Available online at www.sciencedirect.com
ScienceDirect
Procedia Technology 11 (2013) 680 – 688
Procedia Technology

The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)

Architecture Design of Frequency Domain Processing for Flexible and Re-configurable WiMAX OFDMA Receiver

T. Adiono*, N. Sutisna

School of Electrical Engineering and Informatics, Institut Teknologi Bandung Jalan Ganesha 10. Bandung 40132, Indonesia

Abstract

This paper proposes hardware architecture of WiMAX OFDMA frequency domain processing system. The system mainly consists of channel estimator, equalizer, and subcarrier de-allocator. The system is optimized for flexible and re-configurable WiMAX OFDMA receiver. The flexibility feature is obtained by employing flexible control unit approach using task FIFO. Using this scheme, the designed system can handle various and complex data structure within OFDMA frame. The propesed architecture has been implemented in 0.13 µm CMOS technology. The implementation result shows that chip area is about 0.45 mm^2 and able to work in targeted system clock 54 MHz.

Selection and peer-review under responsibility of the Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia.

Keywords : VLSI architecture; Frequency Domain Processing; WiMAX OFDMA; Estimator; Equalizer

1. Introduction

WiMAX, as defined by 802.16e-2005 standard [1], is one of promising mobile communication technology that provide high data rate. Hence, flexibility and re-configurability are required to adapt with unstable standards and requirements without significant overhead cost and also to enable hardware re-use [2, 3].

The WiMax baseband receiver, as depicted in Fig. 1, generally consists two main data processing, which are time domain processing and frequency domain processing. The time domain processing is related to frame synchronizer and CP remover, while the frequency domain processing related to channel estimation, equalization, de-mapper, etc. The data conversion from time domain to frequency domain is performed by FFT.

* Corresponding author.
Email address: tadiono@dnet.net.id

Selection and peer-review under responsibility of the Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia.
doi:10.1016/j.protcy.2013.12.245

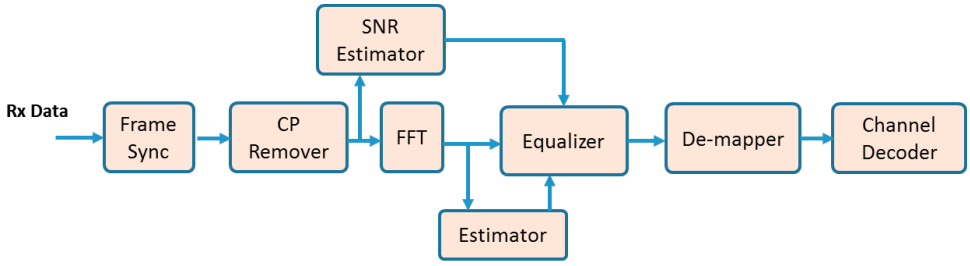

Fig. 1. The Receiver Block Diagram

One of important sub-system in the receiver that most probably modified during development phase is frequency domain processing related modules. Some modification in these blocks should be done due to iterations to find the most suitable method or algorithm that meet performance requirement, for example target bit error ratio (BER) with reasonable hardware cost. Moreover, the algorithm exploration should be perform due to the processing is not defined in standard.

The performance of an OFDM system is greatly influenced by the quality of channel estimation and equalization. Multi-path propagation in wireless transmission channel causes signal amplitude and phase to be changed, depend on the characteristics of channel. This changes lead to decision error in demodulator process. Channel estimator and equalizer will play an important role in OFDMA receiver system. The channel estimator estimates the channel response. Once channel response is obtained, the equalizer will compensate to eliminate signal distortion [4]. Some receiver architectures have been proposed, for example in [5, 6]. However, it did not well clarify the design development from algorithm to architecture.

2. The Architecture of Frequency Domain Data Processing

The frequency domain data processing (rx freq processing) is started with the estimation of channel response, followed by equalization, and demodulation. The calculated channel response will be used for equalization to reconstruct received signal. Further, the received signal will be determined its constellation regarding to mdoulation scheme. The detail processing of each modules and proposed architecture is described as as follow.

