Stateless 64B/66B PCS Coding for all 200G/Lane Breakout Interfaces

Eugene Opsasnick, Broadcom

IEEE P802.3dj Task Force

July 2023 Plenary

Supporters

- Mark Gustlin, Cisco
- Matt Brown, ()
- Kent Lusted, Intel
- Arthur Marris, Cadence
- Daniel Koehler, Synopsys
- Gary Nicholl, Cisco
- David Ofelt, Juniper Networks
- Shawn Nicholl, AMD
- Tom Huber, Nokia
- Xiang He, Huawei
- Adee Ran, Cisco

Introduction

- 64B/66B PCS Encoding
  - The 66b codes are defined in Clause 82, Figure 82-5* “64B/66B Block Formats”
    - These block codes are used for all BASE-R PHYs from 40GbE to 1.6TbE
  - State Diagrams are used to define the legal sequences of blocks
    - Figures 119-14* and 119-15* for 200BASE-R and 400BASE-R (also 800GBASE-R)
    - Figures 82-16 and 82-17 for 40GBASE-R and 100GBASE-R
    - Same as Figures 49-16 and 49-17 for 10GBASE-R
  - A “stateless” option that can be used in place of the state diagrams was adopted for the 800GBASE-R PCS in 802.3df
- The stateless and state-based options are interoperable
  - Using the same 66-bit codes, there are minor differences in error handling

* The coding table and state diagrams are shown in backup slides for reference.

Stateless 64b/66b Encode/Decode

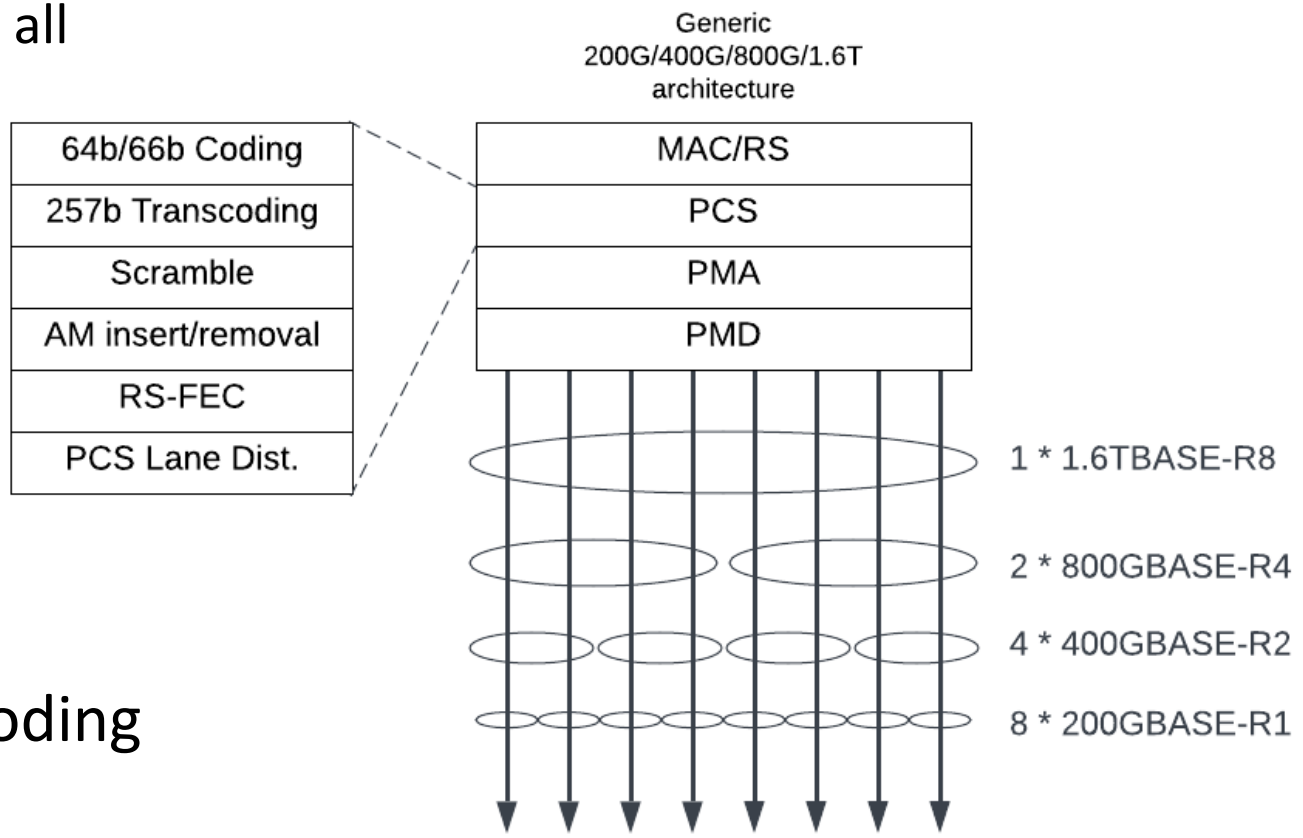
- As port speeds get faster:
  - Internal busses must get wider as it is harder to increase the ASIC clock frequency
    - For Stateful 64b/66b coding, each 8-byte block depends on the result of the previous block
    - It gets harder to implement the “stateful” 64b/66b Encode & Decode on the wider data busses
    - With the Stateless encoder/decoder, this per-block dependency is removed
    - See opsasnick_3df_01a_221005.pdf
  - MTTFPA is acceptable with RS[544, 514] based interfaces
    - See opsasnick_3df_01a_2212.pdf
    - State-based protection relies on redundancy in the 66b sync header that does not exist with transcoding
    - RS[544, 514] give better protect now than state diagrams for interfaces without FEC
- Current adoption of stateless 64b/66b encode/decode option
  - 802.3df adopted the stateless option for 800GbE PCS (CL 172)
  - 802.3dj also adopted the stateless option in the 1.6TbE PCS baseline
- Stateless and State-based encoding/decoding are fully inter-operable
  - It is a device choice to implement either one

