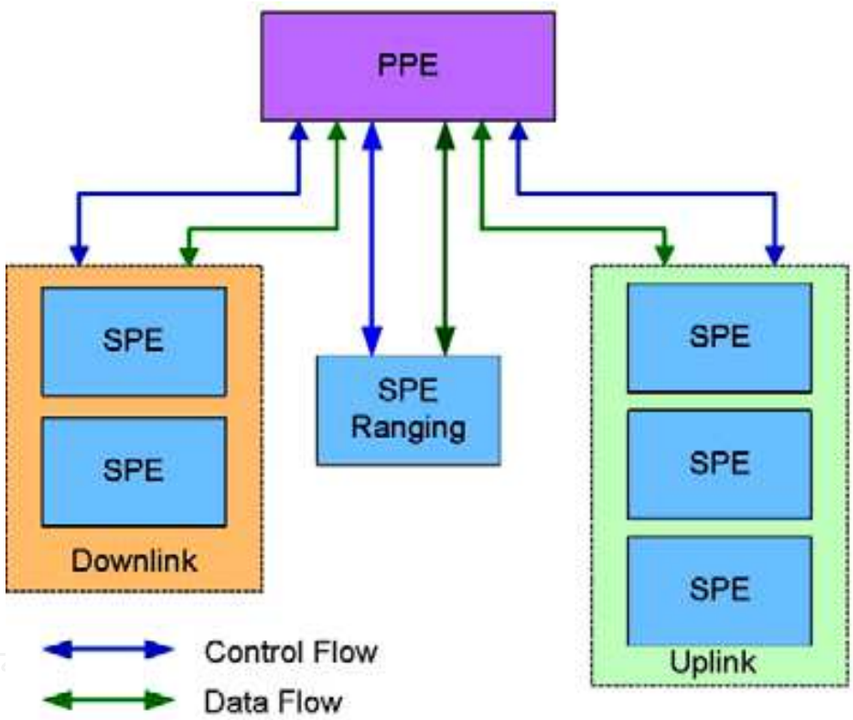

Fig. 5. Workload Partition of Cell

4.4 Framework Design

For the software framework design, we consider three scenarios: sequential framework, PPU synchronization and SPU synchronization. In the sequential framework, PPU is used as a controller for SPU control and data management. SPU is used for data processing. The data is stored in the main memory. SPU fetches the data from the main memory, calculates them and then sends the computation results back. Sequential framework is the simplest one with low efficiency. It can not satisfy the 20Mbps throughput requirement. Therefore,

we only use this framework to verify the system correctness at the beginning of system integration. For the PPU synchronization framework, PPU is used to manage the synchronization of SPUs. This results in the PPU to take heavy workload. If the system (Cell blade server, named as QS20, containing two Cell Processor) wants to support 3 sectors, PPU becomes the bottleneck of system. Therefore, we do not adopt this framework. SPU synchronization is the framework we used in the current system, shown as Fig. 6.
In this design, different modules will work in parallel. SPUs will manage their synchronization through messages passing. Since there is no feedback path in the data flow of both uplink and downlink, pipeline can be used in the framework design. There are two different levels of pipeline:

- SPU Level Pipelining. This level pipelining can be realized by double the input and output buffers.The double buffers are allocated on main memory.
- Functional Level Pipelining. The functional units in one SPU can also work in pipelining, but it is heavily dependent on the algorithms and local store limitation. Only when the local store can support double buffer for both input and output, the pipelining can be used. Functional level pipelining can overlap the time consumption of DMA tasks and computation tasks.

