Analysis and implementation of SDH protocol in 10Gbps POS signal

Ying Wei¹, Jiawei Huang² and Liuyi Ling²,³

¹College of Electrical and Electronic Engineering, Anhui Sanlian University, Hefei 230601, China
²School of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan 232001, China
³E-mail: lyling@aust.edu.cn

Abstract. The data transmission network based on synchronous digital Hierarchy (SDH) can fundamentally solve the problems of capacity, quality, network management, security and so on in the network transmission of IP services. POS (Packet Over SDH) technology is a way to implement IP service in SDH data transmission network. An analytical solution of SDH protocol with Stratix5 FPGA as the core is studied. According to the SDH standard defined by ITU-T, the analytical circuit of SDH protocol in 10Gbps POS signal is designed on FPGA. The circuit is divided into data receiving module, line overhead module, channel spending module and status register output module, to complete data receiving, information extraction, and output of state information and net load data. The results of functional simulation and board level test show that the scheme is accurate, efficient and consistent with the standard, which can provide some reference value for the development of SDH technology.

1. Introduction
With the rapid development of information society, people's demand for modern information transmission network is getting higher and higher. SDH [1] data transmission network, as the core technology of information superhighway, which has the characteristics of large capacity, high quality, strong network management ability and fast transmission speed, has been widely used in telecommunication business in Mainland China.

POS technology is a way to carry IP services in SDH data transmission network. In recent times, POS has enabled transfer rates of up to 10 Gbps, and even 40 Gbps [2]. The advantage of POS is the ability to map IP packets directly over SONET/SDH via PPP/HDLC framing, which is more efficient. POS interface chips, for example PMC's PM5360 and Cortina System's CS1331, are necessary in order to receive POS signals on SDH devices. The production technology of these interface chips is in the hands of foreign chip manufacturers who have been implementing chip control resulting in greatly restricting the production of SDH equipment and the development of telecommunications business in China.

In order to solve the technical problems of SDH, some related research work has been carried out. Zhang et al. [3] proposed a method to simplify the E1 interface to SDH system based on FPGA. Ye et al. [4] designed a PDH to 622 Mb/s SDH/ SONET mapping &de-mapping chip, which integrates DS1/E1/J1 framer, DS1/DS3 multiplexer and E1/E3 multiplexer. Some scholars studied how to extract IP packets from POS signals, and designed the multi-rate POS test module [5], as well as studied the
SDH technology [6-10]. However, none of these studies has made a detailed analysis of the SDH protocol parsing process using FPGA. This paper studies in detail a SDH protocol parsing scheme based on Stratix5 FPGA and the work will have a certain reference value for the design of interface chips in SDH devices.

2. Logic design of FPGA
The overall design block diagram of SDH protocol analytic scheme is shown in Figure 1. It consists of data receiver module, segment overhead module, line overhead module, channel overhead module and status register module. Each module is implemented inside the FPGA, which is the 5SGXMA9N2F45C2 chip of Stratix V series of Altera Company.

![Figure 1. Overall design diagram of SDH analytic scheme.](image)

2.1. Data receiver module
The data receiver module defines the interface between FPGA and external data, which is used to receive SDH data stream. The Stratix V Transceiver Native PHY provided by Altera can parameterize IP cores for data reception, and convert high-speed serial self-synchronous data streams into low-speed, multi-bit wide parallel data streams. Set IP core parameters to 9953.28 Mbps of data rate, 64 bits of interface width and 155.52 MHz of CDR reference clock.

2.2. Segment overhead module
The data processed by the data receiver module is sent to the segment overhead module to complete byte reversal, frame alignment, BIP parity check, descrambling and overhead information extraction.

2.2.1. Byte reversal. The data transferred from the data receiving module to the segment overhead module is bit reversed, that is, the maximum significant bit is changed to bit7 and the least significant bit is changed to bit0, which makes the data format in the FPGA consistent with the data transmission format in the Ethernet.

2.2.2. Frame alignment. The core of segment overhead module is frame alignment. Frame alignment can locate the frame boundary of SDH data stream, that is, to determine the position of each frame. The frame alignment function is accomplished by a state machine, which is shown in Figure 2. The frame alignment state machine works in four states: idle state, A1 alignment state, pre-synchronization state and synchronization state.

1) idle state
The idle state as the initial state of the state machine completes byte alignment and A1 (11110110) sequence detection. If A1 sequence detection fails, i.e. no A1 sequence is found in the data, the state machine remains idle. If A1 sequence is detected in the data, the state machine enters A1 alignment state.
2) A1 alignment state
A1 alignment state completes the detection of SDH frame synchronization bytes A1 and A2. When the number of A1 and A2 exceeds 64, the state machine enters the pre-synchronization state. If the number of A1 and A2 detected does not meet the requirements, that is, the number of A1 or A2 is less than 64, the state machine returns to idle state.

3) Pre-synchronization state
Pre-synchronization completes the initial alignment of SDH frames. A byte counter is used to calculate the number of bytes starting from A2. When the total number of bytes is 15538 (155520-192), it means entering the next frame. When 8 consecutive frames of data are detected successfully, the state machine enters the synchronization state. Otherwise, it returns to the idle state.

4) Synchronization state
The synchronization completes the monitoring of SDH frame pre-synchronization. The number of frame headers A1 and A2 is detected in synchronous state. When the number of A1 or A2 in the frame headers of 8 consecutive frames is less than 32, it indicates that frame alignment fails and the state machine returns to idle state. Otherwise, the state machine remains in synchronous state.

![Figure 2. State machine of frame alignment.](image)

2.2.3. BIP parity check. The SDH frame is checked by BIP-8 parity and the result is output. The data of each byte corresponding to the bit is XOR or XOR with the B1 byte of the next frame after descrambling. If the result is 0, the check is correct, otherwise the check is wrong. The BIP-8 parity check calculation range is the whole frame before scrambling. Starting from the frame header, 155520 bytes are calculated, 8 bytes per cycle and 19440 cycles are calculated. Unscrambling does not include 576 bytes, including A1, A2, J0 and Z0, that is, from 577 bytes to 155520 bytes.

2.2.4. Descrambling. The descrambling area is net load data and overhead bytes. The formula used for de-scrambling is the random number generating formula: $x^7 + x^6 + 1$. Starting at the end of each
frame Z0, the initial value is set to 1111111111. In the internal of the FPGA, the descrambling is realized by using shift and exclusive OR, and the circuit is shown in Figure 3.

Figure 3. Descrambling diagram.

Suppose the values of each register are respectively \( r_0, r_1, r_2, r_3, r_4, r_5, r_6 \) from left to right in turn. The Expressions (1) and (2) are obtained.

\[
\begin{align*}
\begin{bmatrix}
 r_0^{+1} \\
r_1^{+1} \\
r_2^{+1} \\
r_3^{+1} \\
r_4^{+1} \\
r_5^{+1} \\
r_6^{+1}
\end{bmatrix} &=
\begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
 r_0^T \\
r_1^T \\
r_2^T \\
r_3^T \\
r_4^T \\
r_5^T \\
r_6^T
\end{bmatrix} \\
R_{r+k}^T = H^T R_r^T
\end{align*}
\]

The scrambling codes are divided into 10 groups with 7 bits in each group. The scrambling codes of the next cycle are obtained by multiplying \( H^{64} \) by the previous group of scrambling codes. Get high-64 bits to de-scramble the data. It is found that the amount of calculation of \( H^{128} \) is much less than that of \( H^{64} \), so \( H^{128} \) is chosen to de-scramble the data.

Figure 4. Overhead byte map.

2.2.5. Overhead information extraction. The overhead byte map is shown in Figure 4. The byte counter is used to locate each overhead byte in SDH frame data. The results of A1, A2 and Z0 are
checked and compared with those of BIP-8 in the previous frame. The output result is BIP-N (S). The information of J0 is captured and stored, and the state of STM and stu is judged. Finally, all the state information is output to the state register module.

2.3. Line overhead module
Line overhead module completes line overhead byte extraction, data separation between SPE (channel overhead + net load data) and SDH frame, and BIP-24 check.

2.3.1. Extraction of overhead bytes. The overhead bytes such as H1, H2, H3, K1, K2 and S1 are extracted according to the overhead byte map shown in Figure 4. The pointer information of SPE is obtained from the last 10 bits of data after the first byte combination of H1 and H2, and the starting position of SPE is found out. The SPE is sent to the channel overhead module.

2.3.2. BIP check. Similar to BIP check in segment overhead module, BIP-24 check is carried out on all bytes of line overhead byte and SPE after de-scrambling. The result of BIP-24 check is compared with that of B2, and finally the result of BIP check is output.

2.4. Channel overhead module
Channel overhead module receives SPE data from line overhead module, extracts channel overhead bytes and net load data, and completes BIP check.

2.4.1. Extraction of overhead bytes. The parameters of channel overhead, such as J1, C2, B3 and G1, can be extracted from the overhead byte distribution diagram shown in Figure 4 and the SPE structure diagram shown in Figure 5. After stripping out the overhead bytes, the net load data is obtained.

2.4.2. BIP check. Similar to BIP checking in segment overhead module, BIP-8 is calculated for the whole SPE region, and the results of the checking are compared with those of B3. Finally, the results of the checking are output.

2.5. Status register module
The state register module mainly completes the collection and collation of the status information of other modules, and stores the effective information to the corresponding register. The module provides the corresponding register and address, so that users can view the operation status of the internal circuit of the FPGA at any time, and maintain and monitor the various modules.

3. Simulation results and analysis
Write test files with Verilog language to generate complete and continuous multi-frame SDH data. The simulation software, Modelsim SE-64 10.4, is used to simulate the whole design. In the whole SDH parsing module, the design of the segment overhead module is the most complex. It is the scrambling code calculation in the alignment and de-scrambling part of A1 that determines whether the segment overhead module works properly. As shown in Figure 6, the simulation results of A1 alignment state show that the current working state of the state machine is A1 alignment state when
the value of register r_stat is 1. When the value of register r_state is 2, the state machine has completed the task of A1 alignment and worked in the pre-synchronization state. At the same time, the number of bytes of A1 and A2 has been detected to exceed 64, indicating that the frame has been aligned. When the number of consecutive correct frames is less than 8, the data transmission error occurs. The simulation waveform of frame error is shown in Figure 7.

![Simulation waveform of A1 alignment state](image1)

**Figure 6.** Simulation waveform of A1 alignment state.

![Simulation waveform in case of frame error](image2)

**Figure 7.** Simulation waveform in case of frame error.

Figure 8 shows the simulation results of scrambling code calculation in the scrambling part, where the register r_scramble indicates that the final output scrambling code is consistent with the results calculated by the formula.

![Simulation waveform of scrambling code calculation result](image3)

**Figure 8.** Simulation waveform of scrambling code calculation result.

### 4. Experimental results

The IXIA tester (IxLoad 3.2, Huawei Technologies CO., LTD) is used to send out the package, and the development board based on Stratix 5 FPGA is used as the receiving device. The data is transmitted through optical fibers. The status information inside the FPGA is observed in the host computer to determine whether the analysis is correct or not. The detail process does the following steps: the FPGA first detects whether the PMA in the data receiving module is locked; detects the calibration of A1 and 22, frame alignment, scrambling code generation, B1 and Z0 and J0 in the segment overhead module; detects the calculation and indication of the head of SPE area in the line overhead module, the verification of B2 and M0/M1; detects the determination of the net load area in the channel overhead module, and the verification of B3. Check, C2 and J1 and G1. After 12 hours of continuous outsourcing test, the status information output by the internal register module of the FPGA is read by the host computer software. Three different types of data packets are tested. The first test for packet type with fixed data is implemented by transmitting the data from IXIA to FPGA with 2479869 packets of 743960700 bytes and the transmission error rate is zero. The second and third test for the packets with different datat type are also implemented and the test results are shown in Table 1.
Table 1. Test results of data transmission.

| packet type                   | IXIA number of sent packets | FPGA number of received packets | error rate |
|-------------------------------|-------------------------------|---------------------------------|------------|
| with fixed data               | 2479869                       | 2479869                         | 0%         |
|                               | (743960700 bytes)             | (743960700 bytes)               |            |
| with incremental change data  | 4310587                       | 4310587                         | 0%         |
|                               | (1293176100 bytes)            | (1293176100 bytes)              |            |
| with random data              | 7438446005                    | 7438446005                      | 0%         |
|                               | (5749958016193 bytes)         | (5749958016193 bytes)           |            |

According to the test results, the SDH analysis scheme described in this paper can well realize the analysis of the SDH protocol in 10Gbps POS signal.

5. Conclusions
This paper takes 10Gbps POS signal as an example to illustrate the whole process of SDH protocol parsing based on FPGA. Through the simulation of Modelsim software and 12-hour continuous outsourcing test of IXIA tester, the results show that the SDH parsing scheme described in this paper can realize SDH protocol parsing perfectly on FPGA, and can provide some reference value for the development of SDH technology.

Acknowledgements
This work is project supported by the Natural Science Foundation of the Anhui Higher Education Institutions of China, KJ2019A0898 and KJ2019A0890.

References
[1] Zuo Z, Liu D S, Qin J Y, Huang Z P and Su S J 2019 IP over SONET/SDH link-layer processing with multiprocessor International Journal of Pattern Recognition and Artificial Intelligence 33(4) 1958006
[2] Aisawa S, Ono T and Tomizawa M 2011 40Gbit/s multi-lane distribution interface converter and its application to cost-effective optical transceiver for 40G SONET/SDH signals Asia Communications and Photonics Conference and Exhibition (11) 6210790
[3] Zhang X H, Deng B and Huang Y G 2016 Design of SDH service phone based on FPGA technology Optical Communication Technology 40(3) 36-38
[4] Ye B, Li T W, Zhang L J and Luo M 2010 Chip implementation for PDH over 622 Mb/s SDH/SONET mapper Acta Electronica Sinica 38(8) 1945-51
[5] Wang L Y, Wang J F, Wang J Z and Li Y 2012 Implementation of IP Packet Extraction from POS Signal Based on FPGA Radio Engineering 42(12) 13-15
[6] Duan M X 2014 The design principle and method of multirate POS test CPCO model Chinese Journal of Electron Devices (4) 617-621
[7] Bai Y and Chen J H 2015 Design and implement of an transmission scheme for SDH network management information Optical Communication Technology 39(9) 51-53
[8] Zhang X H, Deng B and Huang Y G 2016 Design of SDH service phone based on FPGA technology Optical Communication Technology 40(5) 36-38
[9] Wang X, Wang R, Lu L, Zhu Y, Zhang B F, Wu C X and Wei Y M 2015 Study on high precision time transfer based on SDH optical network Journal of Optoelectronics-Laser 26(2) 251-258
[10] Huang W Z 2019 Research on network time synchronization method based on SDH/PTN transmission network Lecture Notes in Electrical Engineering 463(1) 361-368