PCS SD

Duane Remein (Huawei)
Glen Kramer (Broadcom Ltd)

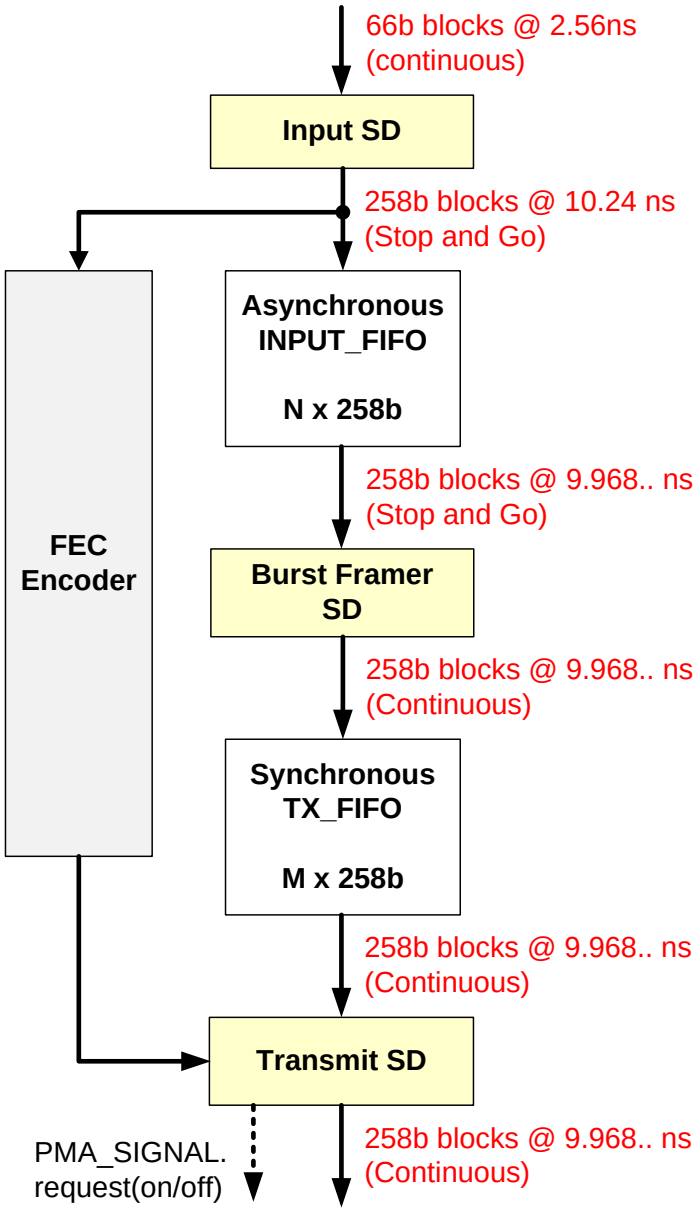
Supporters

- Gaobo (Huawei)

Outline

- PCS diagrams and definitions
  - Concept
  - Function Block Diagram
  - State Diagrams
    - PCS Input SD
    - PCS Framer SD
    - PCS Transmit SD
  - Definitions