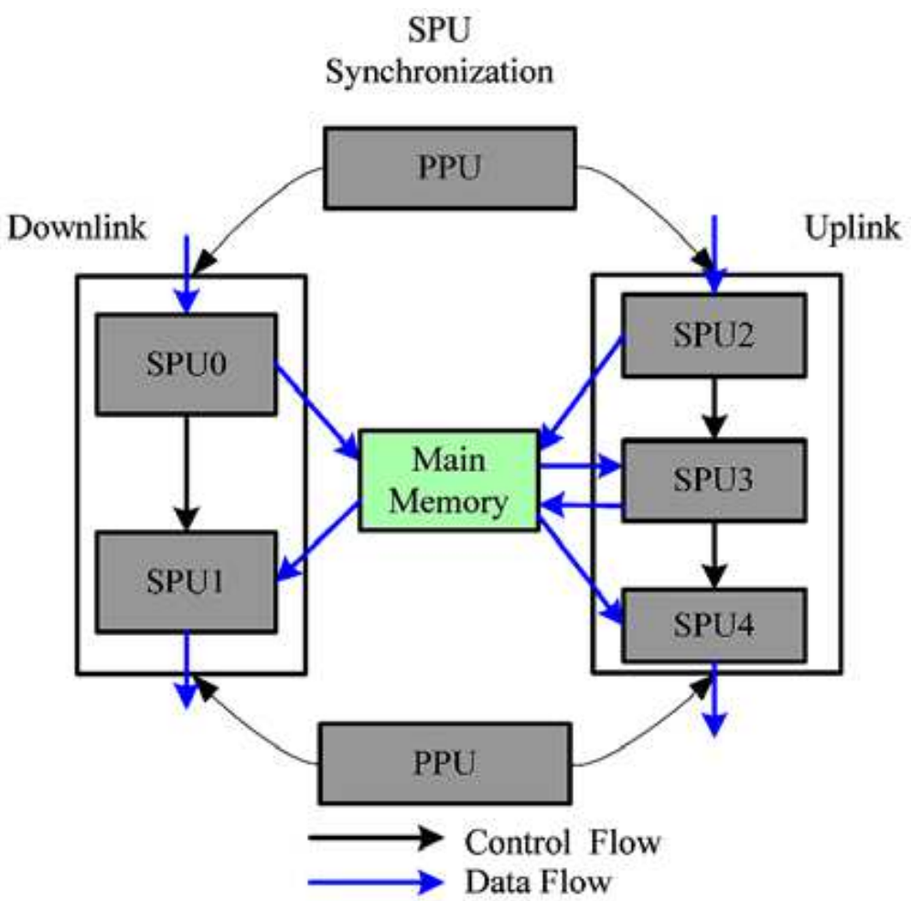

Fig. 6. Software Framework for One Sector

5. Simulation Results and System Performance

The system is implemented on IBM Cell blade server, named QS-20, which has two Cell B.E. processors (a 2-way SMP) operating at 3.2 GHz. We use $2R_x$ X $2T_x$ MIMO technique and the system parameters are set as Table 1. The uplink bandwidth is 10MHz, the subcarrier frequency spacing Δf is 10.94kHz, N=1024, and N_{CP} =128. The following parameters are also assumed: 1/2 convolutional coding with constraint length of 7 and generator polynomial matrix of [133 171]. A discrete channel model based on the Stanford University Interim 3 (SUI-3) [13] model is used, which represents a low delay spread case with $\tau_{rms} = 0.264\mu s$ (low frequency selectivity). The bit-error rate (BER) performance is evaluated by averaging over 200 frames, and each frame has 3 OFDMA symbols. Figure 7 is the simulation results at different stages of the system level simulator.

We evaluate the system performance from two aspects. One is the throughput of uplink and downlink, the other is the system BER. The throughput demonstrates the system processing capability. Table 4 shows the throughput test results. Each sector can achieve 20Mbps throughput whether for downlink or uplink. The total throughput of one QS20 will exceed 60Mbps.

