A low power circuit design of BLE baseband transmission data processing module

Liang Ding[1,*], *Jiansong* Qin[2], *Xiaobo* Zhang[3], *Jinpeng* Zhang[2], *Changbin* Yin[2], and *Zhihai* Yang[2]

[1]State Grid Zhejiang Electric Power Co., Ltd., Shangyu Power Supply Company, Shangyu, China
[2]State Grid Zhejiang Electric Power Co., Ltd., Shaoxing Power Supply Company, Shaoxing, China
[3]State Grid Zhejiang Electric Power Co., Ltd. Innovation and Entrepreneurship Center, Hangzhou, China

Abstract. With the quickly development of the Internet of Things technology, Bluetooth (BLE) protocol has been increasingly adopted in consumer communication applications, industry communication utilization, and even positioning requirement, due to its advantages such as low power consumption, low cost, and low complexity. Especially the outstanding low power capability compared with other short-range communication protocol, many power limited AIoT (Artificial Intelligence IoT) devices select BLE as basic component. So, the low power, high performance, and small area BLE module become the key point of successful of whole AIoT system. In BLE system, the baseband data processing is the foundation. A good hardware circuit design of BLE baseband can improve BLE module performance and decrease power simultaneously. This paper analyzes the baseband data processing function of BLE in detail, and designs the baseband transmission data processing RTL circuit IP using Verilog HDL Hardware description language based on BLE protocol. The IP includes CRC verification module, whitening module and coding mapping function module. The RTL design is simulated and evaluated in workstation server with Synopsys VCS tool. The simulation results show that this design can achieve the Low-power BLE Bluetooth baseband function.

1 Introduction

Bluetooth is a global wireless protocal that supports device short range communication with a wide range of applications. In July 2010, BLE proposal [1-3] opened up a new era of Bluetooth. The power consumption of BLE is almost one tenth of traditional Bluetooth. Its low cost, backward compatibility, faster transmission speed, longer effective distance, more accurate positioning accuracy, more suitable device and other characteristics are more suitable for application requirements of the Internet of Things (IoT).

The penetration rate of Bluetooth chips in multiple functional areas such as audio transmission, data transmission, positioning services, and intelligent wearable devices continues to increase. SIG predicts that the total global shipment of Bluetooth devices is

* Corresponding author: dingliang2k@163.com

expected to reach 7 billion by 2026. Leading overseas manufacturers such as Nordic, Dialogue, and TI, leverage technological development and resource; IC companies such as Botong Integration and Jerry Technology have improved their capabilities in chip design, manufacturing, or packaging testing to explore market development space.

Most BLE chips is with BLE IP for communication, such as CEVA[4,5]. Domestic manufacturers have a significant technical dependence on the IP processing of the underlying hardware of the digital baseband[6~8]. Moreover, due to the addition of new data processing methods for encoding and transmission in the fifth generation Bluetooth, it is more suitable for the modern Internet of Things. Therefore, studying the data processing circuit of the fifth generation BLE baseband controller link layer has practical significance. This article first introduces the research background and significance; Section II introduces the BLE layered structure and baseband link layer specifications; Section III introduces the design of baseband data transmission and processing circuits, and section IV provides functional simulation results; Section V summarizes the work of this paper.

2 BLE baseband link layer specification

2.1 BLE Baseband Structure

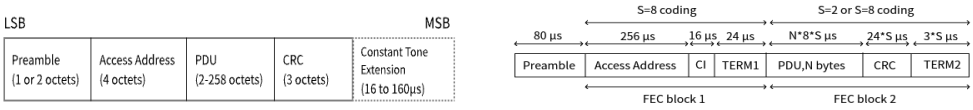
The BLE protocol consists of three parts: application layer, host layer, and controller layer. The application layer defines three types: characteristic, service, and profile. The Host Controller Interface (HCI) is used for the interaction between the controller part and the host part. The host layer is located in the upper layer of the HCI interface and is an important part of the software protocol. It manages communication between devices and implements specific functions, including two specifications and three protocols. The controller layer is located at the lower layer of the HCI interface, defining RF and baseband hardware specifications, including the physical layer and link layer. The physical layer is mainly responsible for sending modulated signals through RF, and the link layer is an important part of digital circuit design. It is mainly responsible for establishing and maintaining the transmission process, switching the RF working state, and packaging, combining, and bitstream processing the corresponding messages. This article designs the bit stream processing part of the link layer, which processes the data to be sent.