Background

- 802.3df Draft 1.1, comment #42
  - Suggested adding the stateless 64b/66b option to Clause 119 for all 200/400GBASE-R PHYs for commonality with the 800GBASE-R PCS
  - The 802.3df ballot resolution committee deemed general updates to Clause 119 to be “out of scope”
    - Only changes related to the PHYs/PMDs being added are in scope
    - 802.3df only adds a single new PHY (400GBASE-DR4-2) based on 100G/lane
- For 802.3dj
  - All new PHYs/PMDs based on 200G/lane are in scope
  - Can now update Clause 119, 200/400G PCS, with respect to these new PHYs

200GbE – 1.6TbE Implementations on 200G/lane

- For 200 Gb/s per lane Interfaces
  - Breakout applications usually support all port speeds from 200GbE to 1.6TbE
- PCS 64b/66b options
  - Baseline for 1.6TBASE-R PCS
    - Stateful or stateless
  - CL 172: 800GBASE-R PCS
    - Stateful or stateless
  - CL 119: 200/400GBASE-R PCS
    - Currently stateful only
- For a common stateless 64b/66b Coding
  - Requires CL 119 changes to allow the stateless 64b/66b option for 200G/lane PHYs



802.3df Draft 2.1, Stateless Encoder Rules

172.2.4.1.2 Stateless encoder

The stateless encoder generates 66-bit blocks based only on the current and preceding 800GMII transfers. Each 800GMII transfer is mapped into a 72-bit vector tx_raw<71:0> (see 172.2.6.2.2). The encoder shall encode each tx_raw<71:0> to a 66-bit block tx_coded<65:0> according to the rules in Table 172–1. Constants LBLOCK_T and EBLOCK_T are defined in 172.2.6.2.1. Variables reset, tx_raw, and tx_coded are defined in 172.2.6.2.2. Functions T_TYPE and ENCODE, and the block types are defined in 172.2.6.2.3.