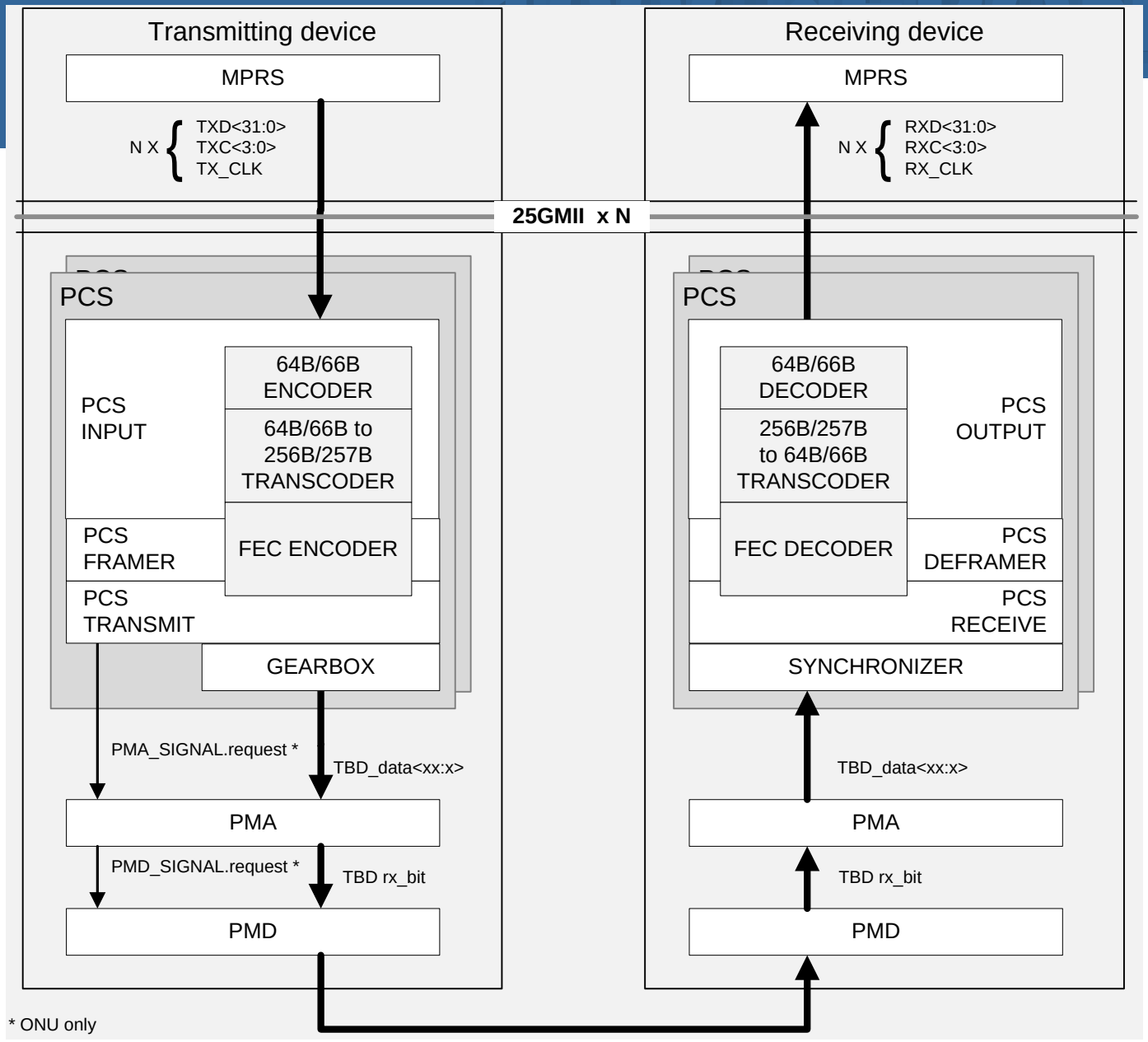
Concept



- **Input SD**
  - Discards Inter-Burst Idles and Rate Adjustment EQs.
  - The rest - Transcode, Scramble, Pass to FEC encoder
- **INPUT_FIFO**
  - Depth = SP_LENGTH.
  - INPUT_FIFO grows while Burst Framing SD inserts the SP sequence, drains otherwise
- **Burst Framer SD**
  - Pauses the INPUT_FIFO to insert SP, SBD, EBD.
  - Also, inserts 10 parity placeholder blocks after every FEC payload transferred from the INPUT_FIFO
- **TX_FIFO**
  - Depth = sufficient to generate parity
  - Constant FIFO delay (one block is added and one is removed on each clock)
- **Transmit SD**
  - Replaces parity placeholder blocks with actual parity values (ready by now)
  - Generates LaserOn signal when sees start of SP
  - Generates LaserOff signal when sees IBI