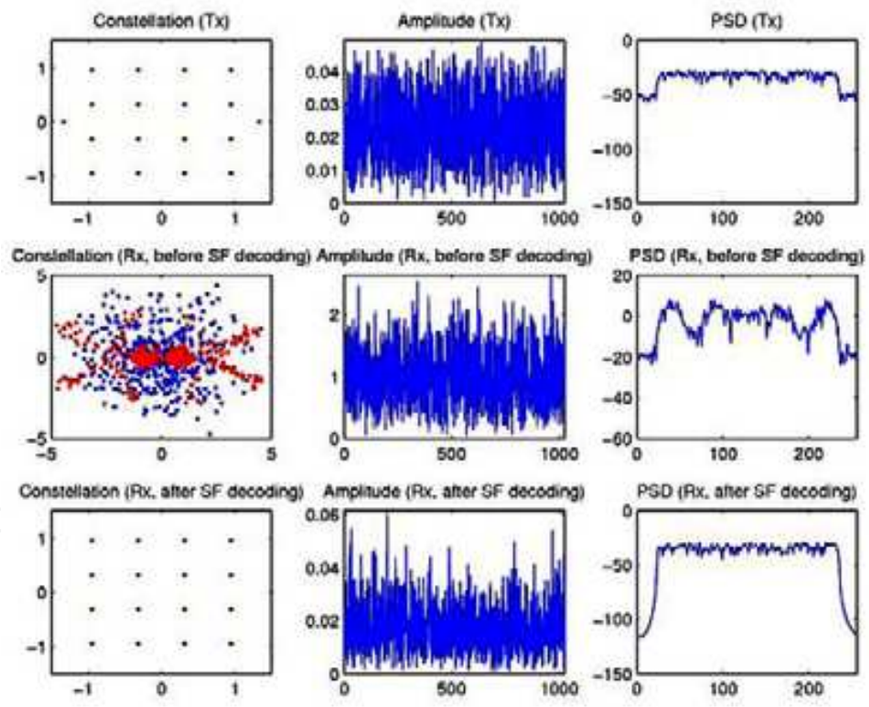

Fig. 7. Simulation Results at Different Stages of the System Level Simulator

Throughput	Downlink (Mbps)	Uplink (Mbps)
Sector1	24.409414	20.970757
Sector2	25.042559	21.517656
Sector3	24.442323	21.473296

Table 4. Throughput of Three Sectors on One QS20

BER performance reflects the correctness of system design and the system precision. Figure 8 is the BER results tested on QS20 and X86 processor (Intel Xeron@2.8GHz) respectively. We tested both AWGN channel and Rayleigh channel on X86 and Cell platform. The results indicate that the BER performances are almost the same for X86 platform and cell platform whether under AWGN channel or Rayleigh channel.

