A Low Power ZigBee Baseband Processor

Kai-Hsin Chen and Hsi-Pin Ma, *Member, IEEE*
Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.
Email: khchen@larc.ee.nthu.edu.tw, hp@ee.nthu.edu.tw

***Abstract*—This paper presents an IEEE 802.15.4 (ZigBee) practicable baseband processor including transmitter and receiver. To estimate and compensate carrier phase error at baseband, the receiver allows full digital solution for carrier phase synchronization. An existing packet detecion algorithm for spread spectrum comunication system is used to estimate large carrier frequency offset. This paper also presents a new decision-feedback algorithm for residue phase error tracking. The proposed receiver achieves system performance as PER = 0.01 at SNR less than 5 dB, which is better than standard specifications. The baseband processor was implemented in simple hardware architecure and designed with low power technique. The chip is fabricated with the TSMC 0.18μm 1P6M CMOS technology with a gate count of 78 k. The area is 1.633 mm x 1.633 mm, and power consumption is about 1.7 mW at receiver mode under supply voltage of 1.8 V and operating frequency of 4 MHz.**

I. Introduction

Unlike other standards for wireless communications, IEEE 802.15.4 is specified for low rate wireless personal area networks (LR-WPANs) [1]. This standard emphasizes a wireless communication system with low data rate (20/40/250 kbps), low power consumption, short range (less than 10 m), and low cost. Its applications targets on low cost market, such as home and building automation, consumer electronics, PC peripherals, and medical monitoring. IEEE 802.15.4 standard is the MAC and PHY specifications, and ZigBee defines the upper layers of protocol stack, from network to application [2]. The standard specifies two operating frequency bands, one is 868/915 MHz band with 20/40 kbps data rate, and the other is 2.45 GHz band with 250 kbps data rate. This paper presents the 2.45 GHz baseband transceiver design.

Due to the brief definitions and specifications in the IEEE 802.15.4 standard, it provides the feasibility to design the baseband processors. There are two different approaches to consider the carrier phase in baseband circuit design. Both of them estimate the phase error in baseband, also demodulate the signals and process the data. The difference is that one compensates the phase error in baseband, and the other just indicates the error information for VCO to lock it. The benefit to handle carrier phase in digital is reducing the complexity of RF, and easy partition in SOC design.

In this paper, carrier phase synchronization including phase error estimation and compensation, is implemented in full digital. And it presents a new residue phase error tracking technique to ensure the data correctness for longer packet length. The whole baseband transceiver is designed with complete cell-based design flow, and will be verified through chip fabrication.

This research is supported in part by the National Science Council, Taiwan, R.O.C. under Grants NSC-96-2220-E-007-020, NSC-95-2219-E-002-020 and NSC-95-2220-E-007-006, supported in part by Ministry of Economic Affairs, Taiwan, R.O.C. under Grants 95-EC-17-A-01-S1-002.

II. Architecture Design

A. Transmitter

The transmitter architecture is to meet the 2.4 GHz PHY specifications in IEEE Std. 802.15.4 [1]. The physical header includes preamble, SFD and frame length fields. The preamble field is used for packet detection, timing and carrier frequency synchronization. After header insertion, the bit-to-symbol block maps 4 bits to one symbol. Then, the symbol-to-chip block performs spread spectrum, which maps one symbol to a 32 chips through 16 quasi-orthogonal spreading codes. And the spread chips are distributed into real part and imaginary part Offset-QPSK (O-QPSK) modulation. O-QPSK with half-sine pulse shaping nearly equals to MSK modulation [3], which is to increase the power spectrum efficiency, and reduce the complexity of power amplifier.

B. Receiver

The receiver architecture is illustrated in Fig. 1. The receiver works by three steps: packet detection, synchronization, and data recovery. At first step, only packet detection block is enalbed, and others are turned off until detecting the packet. Next is the synchronization state, the receiver performs phase synchronization and timing synchronization. At this state, carrier frequency synchronization block is turned on to estimate frequency error by preamble, and phase compensation block works for phase rotation. De-spreading block also operates for resolving the physical header to collect the information about the packet. After preliminary synchronization is done, data recovery state begins. Symbol-to-bit block works to recover the data bit stream for MAC, and phase tracking block starts for fine phase error tracking. Other blocks are turned off to save power consumption.