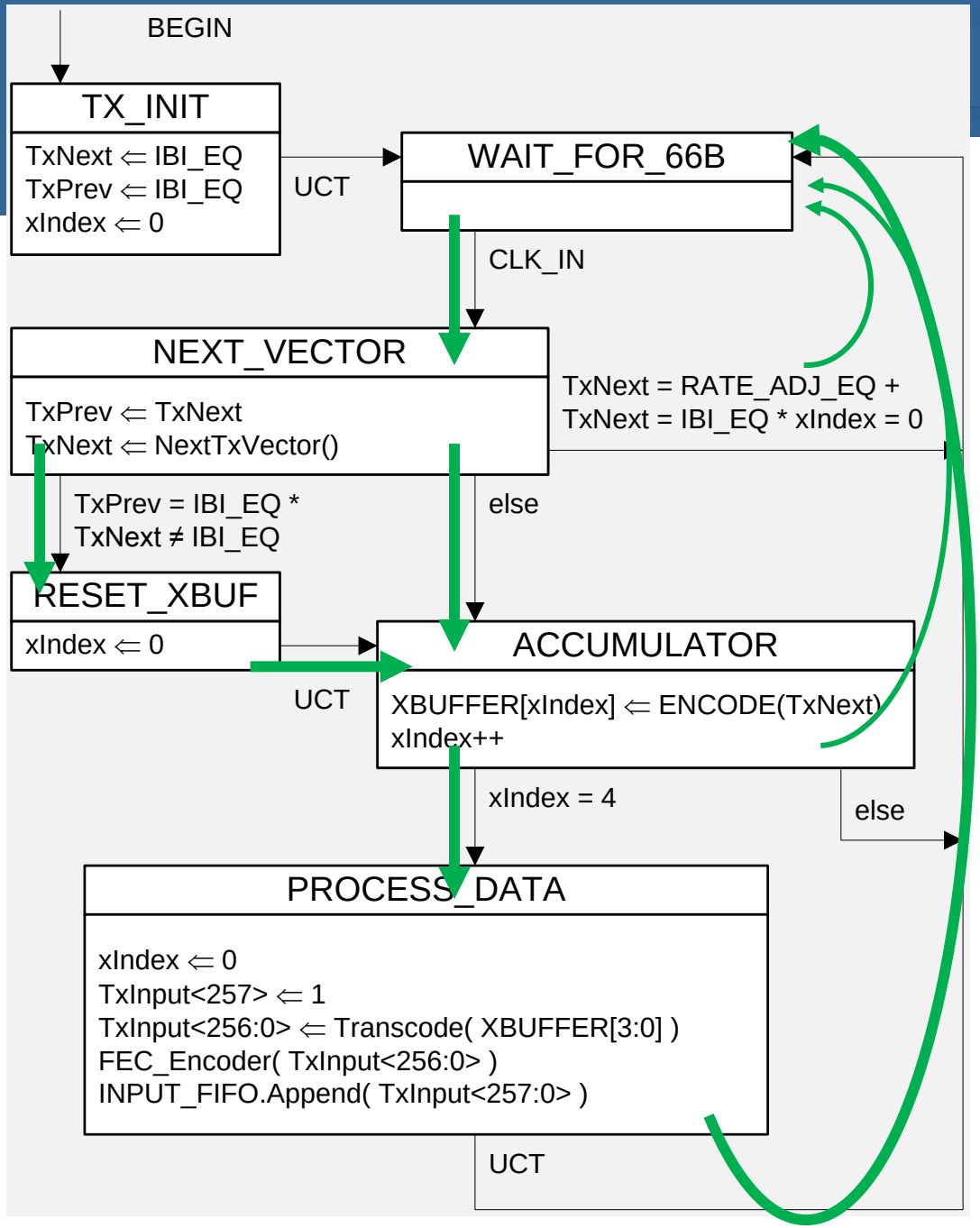
Block Diagram

- Input SD
  - 64B/66B encoding
  - 256B/257B transcoding
  - Scrambler
  - FEC handoff
- Framer SD
  - Indicates burst boundaries
  - Adds Sync Pattern & End of Burst Delimiter
  - Indicates position of Parity word
- Transmit SD
  - Multiplexes Payload & Parity
  - Sets PMA_SIGNAL.request