Table 172–1—PCS stateless encoder rules

reset	T_TYPE(tx_raw_{i-1})[a]	T_TYPE(tx_raw_i)[b]	Resulting tx_coded
1	any block type	any block type	LBLOCK_T
0	C or T	C	ENCODE(tx_raw_i)
0	C or T	S	ENCODE(tx_raw_i)
0	S or D	D	ENCODE(tx_raw_i)
0	S or D	T	ENCODE(tx_raw_i)
0	any combination not listed above		EBLOCK_T

[a] tx_raw_{i-1} is the 72-bit vector that immediately precedes tx_raw_i.
[b] tx_raw_i is the 72-bit vector that is being encoded.

802.3df Draft 2.1, Stateless Decoder Rules

172.2.5.9.2 Stateless decoder

The stateless decoder generates 800GMII transfers based only on the current and preceding 66-bit blocks. The decoder shall decode each 66-bit block rx_coded<65:0> to a 72-bit vector rx_raw<71:0> (see 172.2.6.2.2) according to the rules in Table 172–4. Constants LBLOCK_R and EBLOCK_R are defined in 172.2.6.2.1. Variables reset, rx_raw, and rx_coded are defined in 172.2.6.2.2. Functions R_TYPE and DECODE, and the block types are defined in 172.2.6.2.3.

Table 172–4—PCS stateless decoder rules

reset	R_TYPE(rx_coded_{i-1})[a]	R_TYPE(rx_coded_i)[b]	Resulting rx_raw
1	any block type	any block type	LBLOCK_R
0	any block type	E	EBLOCK_R
0	E	any block type	EBLOCK_R
0	any combination not listed above		DECODE(rx_coded_i)

[a] rx_coded_{i-1} is the 66-bit block that immediately precedes rx_coded_i.
[b] rx_coded_i is the 66-bit block that is being decoded.

Summary

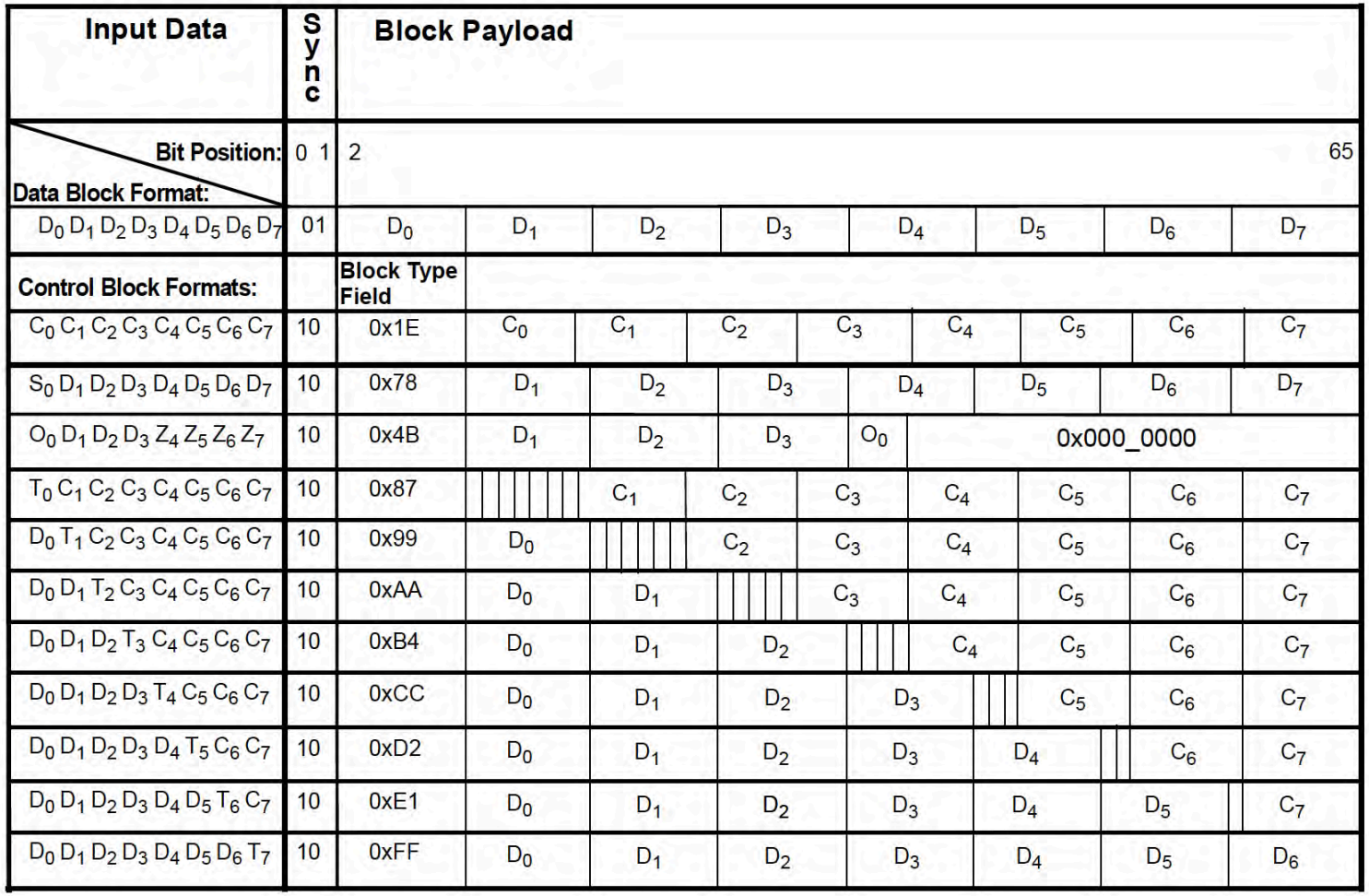
- Suggested course of action
  - Specify stateless 64b/66b encode and decode as an option in Clause 119
    - As defined in 802.3df D2.1 172.2.4.1.2 and 172.2.5.9.2 and shown on slides 7 and 8, for all 200G/lane PHY/PMDs
    - Allows for a single stateless 64b/66b design for common breakout applications
  - This option would not apply to PHY/PMDs not defined in 802.3dj which use Clause 119
    - Does not affect 200GbE and 400GbE PHY/PMDs already defined (over 100G/lane or 50G/lane)