2.2 BLE Link Layer Sending Message Structure and Processing Process

2.2.1 Message structure

The message types of BLE include three types: non encoded types with 1Mbps/s data transmission rate corresponding to the physical layer LE 1M, non encoded types with 2Mbps/s data transmission rate, and encoded types with 1Mbps/s data transmission rate.

The forms of encoding types and non encoding types are different, and the basic structure is shown in Fig.1 and Fig.2 [1-3]. Two types of messages have a common part: Preamble, Access Address, PDU, and CRC, which are sent in 8-bit bytes.



(a) Uncoded PHY (b) Coded PHY

Fig. 1. Link Layer packet format for the LE Uncoded and coded PHY.

2.1.2 Sending Message Bitstream Processing Process

For the two types of BLE messages, bitstream processing is required before sending them to the RF interface to enhance transmission and reception reliability. The bitstream processing processes for non encoding and encoding type are shown in Fig.3, respectively:

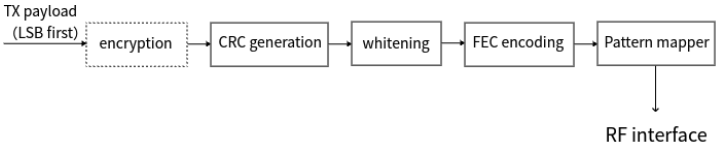


Fig. 3. Payload bit processes for the LE Coded PHYs.

During the transmission process, CRC calculation is for the convenience of comparing with the received data to verify the correctness of the data. Whether it is in the encoding type or non encoding type of message transmission, the PDU part needs to be calculated. Due to the possibility of communication failure caused by the occurrence of long sequences 0 and 1, whitening operations are performed on the PDU and CRC parts. FEC encoding and mapping are carried out for encoding type messages, which have the characteristic of forward error correction. This new encoding processing is to enhance the robustness of the data and adapt to the requirements of long-distance transmission.

3 BLE Database Module Design

3.1 CRC calculation module

The BLE CRC code is 24 bits, and its polynomial form is: Design the circuit diagram of Linear-feedback shift register, as shown in Fig.4:

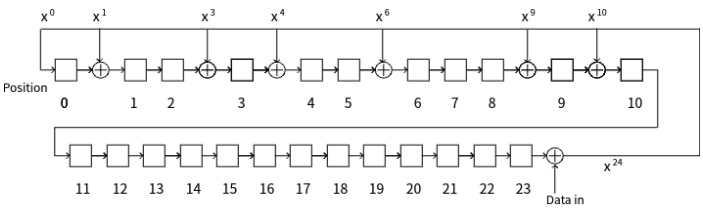


Fig. 4. The LFSR circuit generation.

When transmitting, first load the initial value in the shift register to cover the 24 bit CRC cyclic check digit. This design consists of a CRC polynomial circuit with 24-bit shift register and a modulo two adder. Due to the PDU's transmission direction being from low to high header bits, the crc results stored in crc_ lfsr from low to high bit.

3.2 Whitening module

The BLE whitening module is performed before FEC operation, and the corresponding de-whitening is performed after FEC operation. Whitening and whitening modules are the same, and both use the 7-bit Linear-feedback shift register of polynomial, as shown in Fig.5:

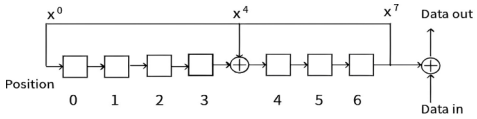


Fig. 5. The LFSR circuit to generate data whitening.

The lowest bit of the initial value sequence of its shift register is 1, and the other high bits are determined by the physical channel index of the message transmission. That is, the high to low bits of the initial value of the high six bits correspond to the high to low bits of the channel index value, respectively. For example, if the channel index is 23 (0x17), the initial value from high to low is 1110101. After initialization, the input data is subjected to a modulo two addition operation, and then output. At the same time, the highest bit and the third bit are subjected to a modulo two addition operation and assigned to the fourth bit. The remaining bits are shifted to the right to form a new sequence, which is continuously executed until the whitening operation of the PDU and CRC is completed.