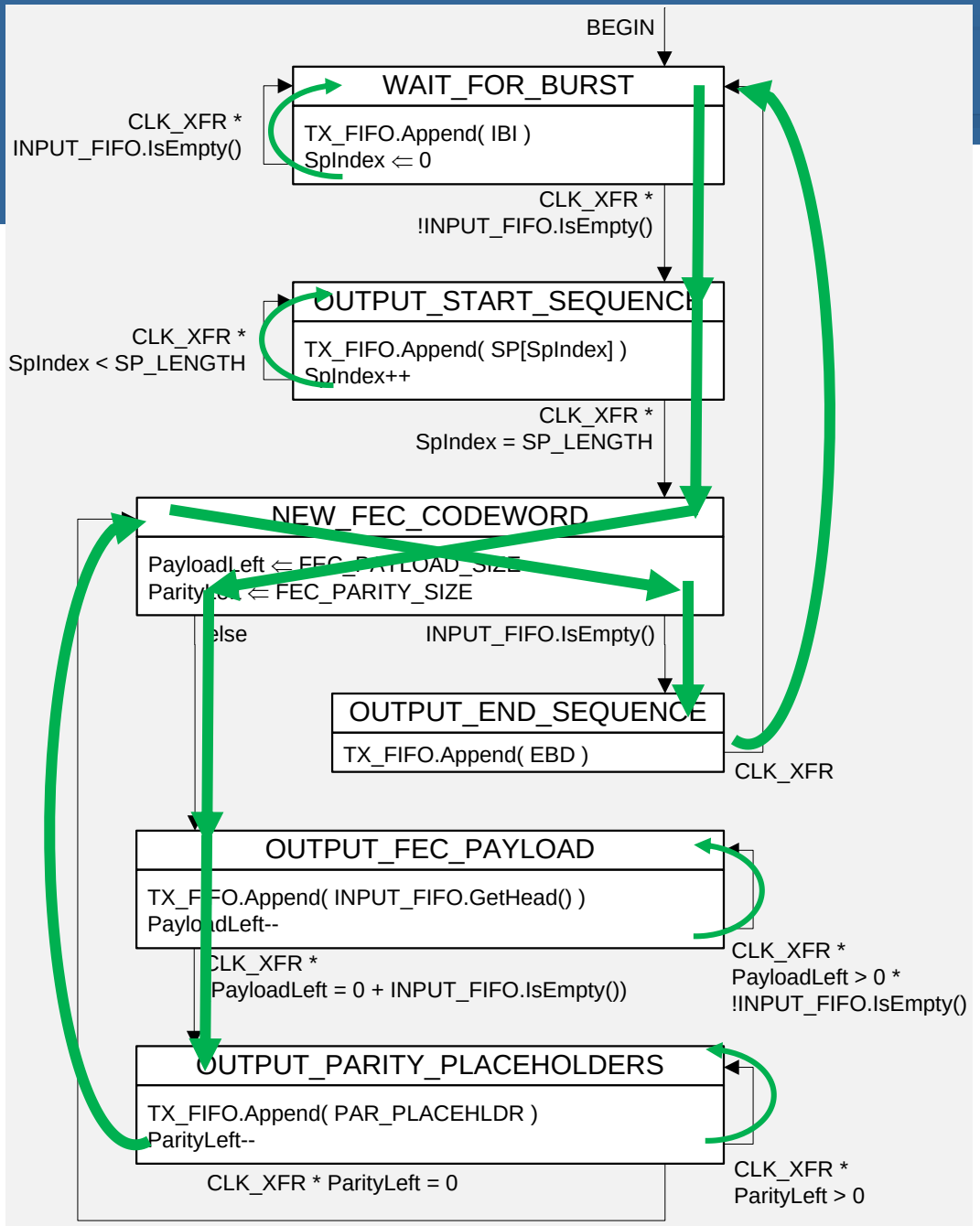
PCS Input SD

- Replaces Transmit/Encoder SD
- Clocked each EQ
- Discards RATE_ADJ_EQs and IBI_EQs (excepting those needed to complete a 257b block)
- Resets transcoding boundary at the beginning of a burst
- Encodes, transcodes, & scrambles data
- Hands off data to FEC engine
- Queues data in INPUT_FIFO with indication it is scrambled data from the Input process (bit 257 = 1)