2.1. Channel Estimator

The estimator block estimates the channel response by using the received data pilot. As defined in WiMAX OFDMA framing, pilot structure depend on the zone type, which are PUSC and FUSC. For PUSC zone, a symbol consists of 120 pilots, while for FUSC, a symbol consists of 82 pilots. The pilot structure for PUSC zone can be illustrated in Fig. 2.

The estimation process is mainly performed in two steps: time domain estimation (horizontal axis) and frequency domain estimation (vertical axis). In order to carry out interpolation process in estimation, pilot response should be provided. Pilot response could be simply calculated by dividing received pilot and reference pilot, as given in following equation :

$$h_p = \frac{y_p}{x_p} \tag{1}$$

where, y_p is the received pilot and x_p is transmitted pilot (reference pilot). Received pilot can be found by selecting subcarrier data in certain position. It is carried out by De-allocator Block. Meanwhile, the reference pilots are calculated with the same scheme to the transmitter. The pilots response will be used as basis for data interpolation, which are time domain and frequency domain.

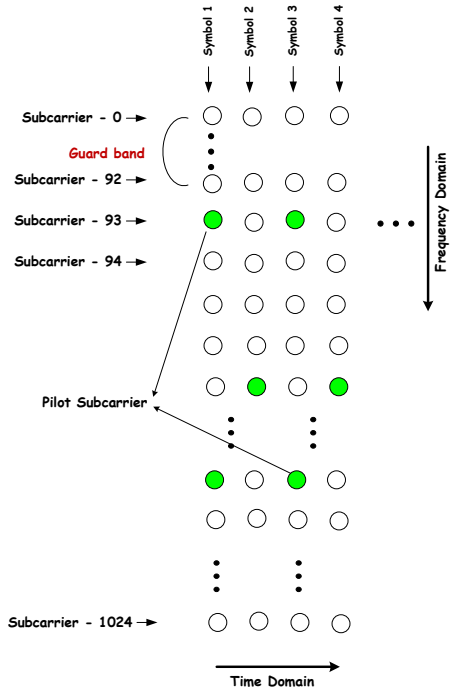

Fig. 2. Pilot Structure for PUSC Type

2.1.1. Time Domain Interpolation

Interpolation on time domain is performed by linear interpolation. The interpolation process will be started from the first row of existing pilot. In initial phase, the interpolation process uses the calculated pilot response, as descried in previous section. As depicted in Fig. 2, the estimated data response should be calculated as linear interpolation from two adjacent pilots. For example, data response for location (1, 2), (1, 4), (13, 2), (13, 4) are calculated as given in following equations.

$$\begin{aligned} h\_int(1,2) &= \{h\_int(1,1)+h\_int(1,3)\}/2 \\ h\_int(1,4) &= \{h\_int(1,3)+h\_int(1,5)\}/2 \\ h\_int(13,2) &= \{h\_int(13,1)+h\_int(13,3)\}/2 \\ h\_int(13,4) &= \{h\_int(13,3)+h\_int(13,5)\}/2 \end{aligned} \tag{2}$$

In case of location of data estimation is in the most left, that lead the linear interpolation from two adjacent pilot could not performed, data response in this location is directly copied from the next pilot response. In this example, the data response for location (5, 1), and (9, 1) are dictated to plot response of location (5, 2) and (9, 2), respectively.

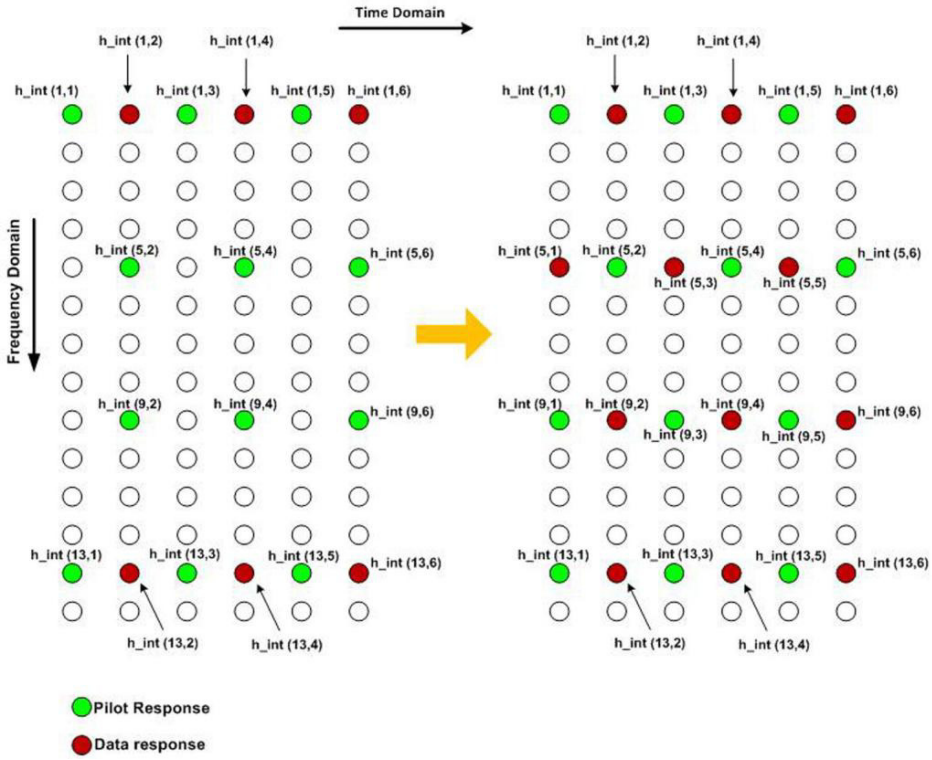

Fig. 3. The Time Domain Interpolation Process

To implement such calculation, it only requires an adder and shifter. The block diagram for time interpolation is depicted in following figure.

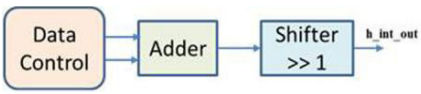

Fig. 4. The Architecture for Time Interpolation

2.1.2. The Frequency Domain Interpolation

Interpolation in frequency domain performed by using provided data from time domain interpolation. This interpolation is performed in row order. Due to row index represent frequency index, this interpolation we called frequency domain interpolation. Referring to Fig. 3 there are still three remaining rows that should be estimated to find out all of data response. The interpolation for each row is performed by weighted-linear interpolation from two available estimated responses. In general, the interpolation in each row could be represented by following equation.

$$h\_int = [h\_int2 - h\_int1] * (k) + h\_int1, \;\; k = 1,2,3 \tag{3}$$

where, *h_int2* and *h_int1* is the data response for the two available rows, *k* is weighted index for interpolation. The resulted from this interpolation step is depicted in following figure. After the data response is available, it shoukld be deliever to next block, whic is equalizer.

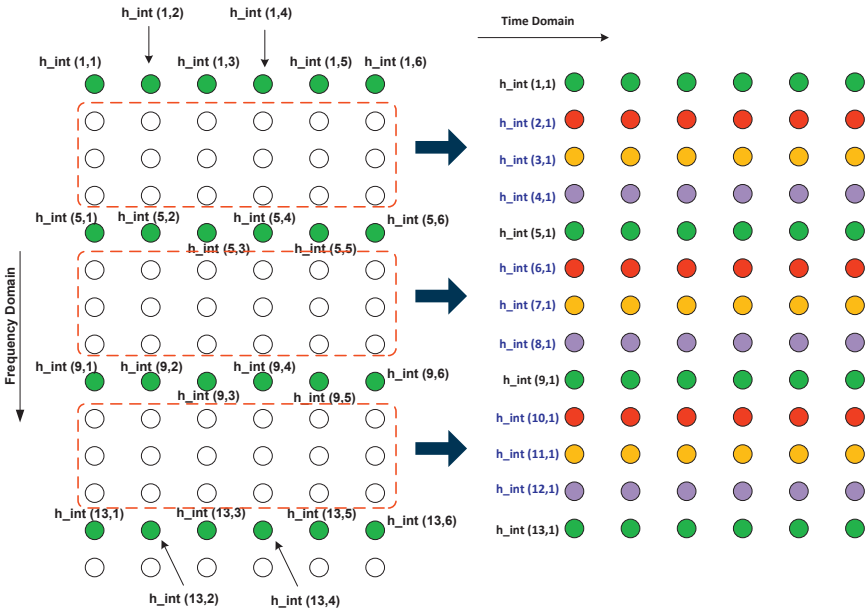

Fig. 5. The Frequency Interpolation Process

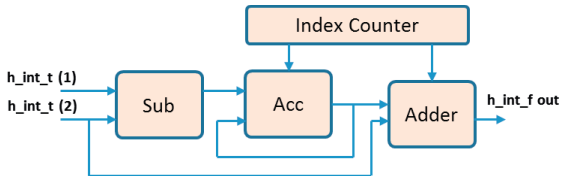

Fig. 6. The Architecture for Frequency Interpolation

The datapath architecture for estimation process is depicted in following figure.

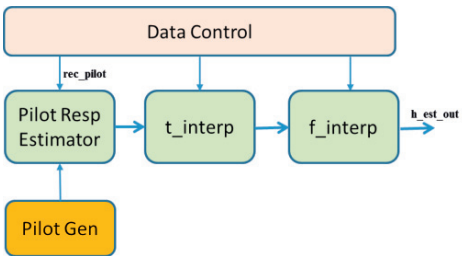

Fig. 7. The Complete Architecture of Estimator

2.2. Equalizer

Data equalization is performed to each data subcarrier using provided channel response and estimated noise power. In order to maintain performance, the MMSE equalizer is selected instead of Zero Forcing algorithm. For MMSE Equalizer, th SNR of received signal should be consider. Assuming that received signal follow this equation :

$$Y = HX + N \tag{4}$$

where, Y is receievd signal, H is channel response, X transmitted signal, and N is some noise induced during transmission.

The corrected signal $\bar{X}$ is calculated by multiplying the equalizer coefficient with received signal that minimizes square error between estimated signal and transmitted signal as described in (5). Meanwhile, equalizer coefficient is consider as one-tap equalizer due to hardware complexity reason. The equalizer coeffcient is expressed in Eq. 6.

$$\overline{X} = GY \rightarrow minimize\ E\left\{\left|\overline{X} - X\right|^2\right\} \tag{5}$$

$$G = \frac{H^*}{\left|H\right|^2 + NP} \tag{6}$$

The final equation for data equalization is provided in following equation.

$$\overline{X} = \frac{H^*}{\left|H\right|^2 + NP} Y \tag{7}$$

Eq. 7 could be implemented by using two complex multiplier, one divider, and one adder.

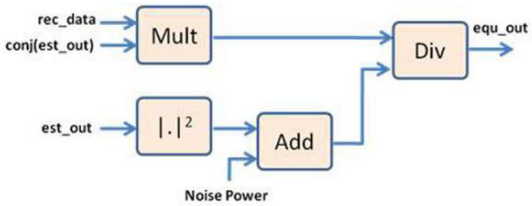

Fig. 8. The Architecture for Equalizer

2.3. Noise Power Estimator

Noise power is required for MMSE equalization. One of method for noise power estimation is estimating based on Null subcarrier (Guard Band). This method is very simple while performance still considered. The noise power can be estimated by averaging power of noise in guard band of each symbol as provided in following equation.

$$avg(A) = \frac{1}{N}\sum_{k=1}^{N} A_k \tag{8}$$

Whereas, guard band are subcarrier no 1-92 and 394-1024, giving the value of N is 183. Hence the equation can be rewrite as follow.

$$avg(A) = \frac{1}{183}\sum_{k=1}^{183} A_k = \frac{1}{256}\frac{256}{183}\sum_{k=1}^{183} A_k = \frac{1}{256}K\sum_{k=1}^{183} A_k ; K = 1.398907 \quad (9)$$

Proposed architecture to implement noise averaging is depicted in Fig. 9.

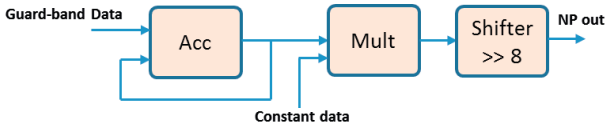

Fig. 9. The Architecture for Noise Power Estimation

2.4. De-allocator (Address Generator Unit - AGU)

De-allocator is part of control unit that provided appropriate data for channel estimator and equalizer. This block generate RAM address to determine subcarrier index regarding to permutation algorithm, as defined in standard. To obtained RAM address, AGU module work as specified by parameter setting in FIFO task. Using this approach, processing will be flexible and can handle various data structure in one OFDMA frame.

2.5. Main Control Unit (Task FIFO)

The main control unit will provide related signals required by all module to perform its task. Main control unit consist of sevaral setting parameter for each data block (cluster) stored in FIFO. Thus, this setting named as FIFO task. The sequence FIFO Task are:

- In the beginning of frame, the FIFO Task should be for FCH processing.
- The second task, FIFO Task instruct estimator and equalizer to process DL-MAP burst. This task may be consist some information, such as number of slot and symbol length. Parameter setting contained in the second FIFO task should be provided by extracting parameter in FCH field.
- Following FIFO Tasks direct estimator and equalizer to process data burst in remaining frame. The FIFO task will be provided for each data cluster. This FIFO task containing paramemter that previously extracted from DL-MAP field.

FIFO task generation is performed by software executing in host processor to provide data flexibility.

2.6. Top Level Integration

Following figure is the top level integration for all blocks in frequency domain processing. The whole system also consists of some memory block for temporary buffering, such as: cluster RAM for buffering data before estimator processing, Response RAM for storing temporary estimator result, and FFT OUT RAM which is used for storing data from FFT block.

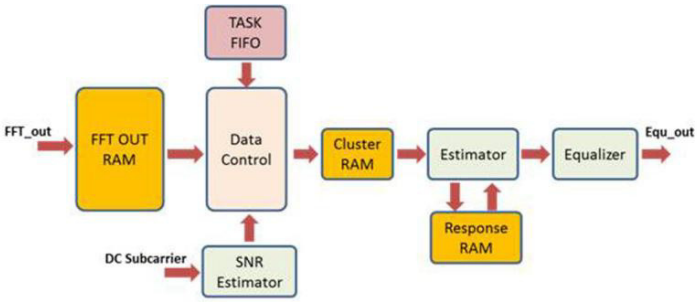

Fig. 10. Top level integration for Frequency Domain Processing

3. Hardware Implementation

Proposed design is implemented in CMOS 0.13 µm as technology target. The designed is intended for 56 MHz clock frequency. Imlemented design shows that area of the designed system is about 0.44 mm^2. Area utilization for each modules is provided in Table 1.

Table 1. Area Utilization of Related Modules

Module Unit	Area (µm2)
Address Generator Unit	60401,71
Data Randomizer	1838,28
Data De-allocator	32822,45
Pilot De-allocator	25740,98
Equalizer (inc. NP Est)	115542,48
Rx Freq Shared-Divider	129179,06
Estimator	37533,39
Pilot Respond	14003,58
Ref Pilot Generator	11114,59
Demodulator	32012,62
FFT Out RAM Ctrl	17914,43
Rx Cluster RAM Ctrl	24547,92
Rx EqFreq RAM Ctrl	5601,42
Total Area	447851,21

4. Conclusion

In this paper, the architecture of Frequency domain processing of WiMAX OFDMA is presented. The proposed architecture provides flexibility and re-configurability features to address complex data processing. The architecture is successfully implemented in CMOS 0.13 μm technology, resulting 0.45 mm^2 chip area and able to work in real-time using system clock of 56 MHz.

References

[1] _____, IEEE 802.16-2005e: Standard for local and metropolitan networks Partl6: Air Interface for Fixed Broadband Wireless Systems, 2005.

[2] A. Nilsson et.al, An 11 mm2, 70 mW Fully Programmable baseband Processor for Mobile WiMAX and DVB-T/H in 0.12 um CMOS, IEEE Journal of Solid-State Circuits, 2009: 44(1).

[3] K. Masselos, S. Blionas, dan T. Rautio, Reconfigurability requirements of Wireless communication systems, IEEE Workshop on Heterogeneous Reconfigurable Systems on Chip, April 2002.

[4] T.-D. Chiueh and P.-Y. Tsai, OFDM Baseband Receiver Design for WirelessCommunications, John Wiley and Sons, 2007.

[5] J.-M. Lin, et al. A Baseband Transceiver for IEEE 802.16 OFDMA Downlink Communications, IEEE Symposium on Advances in Wired and Wireless Communication, 2005.

[6] W.-H Tseng, et al. Digital VLSI OFDM Transceiver Architecture for Wireless SoC Design, IEEE International Symposium on Circuits and Systems (ISCAS), 2005.