3.3 Encoding mapping module

For encoding type messages, an encoding mapping module needs to be designed. Firstly, the message is encoded by FEC convolutional encoder, and then it needs to be mapped and sent according to the mapping table. The encoding mapping operation is performed after the whitening operation, and there are different mapping requirements for different regions.

Let n represent the output bits of the convolutional encoder (n, k, m), k represent the input bits, m represent the encoding register bits, and m+1 represent the constraint length. If the BLE encoding rate is 1/2 and the constraint length is 4 for a (2, 1, 3) convolutional encoder. The circuit structure of the corresponding convolutional encoder is shown in Fig.6, and corresponding polynomial is calculated by Equation (1) and (2):

$$G_0(x) = x^3 + x^2 + x + 1 \quad (1)$$

$$G_1(x) = x^3 + x^2 + 1 \quad (2)$$

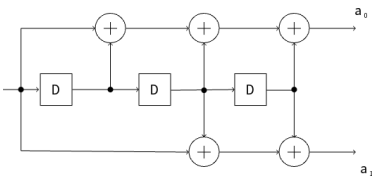


Fig. 6. Convolutional Forward Error Correction encoder.

Firstly, transmit the a0 bit of G0, and then transmit the a1 bit of G1. Encode one input bit into two, and then map the encoded data. The BLE mapping mechanism includes two types: S=2 and S=8, as shown in Table 1 (output sequence in transmission order):

Table 1. Input output comparison table.

Input Data	Ouput Sequence S=2	Output Sequence S=8
0	0	0011
1	1	1100

The mapping methods for different regions of the message have different requirements. For example, AA, CI, and TERM1 are fixed mapping methods with S=8, while the mapping methods for PDU, CRC, and TERM2 depend on the value of CI: when CI=00b, the mapping method with S=8 is used; When CI=01b, use the mapping method of S=2.

3.4 Dual Port Synchronous Read Write RAM Module

This article designs a dual port synchronous read and write module with a width of 32 bits and a depth of 256 bits, which has two sets of control signals, data input signals, and address signals. It can read synchronously without any conflicts. However, to prevent write conflicts, a chip selection priority signal is set, so only the ports with higher priority will be executed for writing. The overall RAM working process is as follows: when the clock arrives, two sets of ports perform operations based on the chip selection signal cs, write valid signal we, and output valid signal The read and write operation requires the chip selection signal to be valid, and the write signal to be valid, while the read and write signal to be invalid and the output signal to be valid.

3.5 State Machine

Set the corresponding sending status based on the encoded and non encoded fields. The messages common parts are preamble, AA, PDU, and CRC. The difference is that the encoding type of message has added CI, TERM1, and TERM2 fields, while the non encoding type has optional CTE fields. The PDU for data physical channels and synchronous physical channels has optional MIC parts, so MIC is not specifically sent as a state. If there is MIC, it is sent along with payload. The state machine is shown in Fig.7.

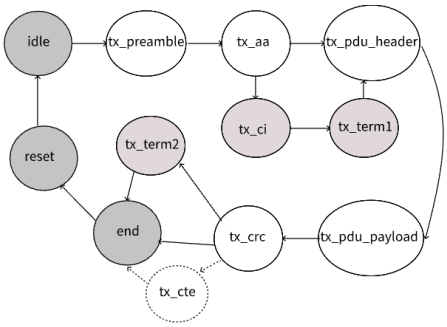


Fig. 7. Sending data state machine state jump diagram.

The payload of PDU and the total number of bytes of possible MICs is determined by idle status r_bytes in BLE_HEADER[15:8]. The tx_preamble state uses r_cnt to count the number of bits. When sending a non coding type of 2Mbps, 16 bits of data are sent, and when the r_cnt count is 15, it jumps to the tx_aa state; When sending message of 1Mbps non coding type, send 8-bit data and jump to tx_aa state when r_cnt count=7; When

sending message with 1Mbps encoding type, send 80 bits of data and jump to tx_aa state when the r_cnt count is 79. When jumping to the tx_aa state, load the 32-bit access address BLE_ACC_ADDR into the shift register.