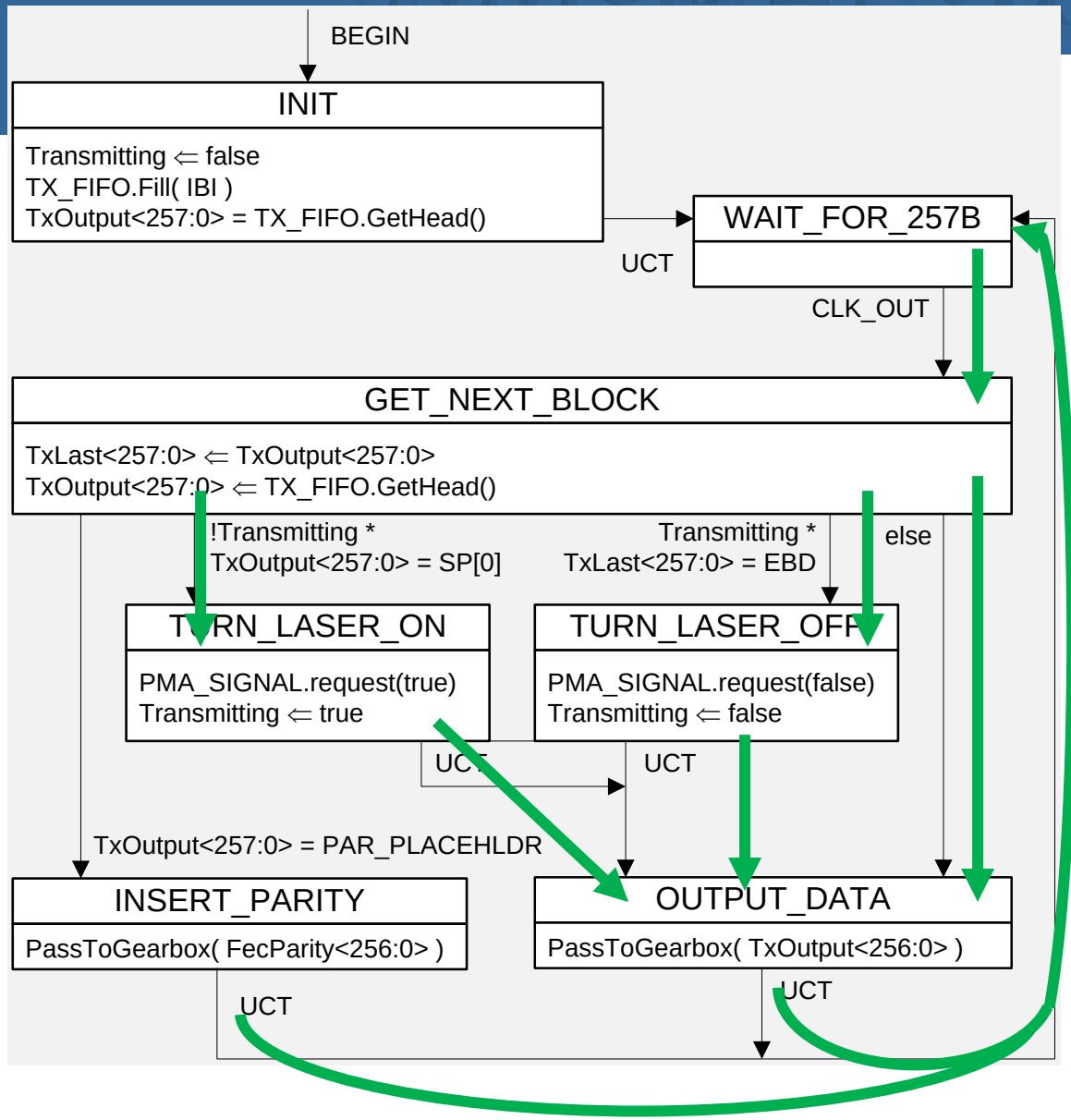
PCS Framer SD

- Clocked each 257b @25.78125Hz
- IDs beginning of burst
  - INPUT_FIFO not empty
- Queues Sync Pattern
- Queues FEC Payload
- Queues FEC Parity
- IDs end of burst
  - INPUT_FIFO is empty
- Queues last FEC Parity word
  - Possibly shortened CW
- Queues End-of-Burst Delimiter (EBD)



PCS Transmit SD

- Clocked each 257b @25.78125Hz
- Retrieves one 258b block from OUTPUT_FIFO
- Determines:
  - Beginning of burst
    - Sets PMA_SIGNAL.request true
  - Parity insertion point
  - End of burst
    - Sets PMA_SIGNAL.request false



New constants, variables, functions (1 of 4)

Note to Ed: include the following paragraph before Variables/Constants/Functions sections.

Various variables and buffers in the PCS are structured as 258-bit wide vectors with bits 0 through 256 holding one line-coding unit (a 257-bit block) and bit 257 conveying the origin of the block to be either the PCS Input process (bit 257 is one) or the PCS Framer process (bit 257 is zero). The value of bit 257 being one implies that the 257-bit block has been transcoded and scrambled.

Constants

EBD TYPE: 258-bit vector
Value: {msb = 0, the remaining bits TBD}
The EBD constant holds the value of the end of burst delimiter.

FEC_PARITY_SIZE TYPE: Integer
Value: {10 TBC}
The FEC_PARITY_SIZE constant indicates the size of the Parity portion of a FEC codeword in units of 257 bits.

FEC_PAYLOAD_SIZE TYPE: Integer
Value: {56 TBC}
The FEC_PAYLOAD_SIZE constant indicates the size of the Payload portion of a FEC codeword in units of 257 bits.

IBI Type: 258-bit vector
Value: {msb = 0 the remaining bits TBD }
The IBI constant holds the value of the inter burst idle pattern.

IBI_EQ See 143.xxx.xxx
Note to Ed. add definition as shown below in the updated MPRS Input SD section
TYPE: 72-bit vector
Value: 0xFF 0A 0A 0A 0A 0A 0A 0A 0A
The IBI_EQ constant indicates to the PCS that the burst is terminated.

PAR_PLACEHLDR TYPE: 258-bit vector
Value: {msb = 0 the remaining bits TBD}
The PAR_PLACEHLDR constant represents the value of a 258-bit vector inserted into the data stream by the PCS Framer Process in order to reserve the location where FEC Parity and the FEC Delimiter is to be inserted into the data stream by the PCS Transmit Process.
Note to Ed: remove definition in CL143

New constants, variables, functions (2 of 4)

RATE_ADJ_EQ See 142.xxx.xxx

Note to Ed. add definition as shown below in the updated MPRS Input SD section

TYPE 72-bit vector

Value: 0xFF 09 09 09 09 09 09 09 09

The value of an EQ which is used as a placeholder to allow for rate differences between the MAC and the PHY layers.

Variables

CLK_IN TYPE: Boolean

The clear on read variable CLK_IN is set to true on each falling edge of the 25GMII clock.

CLK_OUT TYPE: Boolean

The clear on read variable CLK_OUT is set to true once for each 257-bits of data output by the PMD.

CLK_XFR TYPE: Boolean

The clear on read variable CLK_XFR is set to true once for each 257-bits of data output by the PMD.

INPUT_FIFO[] TYPE: array of 258-bit vectors

The INPUT_FIFO receives data from the Input process and hands it off to the Framer process. Its primary function is to absorb data while the PCS is transmitting burst overhead or FEC Parity. This FIFO holds at most SP_COUNT elements.

TX_FIFO[] TYPE: array of 258-bit vectors

The TX_FIFO[] holds information queued by the Framer process for output by the PCS and enforces a fixed delay that is implementation dependent. The fixed delay ensures the PHY has sufficient time to generate FEC Parity given that the Framer process inserts SP_LENGTH 257-bit blocks at the beginning of the burst. The length of the TX_FIFO[] is defined as:

MAX{ FEC_DELAY - SP_LENGTH, 2 }

ParityLeft TYPE: Integer

The ParityLeft variable indicates the number of 257-bit Parity vectors needed to complete the current FEC codeword being processed by the PCS Framer.

PayloadLeft TYPE: Integer

The PayloadLeft variable indicates the number of 257-bit payload vectors needed to complete the current FEC codeword being processed by the PCS Framer.

New constants, variables, functions (3 of 4)

Variables (cont)

SP[] TYPE: array of 258-bit vectors
An array of SP_LENGTH elements. Each element consists of MSB 0 and the 257-bit blocks as provisioned for SP1, SP2, SP3 by the MPCP.

SP_LENGTH TYPE: integer
The SP_LENGTH variable represents the length of the synchronization pattern as determined by the most recent settings of SP1_RepeatCount, SP2_RepeatCount and SP3_RepeatCount received at the MPMC layer.

SpIndex TYPE: integer
The SpIndex variable is a pointer into the SP[] array that indicates which 258-bit vector from the array should be sent to the TX_FIFO.

Transmitting TYPE: Boolean
The Transmitting variable indicates whether the device is transmitting or not.

TxInput TYPE: 258-bit vector
This variable holds one transcoded 257-bit vector prepended with a binary one indicating the 257-bit block originated in the Input process.

TxNext TYPE: 72-bit vector
The next 72-bit vector to be processed by the Input process.

TxLast TYPE: 258-bit vector
This variable holds the 258-bit that was read from TX_FIFO in the previous CLK_OUT cycle.

TxOutput TYPE: 258-bit vector
This variable holds one 258-bit vector retrieved from the OUTPUT_FIFO

TxPrev TYPE: 72-bit vector
This variable holds one 72-bit vector received by the Input Process from the 25GMII in the previous CLK_IN cycle.

XBUFFER[] TYPE: array of 66-bit vectors
This buffer holds four 66-bit vectors of 64B/66B encoded data to be transcoded into one 257-bit block.

xIndex TYPE: Integer
An index into the XBUFFER indicating the number of encoded vectors contained in the buffer that are ready to be transcoded.

New constants, variables, functions (4 of 4)

Functions

F.Append(v) This function adds the vector v to the input of FIFO F.

F.Fill(v) This function writes vector v, to each element of FIFO F.

F.GetHead() This function returns the oldest (head) element in the FIFO F, and removes that element from the FIFO, decreasing the length by one.

F.IsEmpty() This function returns true if the FIFO F is empty (has no elements), otherwise the function returns false.

ENCODE(v) This function performs 64B/66B encoding of a 72-bit vector v per 49.2.13.2.3 and returns the result.

FEC_Encode(v) This function passes a 257-bit vector v to the FEC engine for encoding.

NextTxVector() This function returns a 72-bit vector carrying a single EQ as shown in Figure 143-2. The vector is constructed from the data received from 25GMII over two subsequent 36-bit transfers: the first transfer is on rising TX_CLK25 edge and the second transfer is on the falling TX_CLK25 edge.

PassToGearbox(v) This function passes a 257-bit vector v to the Gearbox for outputting to the PMA.

Transcode() This function performs 64B/66B to 256B/257B transcoding and scrambling per 91.5.2.5 and returns the result. It takes an array of four 66-bit blocks a[4] as an argument and returns a 257-bit vector.

Thank you

Questions?
Comments?

Motion #?

Move to accept the figures, state diagrams, and definitions on slides 5-12 of remein_3ca_3a_0518.pdf and instruct the Editor to include in the draft 1.1.

Moved: Duane Remein

Second:

For: ____

Against: ____

Abstain: ____

Motion: Passes/Fails

Motion is Technical (≥75%) Procedural(<50%)