- A straw poll is planned to gauge support

Thank You

Backup/Reference slides follow

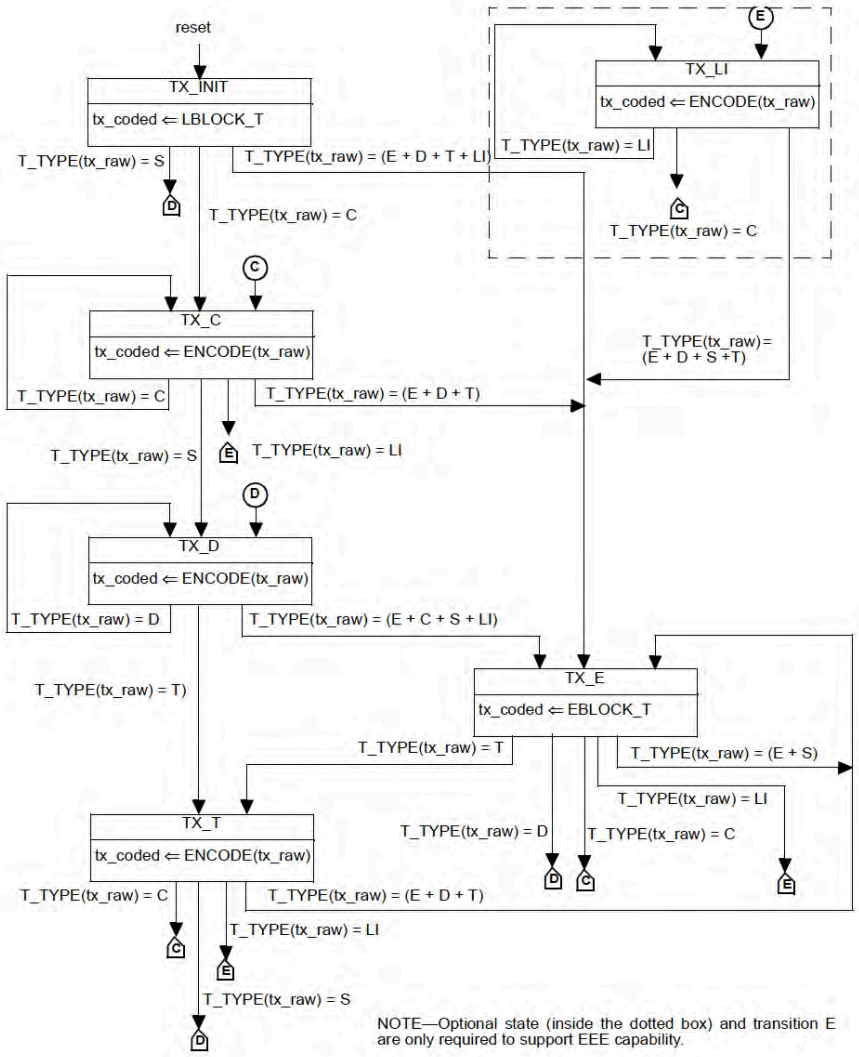
802.3 Figure 82-5: 64B/66B Block Formats