The tx_aa status can be divided into two situations: when the message is of the encoding type of 1Mbps, change the vld_flag_coded flag of valid encoding, carry out encoder input and shift once every 8 cycles, use r_cnt to count the eight bits sent successively after each address data passes the encoding mapping, and jump to the tx_ci status when all 32-bit mapped data is sent; When the message is of another type, the shift register is shifted to the right to send the lowest bit, and the r_cnt counts the number of bits sent to the access address. When the r_cnt counts to 31, it can jump to the next state tx_pdu_header and load the 24 bit BLE_HEADER into the shift register.

Before jumping to the tx_pdu_header state, initialize the CRC calculation module and whitening module to facilitate subsequent CRC calculation and data whitening sending. At the same time, point the address of RAM port 2 to the end for subsequent extraction.

The tx_ci state mapping and transmission method are similar to tx_aa, requiring an interval of 8 cycles to shift the shift register and send new data to the encoding mapping module. After sending the CI field encoding mapping data, it can jump to the txterm1 state and load the 3-bit BLE_TERM1 into the shift register.

The tx_term1 state is similar to the sending of CI fields in the tx_ci state. After sending a 3-bit TERM1, it jumps to the tx_pdu_header state and sets the 24-bit BLE_HEADER is loaded into the shift register. For subsequent whitening and CRC calculation, initialize the CRC calculation and whitening module before jumping.

Tx_pdu_header status: For an encoding type of 1Mbps, the mapping method S of the header and its subsequent Payload, CRC, and TERM2 depends on the input CI field: when CI=00, S=8; When CI=01, S=2. When S=8, keep the effective signal of the input data of the CRC and whitening module for 1 cycle and shift transmission every 8 cycles to ensure that the transmitted data is the data passing through the whitening and encoding mapping module, and CRC calculation always follows data transmission rate of 125kbps; Similarly, when S=2, data transmission rate is 500kbps. When r_cnt=0 & r_cnt_cnt=12, the chip selection signal and output valid signal of RAM port 2 are set valid, and write enable signal is set invalid. In the next r_cnt=0, the RAM read data is written to the BLE_PAYLOAD register. Finally, when the header sends the completed state and jumps to tx_pdu_payload, BLE_PAYLOAD is written to the shift register, and output the lowest 32-bit Payload.

For other non coding types, it is necessary to determine whether it is a data type message and whether the header CP bit is 1 to determine whether there is a constant extension field. If there is no need to transfer 16 bit header, then 24 bit header needs to be transferred. After judgment, first set the initialization signal of the CRC calculation and whitening module to invalid, and then perform data shift, whitening and transmission, and CRC synchronous operations. It is necessary to count r_cnt to the last four bits (i.e. count to 20 when transmitting a 24-bit header and 12 when transmitting a 16-bit header) to effectively maintain the read signal of RAM port 2, And in the next r_cnt count, the 32-bit data read from RAM port 2 will be written to the BLE_PAYLOAD register. When jumping to the tx_pdu_payload state, the lowest 32-bit Payload content retrieved, BLE_PAYLOAD, will be loaded into the shift register.

The tx_pdu_payload state continues to be whitened and CRC calculated. Unlike other fixed length fields, Payload, as a field with a larger range, needs to be written in RAM in advance, and read out and loaded into the shift register in sequence when sending Payload content.

When transitioning to the new state of tx_crc, close the CRC calculation module and load the 24-bit register results of the CRC module into the shift register. The whitening module needs to continue to be turned on for subsequent CRC whitening processing.

The tx_crc status sending method is similar to the PDU method, and the whitening module needs to enable whitening of the CRC before performing different processing of encoding and non encoding. After sending the CRC, the whitening module can be closed and jump to a new state: for encoding types, it is necessary to jump to the tx_term2 state and load the fixed sequence of TERM2 into the shift register; For non coding types, it is necessary to determine whether there is a CTE field. If there is no CTE field, it can be directly jumped to the end state to end the transmission.

The tx_term2 state is used to reset the register of the FEC encoder, so the encoding mapping module sends out TERM2 and jumps to the end state.

Tx_cte determines the number of bytes to send based on the CTETime information in the header's CTE information, and then sends them. After sending, it jumps to the end state, and then the valid flag for sending data is cleared.

4 Simulation

The transmission order of BLE messages is sent from the lowest bit, where the state setting corresponds to idle=4'b0000, tx_preamble=4'b0001, tx_aa=4'b0010, tx_ci=4'b0011, tx_term1=4'b0100, tx_pdu_header=4'b0101, tx_pdu_payload=4'b0110, tx_crc=4'b0111, tx_term2=4'b1000, tx_cte=4'b1011. When preamble=8'b01010101, access_address=0xAA08192B, CRC_initial_value=0xC4C181, channel_index=16, PDU_header=0x0516, PDU_payload=0x0504030201. As shown in Fig.8, when r_fsm_data_vld is valid, r_fsm_data is sent orderly, the results are consistent with BLE.

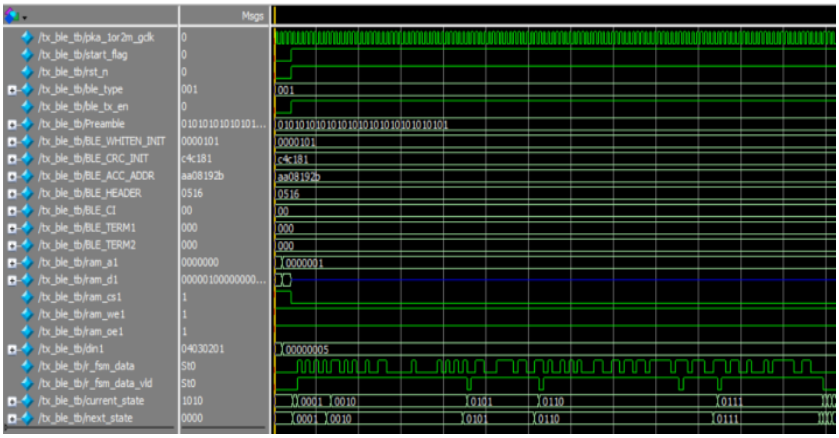


Fig. 8. Verification of message sending without CTE for non coding types.

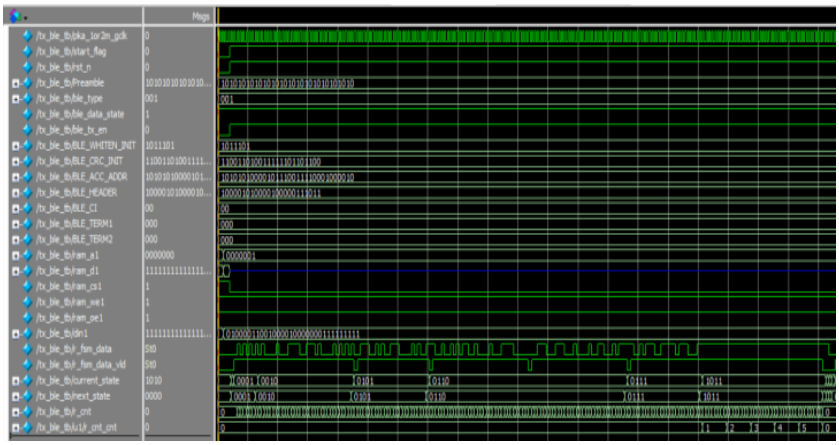


Fig. 9. Verification of sending non coding type messages with CTE fields.

According to the sample message, PDU_preamble=8'b01010101, access address is 0xAA173C42, initial value is 0xCD3F6C, channel index is 29, whiten initial value is 1011101b, PDU_header=0x85083B,PDU_payload=0x4321_01FF_FFFF_DB01. As shown in Fig.9, when r_fsm_data_vld is valid, data sequence was sent during tx_cte state, and the result is consistent with the BLE. For encoding type messages, preamble is 10 {00111100b}, the access address is 0x8E89BED6, CRC initial value is 0x555555, channel index is 7, PDU_header=0x0300, and PDU_payload=0x00454C42. As shown in Fig.10 and Fig.11, it is consistent with the encoding type of BLE.