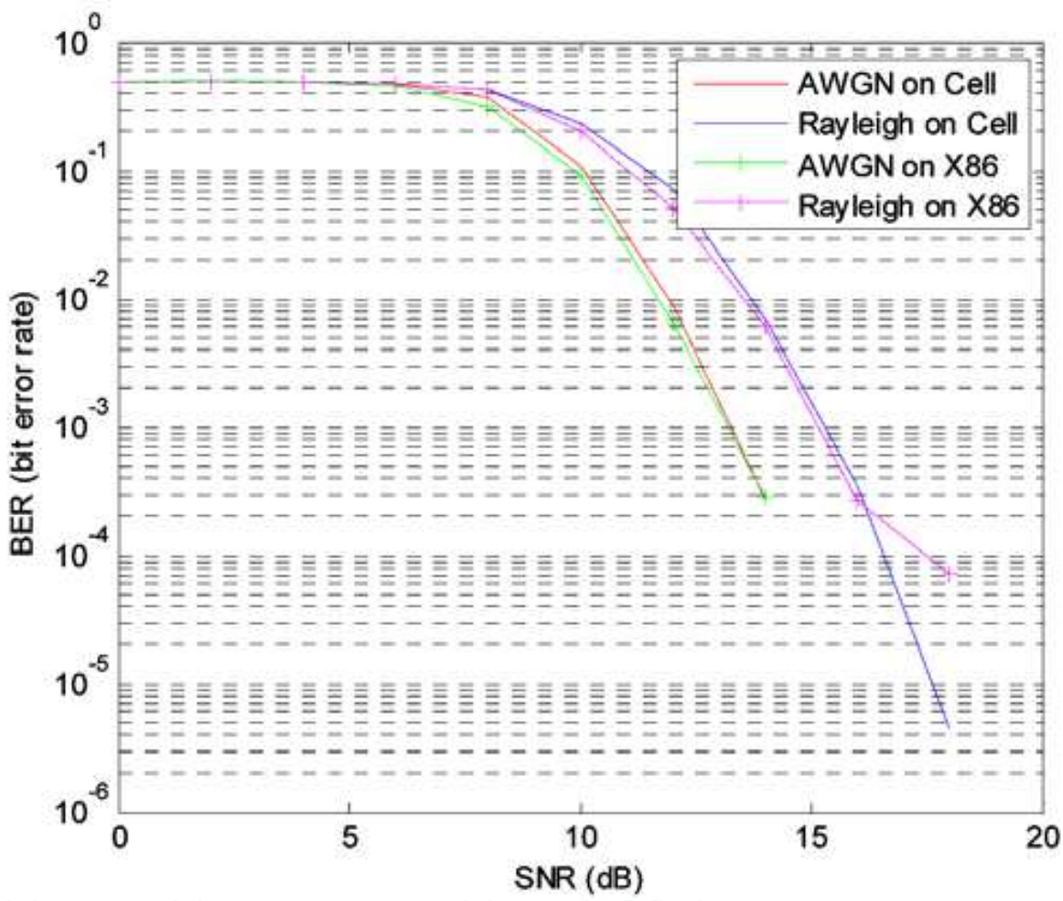

Fig. 8. BER Performance for AWGN and Rayleigh Channel under Different Platforms

6. Summary

In this chapter, we propose the possible solutions for the issues during WiMAX BS implementation, such as the platform selection, algorithm selection, and performance optimization. And we design and implement a WiMAX BS (PHY, baseband) on Cell processor as an example for illustration. The system requirements decide the platform selection, and the system processing capability and system performance requirements are the main factors considered during the BS design. The performance optimization can be classified as individual module optimization and system framework optimization. Both of them heavily depend on system hardware structures. Although different platforms have their specific optimization methods according to the system structures, efficient

communications between each modules and acceleration for some key modules with heavy workloads are general methods that should be considered.

7. Reference

[1] WiMAX Forum™ Mobile System Profile 3-- Release 1.0 Approved Specification (Revision 1.7.1: 2008-11-07), WiMAX Forum.

[2] Qing Wang, Da Fan1, Jianwen Chen, Yonghua Lin and Zhenbo Zhu (2008). WiMAX BS Transceiver Based on Cell Broadband Engine, Proceedings of IEEE International Conference on Circuits & Systems for Communications, May 2008.

[3] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy (2005). Introduction to the cell multiprocessor, IBM Journal of Research and Development, vol. 49, no. 4/5, 2005.

[4] Intel (2003). Performance benchmarks for intel integrated performance primitives-white paper, 2003.

[5] Qing Wang, Da Fan, YongHua Lin, Jianwen Chen, Zhenbo Zhu (2008). DESIGN OF BS TRANSCEIVER FOR IEEE 802.16E OFDMA MODE, Proceedings of ICASSP08, April 2008. [6] P. Moose (1994). A technique for orthogonal frequency division multiplexing frequency offset correction, IEEE Trans. on Communications, vol. 42, pp. 2908-2914, 1994. [7] T. Schmidl and D. Cox (1997). Robust frequency and timing synchronization for ofdm, IEEE Trans. on Communications, vol. 45, pp. 1613-1621, 1997.

[8] M. Sandell J. van de Beek and P. Borjesson (1997). ML estimation of time and frequency offset in ofdm systems, IEEE Trans. Signal Processing, vol. 45, pp. 1800-1805, July 1997. [9] J.-J. van de Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Borjesson (1995). On channel estimation in OFDM systems, Proc. IEEE 45th Vehicular Technology Conf., vol. 45, pp. 815-819, Chicago, IL,July 1995.