Input Data	Sync	Block Payload							
Bit Position: / Data Block Format:	0 1	2							65
$D_0\ D_1\ D_2\ D_3\ D_4\ D_5\ D_6\ D_7$	01	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
Control Block Formats:		**Block Type Field**							
$C_0\ C_1\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7$	10	0x1E	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
$S_0\ D_1\ D_2\ D_3\ D_4\ D_5\ D_6\ D_7$	10	0x78	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$O_0\ D_1\ D_2\ D_3\ Z_4\ Z_5\ Z_6\ Z_7$	10	0x4B	D_1	D_2	D_3	O_0	0x000_0000		
$T_0\ C_1\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7$	10	0x87		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$D_0\ T_1\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7$	10	0x99	D_0		C_2	C_3	C_4	C_5	C_6	C_7
$D_0\ D_1\ T_2\ C_3\ C_4\ C_5\ C_6\ C_7$	10	0xAA	D_0	D_1		C_3	C_4	C_5	C_6	C_7
$D_0\ D_1\ D_2\ T_3\ C_4\ C_5\ C_6\ C_7$	10	0xB4	D_0	D_1	D_2		C_4	C_5	C_6	C_7
$D_0\ D_1\ D_2\ D_3\ T_4\ C_5\ C_6\ C_7$	10	0xCC	D_0	D_1	D_2	D_3		C_5	C_6	C_7
$D_0\ D_1\ D_2\ D_3\ D_4\ T_5\ C_6\ C_7$	10	0xD2	D_0	D_1	D_2	D_3	D_4		C_6	C_7
$D_0\ D_1\ D_2\ D_3\ D_4\ D_5\ T_6\ C_7$	10	0xE1	D_0	D_1	D_2	D_3	D_4	D_5		C_7
$D_0\ D_1\ D_2\ D_3\ D_4\ D_5\ D_6\ T_7$	10	0xFF	D_0	D_1	D_2	D_3	D_4	D_5	D_6	