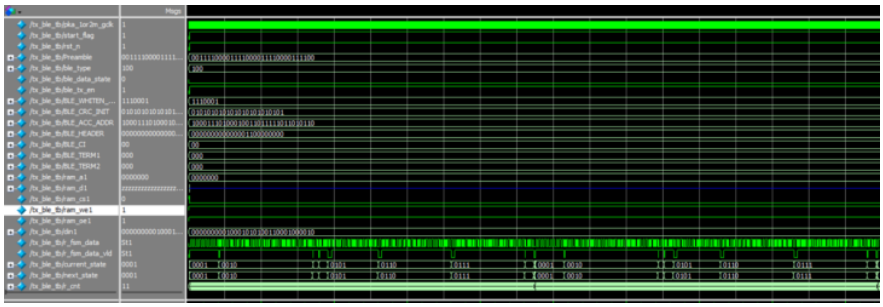


Fig. 10. Message sending verification for encoding type (CI=00)

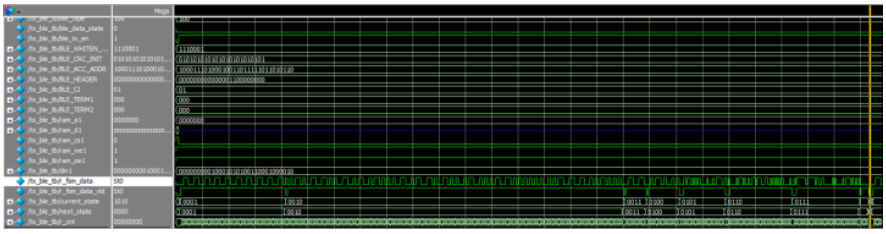


Fig. 11. Message sending verification (CI=01) for encoding type.

5 Conclusions

This paper studies the processing process of baseband transmission data based on the BLE 5.3 specification. Design the data transmission module in the Bluetooth baseband controller

link layer, use Verilog hardware description language for RTL design, construct a bit stream processing function module for CRC calculation, whitening processing, and encoding mapping, use a state machine for segmented data processing and transmission, and finally perform simulation verification. The results are compared and verified with the complete message of sample, and obtain correct transmission result. Simulation experiments show that this design can achieve low-power BLE baseband functionality.

This work was funded by the Corporation Science and Technology Project with Ningbo Transmission & Transformation Construction Ltd, which is Key Technologies for Underground Cable Intrusion Early Warning Based on MEMS Sensor Array. Thanks the great effort from all team members.

This study is supported by the Mass Entrepreneurship and Innovation Program of State Grid Zhe Jiang Electric Power Company. The project is Key Technology Research and Demonstration Application of Underground Cable Intrusion Early Warning System Based on MEMs Sensor Array. Thanks the great effort from all the team member co-collaborator.

References

1. Bluetooth SIG. Specification of the Bluetooth System Version 5.3[EB/OL].http://www.bluetooth.org. 2021.
2. Bluetooth SIG. Specification of the Bluetooth System Version 5.1[EB/OL].http://www.bluetooth.org. 2019
3. Bluetooth SIG.2022 Market Update[EB/OL].2022.
4. J. Liu, and M. Cai, “GFSK Modulation for BLE Baseband IC Design”, International Conference on Electron Devices and Solid-State Circuits (EDSSC), 2017.
5. A. Pipino, A. Liscidini, K. Wan, A. Baschirotto, “Bluetooth low energy receiver system design”, IEEE International Symposium on Circuits and Systems (ISCAS), 2015.
6. B. Park, K. Kwon, “2.4-GHz Bluetooth Low Energy Receiver Employing New Quadrature Low-Noise Amplifier for Low-Power Low-Voltage IoT Applications”, IEEE Transactions on Microwave Theory and Techniques, Vol. 69, Issue. 3, 2021.
7. S.C. Folea, G.D. Mois, “Lessons Learned From the Development of Wireless Environmental Sensors”, IEEE Transactions on Instrumentation and Measurement, Vol.69, Issue.6, 2021.
8. I. Bukreyev, H. Ken, A. Apsel, “Scalable Digital Synchronizer for Enabling Hardware-Level BLE Mesh Networks under 1 mW”, IEEE 47th European Solid State Circuits Conference (ESSCIRC), 2021, pp.251-254.