[10] K. F. Lee and D. B. Williams (2000). A space-frequency transmitter diversity technique for OFDM systems, Proc. IEEE GLOBECOM, pp. 1473-1477, San Francisco, CA, Nov.2000.

[11] IBM(2006). Cell broadband engine processor based systems-white paper, p. 3, Sep. 2006. [12] IBM(2007). Software Development Kit for Multicore Acceleration Version 3.0--Programming Tutorial.

[13] V. Erceg, K.V.S. Hari, and et al. M.S. Smith (2003). Channel models for fixed wireless applications, Contribution IEEE 802.16a-03/01, Jun. 2003.

[14] IEEE Std 802.16-2004. Part 16: Air Interface for Fixed Broadband Wireless Access Systems, ," Oct. 2004.

[15] IEEE Std 802.16e-2005. Part 16: Air Interface for Fixed, Mobile Broadband Wireless Access Systems Amendment2: Physical, Medium Access Control Layers for Combined Fixed, and Mobile Operation in Licensed Bands., ," Feb. 2006.

[16]Hassan Yagoobi (2004). Scalable OFDMA physical layer in IEEE 802.16 wirelessman, Intel Technology Journal, vol. 8, Aug. 2004.

[17]Daniel A. Brokenshire (2006). Maximizing the power of the Cell broadband engine processor: 25 tips to optimal application performance, IBM developerWorks, June 2006.

[18] IBM(2006). Cell broadband engine programming handbook v1.0, pp. 681-708, April 2006.
[19] M. Hsieh, C. Wei (1998). Channel Estimation for OFDM Systems Based on Comb-type Pilot Arrangement in Frequency Selective Fading Channels, IEEE Trans. Consumer Electron., vol. 44, no. 1, Feb. 1998.

IntechOpen

IntechOpen

WIMAX New Developments

Edited by Upena D Dalal and Y P Kosta

ISBN 978-953-7619-53-4
Hard cover, 442 pages
Publisher InTech
Published online 01, December, 2009
Published in print edition December, 2009

WiMAX (Worldwide Interoperability for Microwave Access) is a wireless broadband access network named by industry group called the WiMAX forum formed in June 2001. It is Wireless MAN with IEEE 802.16 family standards. Loosely, WiMAX is a standardized wireless version of Ethernet that enables the last mile, intended primarily as an alternative to wire technologies (such as Cable Modems, DSL and T1/E1 links) to provide broadband access to customer premises. Mission of the WiMAX forum is to promote and certify compatibility and interoperability of broadband wireless products. This book touches most of the above issues in form of 22 individuals' papers containing research work in WiMAX domain in particular. WiMAX has two important standards/usage models: a fixed usage model IEEE 802.16-2004 for Fixed Wireless Broadband Access (FWBA) and a portable usage model IEEE 802.16e-2005, which is mainly concentrated on Mobile Wireless Broadband Access (MWBA). Both are released as standards and amendments are available in form of drafts. Higher data rate transmissions (@ 100 Mbps) are achieved in IEEE 802.16-2004 WiMAX through LOS communications which incorporate a stationary transmitter and receiver but IEEE 802.16e supporting NLOS communication is much complicated and little less bit rate is achieved. 2-11 GHz licensed band is the range of frequencies with TDD and FDD supports. The book will provide a wide horizon to visualize the WiMAX technology and its developments leading towards 4G systems. It will provide a good platform to the researchers with clues to the innovative ideas in WiMAX domain. I wish all the best to the authors and readers of this book in their successful research of WiMAX technology.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Qing Wang, Yonghua Lin, Jianwen Chen, Da Fan and Ling Shao (2009). Design and Implementation WiMAX Transceiver on Multicore Platform, WIMAX New Developments, Upena D Dalal and Y P Kosta (Ed.), ISBN: 978-953-7619-53-4, InTech, Available from: http://www.intechopen.com/books/wimax-new-developments/design-and-implementation-wimax-transceiver-on-multicore-platform

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820

www.intechopen.com

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821

IntechOpen

IntechOpen

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen