

Physical Coding Sub Layer Implementation of Common Public Radio Interface

T.Tirumala Rao¹, P.V.L.Suvarchala², M.Harikrishna Reddy³

¹Department of ECE, Amrita Sai Institute of Science and Technology, Paritala, Krishna (dt), A.P, India.

²Associate.proffesor Department of ECE, Amrita Sai Institute of Science and Technology, Paritala, Krishna (dt), A.P, India.

³ASIC Verification Engineer, Appex Semiconductors, Mahadevapura, Bangalore, Karnataka, India.

Email: tirumalाराothokala@gmail.com¹, suvarchala_pvl@yahoo.com², harikrishna426@gmail.com³

Abstract: - In radio communication systems for high speed data transmission and to avoid unbalanced code flow we use PCS. This paper mainly deals with encoding and decoding techniques used in Physical Coding Sub layer. In this PCS layer we use encoding, decoding, mux, mode select register and synchronization state machine. So, in the transmitter side we will deal with how to encode the data and what are the rules to follow to encode the data. Similarly, in the receiver side we will deal with how to decode the data and what are the rules to follow to decode the data. The proposed circuit is simulated in Modelism and the results obtained are presented in this paper.

Index Terms: - Common Public Radio Interface, Physical coding sub layer, 8B/10B algorithm, 16B/20B algorithm, 32B/40B algorithm, Synchronization State Machine, Multiplexer.

1. INTRODUCTION

Radio base station system is composed of two basic subsystems, Radio Equipment Control (REC) and Radio Equipment (RE). The interface between REC and Higher layers is Network interface and the interface between RE and Air is Air interface. The Common Public Radio Interface (CPRI) is an internal communication link between two radio base stations REC and RE. This interface supports different data rates ranging from 614.4Mbits/sec to 12165.12Mbits/sec. In this interface we have several layers, but main concentration is in bottom two layers which are physical layer(layer 1) and datalink layer(layer 2). In Cpri interface the bottom layer i.e., Physical Layer plays a crucial role. Inside this Physical Layer we have Physical Coding Sub layer (PCS) and SERDES. The main purpose of PCS is to encode in the transmitter side and to decode in the receiver side. SERDES will do parallel to serial conversion in the transmitter side and serial to parallel conversion in the receiver side.

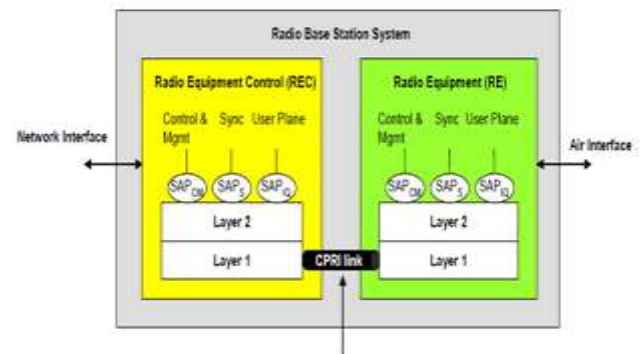


Figure 1. Common Public Radio Interface

2. PHYSICAL CODING SUBLAYER

In ISO-OSI we have seven layers, in those layers bottom layer is physical layer. Physical coding sublayer (PCS) is present inside the physical layer. It will perform the following actions.

- It will encode the incoming data during transmission.
- It will decode the data during reception.

2.1. PCS Transmitter:

In PCS transmitter we have encoder. In encoder we have encoding schemes and mux. Mux will perform the mode select operation.

The select line to the mux is of 2bit size. If the select line is

00: it will perform
8b to 10b encoding.
01: it will perform
16b to 20b encoding.
10: it will perform
32b to 40b encoding.
11: it will perform
default operation.
The default operation is 8b to 10b encoding.

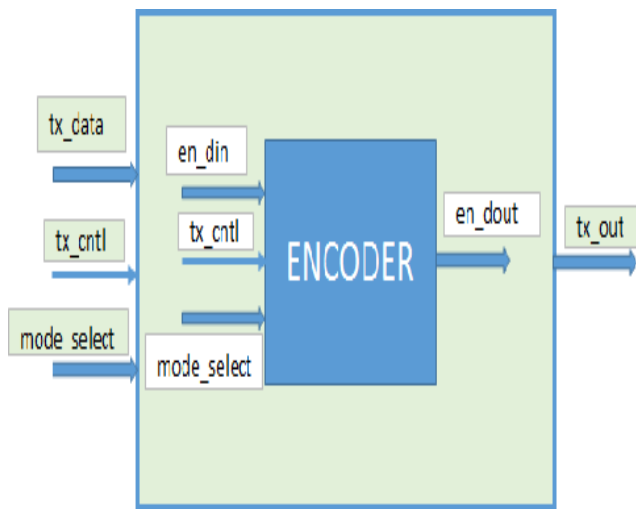


Figure 2. PCS transmitter block diagram

2.1.1 Encoder:

The main purpose of encoder is to maintain the enough no.of transitions between 0’s and 1’s in the serial line. While performing encoding operation we have to consider disparity rules. The encoding schemes present inside this encoder are 8b to 10b, 16b to 20b and 32b to 40b. These encoding schemes are used for high speed data transmission.

i. Disparity rules:

Running Disparity is a technique used in digital communication in order to maintain a DC balance in digital transmissions.

- Running disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the six-bit sub-block if the six-bit sub-block is 000111, and it is positive at the end of the four-bit sub-block if the four-bit sub-block is 0011.
- Running disparity at the end of any sub-block is negative if the sub-block contains more

zeros than ones. It is also negative at the end of the six-bit sub-block if the six-bit sub-block is 111000, and it is negative at the end of the four-bit sub-block if the four-bit sub-block is 1100.

- Otherwise, running disparity at the end of the sub-block is the same as at the beginning of the sub-block.

ii. 8b to 10b encoding:

In 8b to 10b encoding it will convert the 8bit incoming data to 10bit data by using the valid code group tables given in IEEE 802.3 Standard.

Table 1. Some valid data code groups

Code Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D0.0	00	000 00000	100111 0100	011000 1011
D1.0	01	000 00001	011101 0100	100010 1011
D2.0	02	000 00010	101101 0100	010010 1011
D3.0	03	000 00011	110001 1011	110001 0100
D4.0	04	000 00100	110101 0100	001010 1011
D5.0	05	000 00101	101001 1011	101001 0100
D6.0	06	000 00110	011001 1011	011001 0100
D7.0	07	000 00111	111000 1011	000111 0100
D8.0	08	000 01000	111001 0100	000110 1011
D9.0	09	000 01001	100101 1011	100101 0100

Table 2.valid special code groups

Code Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD –	Current RD +	Notes
			abcdei fghj	abcdei fghj	
K28.0	1C	000 11100	001111 0100	110000 1011	1
K28.1	3C	001 11100	001111 1001	110000 0110	1,2
K28.2	5C	010 11100	001111 0101	110000 1010	1
K28.3	7C	011 11100	001111 0011	110000 1100	1
K28.4	9C	100 11100	001111 0010	110000 1101	1
K28.5	BC	101 11100	001111 1010	110000 0101	2
K28.6	DC	110 11100	001111 0110	110000 1001	1
K28.7	FC	111 11100	001111 1000	110000 0111	1,2
K23.7	F7	111 10111	111010 1000	000101 0111	
K27.7	FB	111 11011	110110 1000	001001 0111	
K29.7	FD	111 11101	101110 1000	010001 0111	
K30.7	FE	111 11110	011110 1000	100001 0111	

NOTE 1—Reserved.

NOTE 2—Contains a comma.

iii. 16b to 20b encoding:

In 16b to 20b encoding scheme we will convert 16bit input data into 20bit data by using two 8b to 10b sub blocks.

iv. 32b to 40b encoding:

In 32b to 40b encoding scheme we will convert 32bit input data into 40bit data by using four 8b to 10b sub blocks.

2.2. PCS Receiver:

In PCS receiver we have decoder, state_machine, k_alignment_logic. In decoder we have decoding schemes and mux. Mux will perform the mode select operation.

The select line to the mux is of 2bit size. If the select line is

00: it will perform 10b to 8b decoding.

01: it will perform 20b to 16b decoding.

10: it will perform 40b to 32b decoding.

11: it will perform default operation.

The default operation is 10b to 8b decoding.

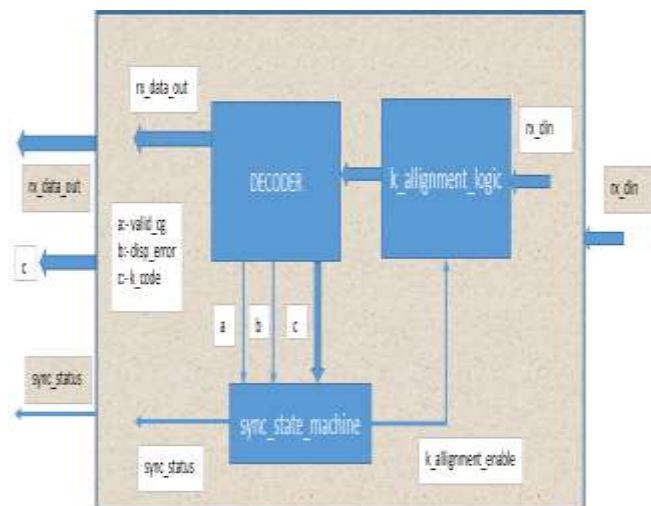


Figure 3: PCS receiver block diagram

2.2.1 Decoder:

The main purpose of the decoder is to decode the incoming data. In this we have three decoding schemes. In decoder output we have 3 status signals. They are k_code, valid_cg and disparity_error.

- If the input code group of the decoder is k28.1, k28.5 and k28.7 then k_code is 1, otherwise k_code is 0.
- If the input code group of the decoder is present in the code group table given in IEEE 802.3 section-3 clause-36 and produce valid decoded data depending on input disparity then that code group is valid_codegroup else it is invalid
- If the 10-bit code group has more ones than zeros or more zeros than ones then the disparity_output must be opposite to

disparity_input then that 10-bit code group has no disparity_error. Otherwise it has disparity_error.

- If the 10-bit code group is Neutral (no.of 1's = no.of 0's) then the disparity_output must be equal to disparity_input then that 10-bit code group has no disparity_error. Otherwise it has disparity_error.

i. 10b to 8b decoding:

In 10b to 8b decoding scheme we will convert 10bit data into 8bit data by using valid code group tables shown in figure 3 and figure 4.

ii. 20b to 16b decoding:

In 20b to 16b decoding scheme we will convert 20bit data into 16bit data by using two 10b to 8b sub blocks.

iii. 40b to 32b decoding:

In 40b to 32b decoding scheme we will convert 40bit data into 32bit data by using four 10b to 8b sub blocks.

2.2.2. Synchronization state machine:

It will monitor the consistency of the incoming data by detecting the pattern K D D D. For detecting the pattern it has the following steps.

- Initially upon reset we will go to loss_of_sync state and make sync_acquired signal 0. If we detect K in this state we will go to comma_detect_1 state. If not we will go to loss_of_sync state.
- In comma_detect_1 state if we detect D then we will go to comma_detect_2 state. If not we will go to loss_of_sync state.
- In comma_detect_2 state if we detect D then we will go to comma_detect_3 state. If not we will go to loss_of_sync state.
- In comma_detect_3 state if we detect D then we will go to sync_acquired_state and make sync_acquired signal 1. If not we will go to loss_of_sync state.
- In sync_acquired state if we detect continuous valid good code groups then we will be in current state. If we detect 3 continuous invalid code groups then we will go to loss_of_sync state.

Radio and Telecom etc. We can use this one where ever we required a physical layer.

REFERENCES

- [1] CPRI Specification V6.1 (2014-07-01).
- [2] IEEE 802.3-2008_section3.
- [3] IEEE-Verilog-LRM.
- [4] J. Bhasker, A Verilog HDL Primer, second edition.
- [5] Andrew S. Tanenbaum, COMPUTER NETWORKS, 4th edition, Pearson Education, Inc, 2003.
- [6] M. Morris Mano, Digital Design with an introduction to Verilog HDL, fifth edition.
- [7] John F. Wakerly, Digital Design Principles & Practices, Prentice-Hall, third edition.