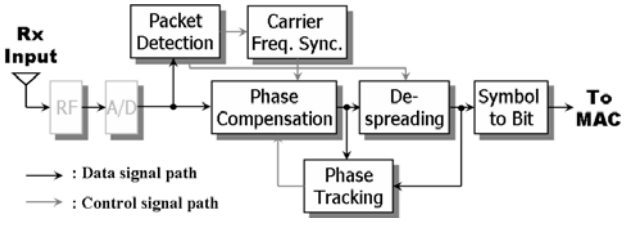

Fig. 1. Block diagram of the receiver.

C. Packet Detection

Due to less margin of carrier frequency offset estimation, the conventional delay-and-correlation method [4] cannot be adopted directly. The packet detection algorithm in this work is a modified version of conventional delay-and-correlation method [5]. The proposed packet detection scheme is shown in Fig. 2, and the algorithm works as:

$$C_n = \sum_{i=0}^{L-1} [z_{n-i} z^*_{n-i-D}] d_i \tag{1a}$$

$$P_n = \sum_{i=0}^{L-1} |z_{n-i-D}|^2 \tag{1b}$$

$$M_n = \frac{|C_n|}{P_n} \tag{1c}$$

$$d_i = p_i p^*_{i+D} \quad \text{where } p_i \text{ is the preamble sequence} \tag{1d}$$

Where d_i is calculated by cross-correlation of preamble sequence, and works like the coefficients of a matched filter. C_n is the result of cross-correlation of received samples passing through a matched filter with coefficients d_i. P_n is auto-correlation of received samples, the same as before. By definition, M_n is the quotient of C_n divided by P_n, indicates a normalized value between 0 to 1. If the received samples is just preamble sequences, and its timing is matched, it will produce a peak in the term M_n. If the peak value is above a pre-defined threshold, it is called a valid peak. Because there are 8 successive preamble symbols in the head of a packet, there will be ideally 8 valid peaks produced by the packet detection algorithm. It is useful to detect packet and to detect symbol boundary (timing synchronization) by counting the number of valid peaks. Packet detection is assumed successful if detecting one valid peak. And symbol boundary detection is called successful if detecting a few valid peak consecutively.

The carrier frequency offset $\Delta\hat{f}$ is calculated at the moment of valid peak occurred by:

$$\Delta\hat{f} = \frac{1}{2\pi DT} \arg\left\{ \sum_{i=0}^{L-1} [z_{n-i} z^*_{n-i-D}] d_i \right\} \tag{2}$$

And the maximum carrier frequency offset which this algorithm can detect is:

$$\Delta\hat{f}_{max} = \frac{1}{2\pi DT} \cdot \pi = \frac{1}{2 \cdot 8 \cdot 0.25 e^{-6}} = 250\ kHz \tag{3}$$

Undoubtedly, the maximum estimated range is wider than that specified by standard.

D. Carrier Phase Synchronization

Because of the carrier frequency of oscillator in transmitter is not matched to that in receiver, there exists a carrier frequency offset. The received samples will rotate by an amount of phase offset. Thus phase synchronization is required to compensate the phase mismatch. Besides, due to the interference of noise in channel, the amount of phase offset cannot be estimated exactly. It requires extra operations to calculate the residue phase error between ideal and estimated values.

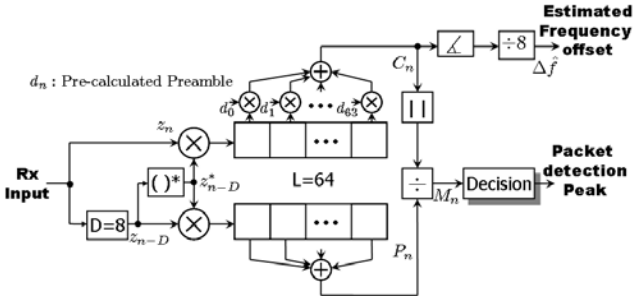

Fig. 2. Block diagram of packet detection.