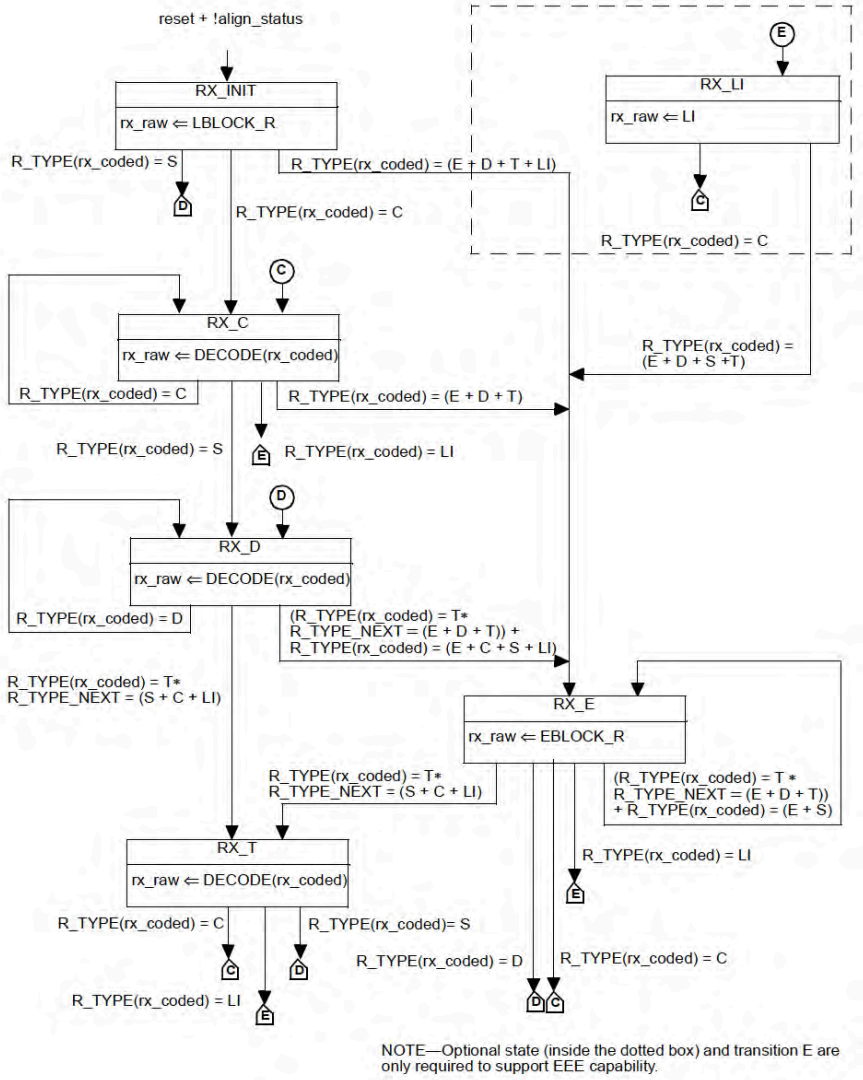
- (D)ata Block
  - Sync=01
- (C)ontrol Block
  - Sync=10
  - BT=0x1E or 0x4B
- (S)tart Block
  - Sync=10
  - BT=0x78
- (T)erminate Block
  - Sync=10
  - BT=0x87, 0x99,0xAA,0xB4,0xCC, 0xD2,0xE1, or 0xFF
- (E)rror Block
  - Anything else

802.3 Figure 119-14: Transmit State Diagram



- After Reset, the “normal flow” is:
  1. State TX_C
     - One or more IDLE Control blocks
  2. State TX_D
     - Transmit a (S)tart block
  3. State TX_D * n
     - Several blocks of (D)ata
  4. State TX_T
     - Transmit a (T)erminate block
  - Repeat sequence 1-4
    - Can skip #1 and directly send (S) block in step #2
- In TX_E state:
  - (S)tart blocks are not propagated
  - This restriction on (S)tart blocks is not necessary

802.3 Figure 119-15: Receive State Diagram



- After Reset, the “normal flow” is
  - Same as TX flow
  1. State RX_C (Idles)
  2. State RX_D (Start)
  3. State RX_D (Data)
  4. State RX_T (Terminate)

     Repeat
- In RX_E state:
  - Same unnecessary restriction on (S)tart blocks
- Additional restrictions on (T)erminate blocks
  - The next block after the (T) must be a correct block type: (S) or (C) or (LI)
  - This restriction on (T) blocks is not necessary

Additional Historical References

- The 64b/66b code and PCS State diagrams were introduced in 802.3ae for 10GbE without FEC, see:
  - walker_1_0300.pdf, walker_1_0500.pdf, and walker_1_0700.pdf

- Considerations
  - CRC32 has a hamming distance of 4
    - Up to 3 bits in error are always detected
  - The state diagrams make use of the 2-bit sync header in the 66-bit code to make sure there are at least 4 bit errors before an “undetectable” packet framing error occurs.
    - To maintain the MTTFPA
  - However, with the addition of RS-FEC in more recent projects, the protection of the state diagrams is diminished since the redundant sync header bits are removed by the 257b transcoding.
  - The RS[544] FEC gives better protection now than the state diagrams and hamming distance of 4.