1) Initial Phase Estimation: Though the carrier frequency offset is estimated in the packet detection part, we only know the phase offset of two adjacent samples. Because transmitter and receiver are asynchronous, the accumulated phase error of first sample in receiver is still unknown. For convenience, we have to assume the initial phase of first sample to be compensated as zero. Thus it is necessary to estimate the initial phase of first sample, and assures the correctness of phase compensation.

There are two problems in initial phase estimation. First, noise has a significant influence on the accuracy of estimation. Estimation by a few samples is useless, averaging a period of samples is necessary. Besides, because of a large value of carrier frequency offset, the phase error will be accumulated too much to affect during average procedure. Thus, estimation by just averaging a period of samples is not enough. For the purpose, carrier frequency offset is considered during estimation initial phase.

The procedure to estimate initial phase is: At the start, to de-rotate the received samples by step-up carrier frequency offset. Then, to calculate the phase difference between the received samples after de-rotated and the known preamble sequence. At last, to average a period of phase difference, and to get the value of initial phase.

At the first step, the effect of carrier frequency offset is subtracted from the received preamble samples. Obviously, the phase difference between the received preamble samples and the known preamble samples is the initial phase.

The initial phase $\hat{\phi}$ is calculated by:

$$\hat{\phi} = \arg\left\{ \sum_{k=0}^{L-1} z_{n+k} (p_{n+k} e^{j(k\Delta\omega)})^* \right\} \tag{4}$$

2) Residue Phase Error Tracking: Under low signal-to-noise ratio (SNR), carrier frequency offset cannot be estimated very precisely in packet detection algorithm. The same reason for initial phase estimation. Thus it results in a little amount of phase error between ideal phase and estimated phase of received samples, that called residue phase error. Because the rotated angle caused by carrier frequency offset is large, the residue phase error will be accumulated a lot over a long set of samples. If the cumulative residue phase error is too large to exceed the spacing of sample constellation (in O-QPSK, it is 90 degree), the sample fails at demodulation. This is why we have to track the residue phase error to prevent incorrect

decision during demodulation.

In this work, we adopt decision-feedback method to track the phase error. First, we use the output symbol of de-spreading circuit to look up the spreading code table, and get the corresponding spreading code. At the same time, the received samples without phase compensation are delayed by a symbol period before entering as inputs. Then we calculate the phase difference of this spreading code and the delayed received samples per sample in a symbol period. Finally, averaging the phase difference, and the residue phase error is estimated done. The procedure of estimating residue phase error is much similar to that of estimating the initial phase.

The residue phase error $\hat{\theta}$ is calculated by:

$$\hat{\theta} = \arg\left\{\sum_{k=0}^{L-1} z_{n+k}(p_{n+k})^*\right\} \quad (5)$$

Fig. 3 shows the proposed phase synchronization architecture. Phase estimation part, which includes carrier frequency offset estimation, initial phase estimation, and residue phase error tracking. This part estimates the values of three kinds of phase errors, and saves these angle values for further calculations. Phase accumulation/calculation part performs phase calculation with the data from phase estimation part. Phase compensation part includes an NCO and a complex multiplier. It converts the phase from an angle to a complex number, and compensates the phase error by multiplying the complex number to the original sample.

There are two operation modes in phase synchronization procedure, acquisition mode, and tracking mode. First, after packet detection is successful, the phase synchronization circuit starts working, and enters acquisition mode. During the acquisition mode, there are two known phase, carrier frequency offset $\Delta\hat{\omega}$, and initial phase $\hat{\phi}$. Because carrier frequency offset is the phase shift between two successive received samples, it is needed to accumulate carrier frequency offset per sample for correct phase. The total estimated phase error $\hat{\psi}$ in this mode is calculated by:

$$\hat{\psi} = \hat{\phi} + \Delta\hat{\omega} \cdot \text{counter} \quad (6)$$

The acquisition mode is working until receiving the end of preamble sequence. When the de-spreading circuit begin to de-spread data, the phase synchronization circuit enters the tracking mode. In this mode, the residue phase error $\hat{\theta}$ should be considered for fine phase error tracking. Though the residue phase error is calculated per symbol (64 samples) period, the total estimated phase error $\hat{\psi}$ at one symbol start is calculated by:

$$\hat{\psi} = \hat{\phi} + \Delta\hat{\omega} \cdot \text{counter} + \hat{\theta} \quad (7)$$

And at the other time in a symbol period, the total estimated phase error $\hat{\psi}$ is calculated just adding carrier frequency offset per sample:

$$\hat{\psi}_{\text{current sample}} = \hat{\psi}_{\text{previous sample}} + \Delta\hat{\omega} \quad (8)$$

In real circuit design, area and power can be saved by hardware reuse. For instance, it is not necessary to calculate the phases of carrier frequency offset, initial phase and residue phase error individually. Only one devider and one arc-tangent rom table are required for these three complex numbers. Furthermore, the circuits for initial phase estimation and residue phase error tracking can be shared because of common terms of them.

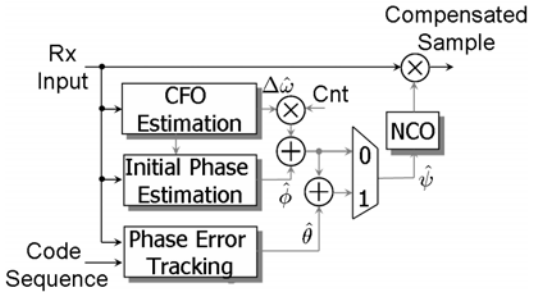

Fig. 3. Block diagram of phase synchronization.

III. Simulation and Verifications

A. System Performance Simulation

The system performance is evaluated by floating-point simulation results. The simulation environment is AWGN and allowable maximum carrier frequency offset (±40 ppm) by standard specification. The simulation results is shown in Fig 4, including a comparison with [7] and [8]. The former presents a MMSE receiver with training correlators, but simulates without carrier frequency offset. And the latter presents a FPGA baseband solution without compensating phase error in digital. In this paper, the packet error rate (PER) can achieve to 0.01 at SNR lower than 5 dB. Considering the PER degradation caused by carrier phase error and packet synchronization, this paper with fully digital synchronization has better performance.

B. Fixed-point Simulation

it is beneficial to run fixed-point simulation and decide the word-length of module I/O. It also ensures the balance of circuit complexity and system performance, improves hardware efficiency. The fixed-point simulation performance is close floating-point simulation results, just less than 1 dB degraded.

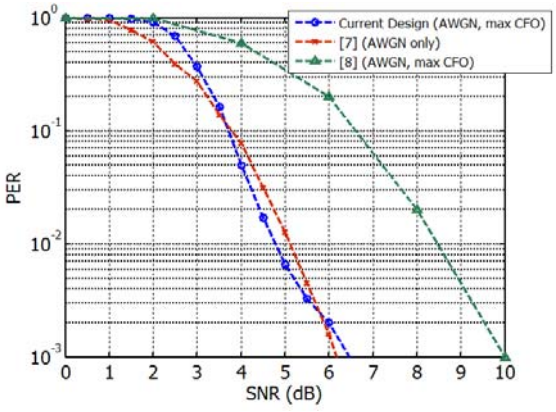

Fig. 4. PER versus SNR simulation.

TABLE I
SUMMARY OF ZIGBEE BASEBAND PROCESSOR IC.

Technology	TSMC 0.18 μm CMOS
Supply Voltage	1.8 V
Clock Freq.	4 MHz
Power Consumption	78.3 μW (Tx) / 1.719 mW (Rx)
Gate Count	78k
Chip Area	1.07 x 1.06 mm^2 (core) /1.63 x 1.63 mm^2 (chip)

IV. ASIC IMPLEMENTATION

By area and power analysis of the whole design, it is found that packet detetion and despreading is the major part that occupy most area and consume most power. Fortunately, recalling the operating procedure of receiver, these two blocks neither works at the same moment. That means one can be turned off while the other is at work. Gated-clock technique is used in circuit design level for the purpose to save power. The enable signal can be easily generated by observing circuit states of control logics. The power consumption can be reduced more than 50% with gated-clock usage. Furthermore, the circuit is designed and operatiog at 4 MHz, which is a lower speed in ASIC. Basic and simple circuit elements are adopted instead of speedy or comlicated elements for saving area and power.

The proposed ZigBee baseband processor was fabricated in the TSMC 0.18μm 1P6M CMOS technology. The die size is about 1.633 mm $\times$ 1.633 mm and packaged in a 68-pin CLCC. The layout and the summary of the chip are shown in Fig 5 and Table I, respectively. The power flows of transmitter and receiver parts are separated for measuring power consumption individually.

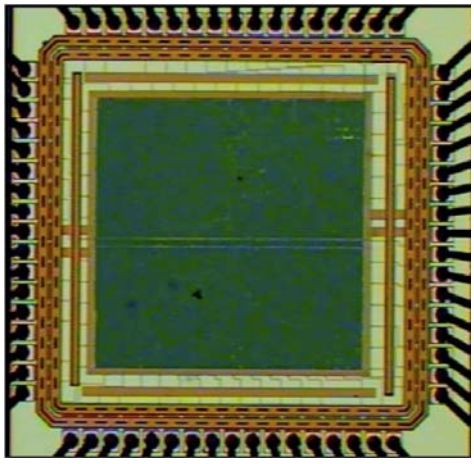

Fig. 5. Chip photo.

The baseband processor is working at an internal clock rate of 4 MHz, which is two times oversampling of chip rate. Transmitter consumes 78.3 μW and receiver consumes 1.719 mW at V_{DD} = 1.8 V. Table II compares the proposed chip with other published chips, and shows the poposed chip with less power consumption.

TABLE II
COMPARISON OF ZIGBEE PROCESSOR ICS. THE THIRD IC INCLUDES RF TX AND RF RX POWER CONSUMPTION.

Design	Supply Voltage	Clock Freq.	Tx Power	Rx Power
Ours	1.8 V	4 MHz	78.3 μW	1.72 mW
[8]	1.8 V	2.4 MHZ	3.28 mW	3.29 mW
Atmel - AT86RF210 Z-Link	1.8 V	2.4 MHz	108 mW	26.1 mW

V. CONCLUSION

In this paper, the system design and circuit implementation of ZigBee baseband processor is presented. The proposed receiver contains a packet detection, a carrier phase synchronization, and de-spreading. Especially the carrier phase synchronization including phase estimation, phase tracking ,and phase compensation is designed in full digital. With synchronization, the system performance (PER) achieves 1% under 5 dB SNR. The baseband receiver is designed with simple architecture and low hardware complexity. In order to save power, gated-clock technique is adopted and the receiver works with a low power consumption of 1.7 mW.

REFERENCES

[1] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4, 2003.

[2] (2006) The ZigBee Alliance Website. [Online]. Available: http://www.zigbee.org/.

[3] S. A. Gronemeyer and A. L. McBride, "MSK and Offset QPSK Modulation," in *IEEE Transaction on Communications*, vol. COM-24, pp.809–820.1976.

[4] Juha Heiskala and John Terry,*OFDM Wireless LANs: A Theoretical and Practical Guide.* Indianapolis, USA: Sams, 2001.

[5] F. Classen, H. Meyr and P. Sehier, "An all Feedforward Synchronization Unit for Digital Radio," in *Proc. IEEE Vehicular Technology Conf.*, pp. 738–741, Secaucus, NJ, May 1993.

[6] John G. Proakis, *Digital Communications.* 4th ed., New York: McGraw-Hill, 2001.

[7] Karen E. L. Scott and Robert W. Stewart, "Design and Performance of IEEE 802.15.4 Compliant MMSE Receivers," in Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004. vol. 2, pp. 2051–2055. Nov. 2004.

[8] Chua-Chin Wang, Jian-Ming Huang, Chih-Yi Chang, Kuang-Ting Cheng and Chih-Peng Li, "A 6.57 mW ZigBee transceiver for 868/915 MHz band," in Proc. of 2006 International Symposium on Circuits and Systems, pp. 5195–5198. May 2006.