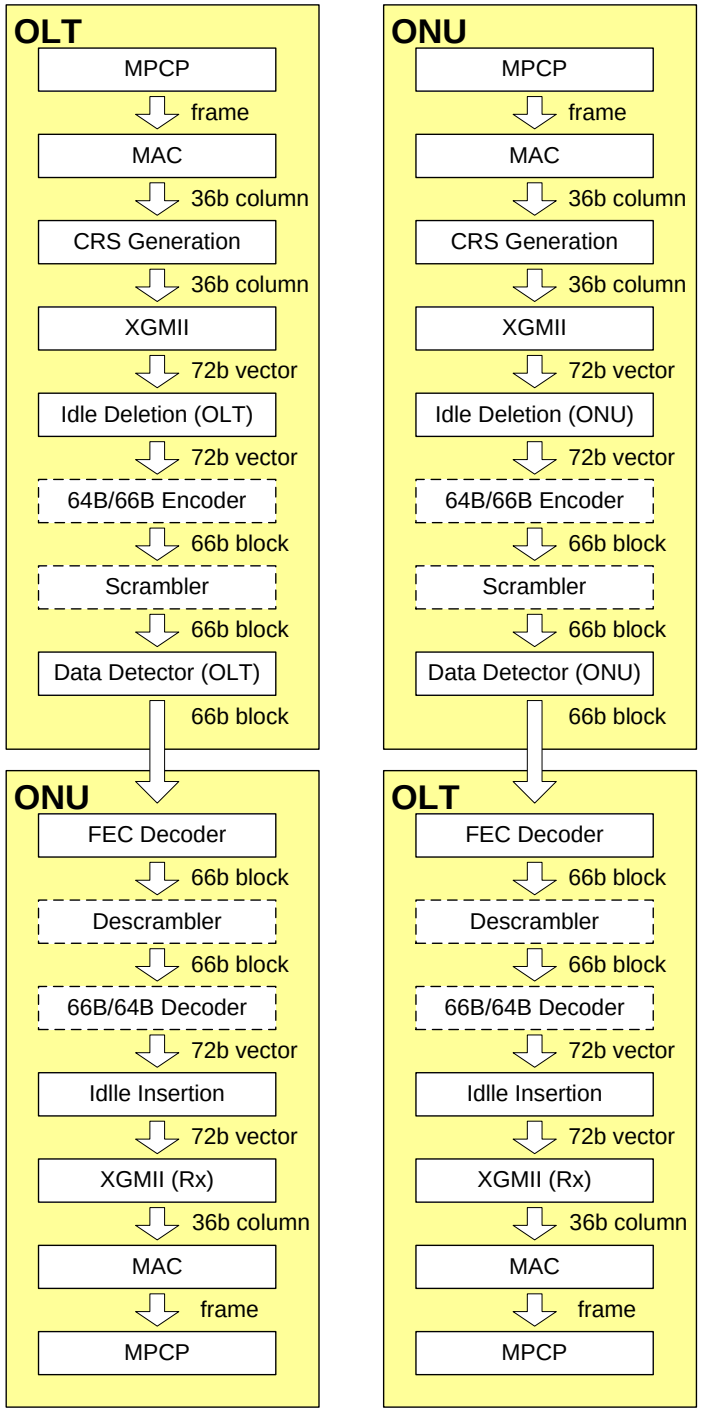
Data Detector: Issues Uncovered During Simulation

Glen Kramer

Introduction

- Model was created to analyze the delay variability (MPCP-to-MPCP) in downstream and upstream directions
- While creating the model, several problems with the Data Detector state diagram were uncovered



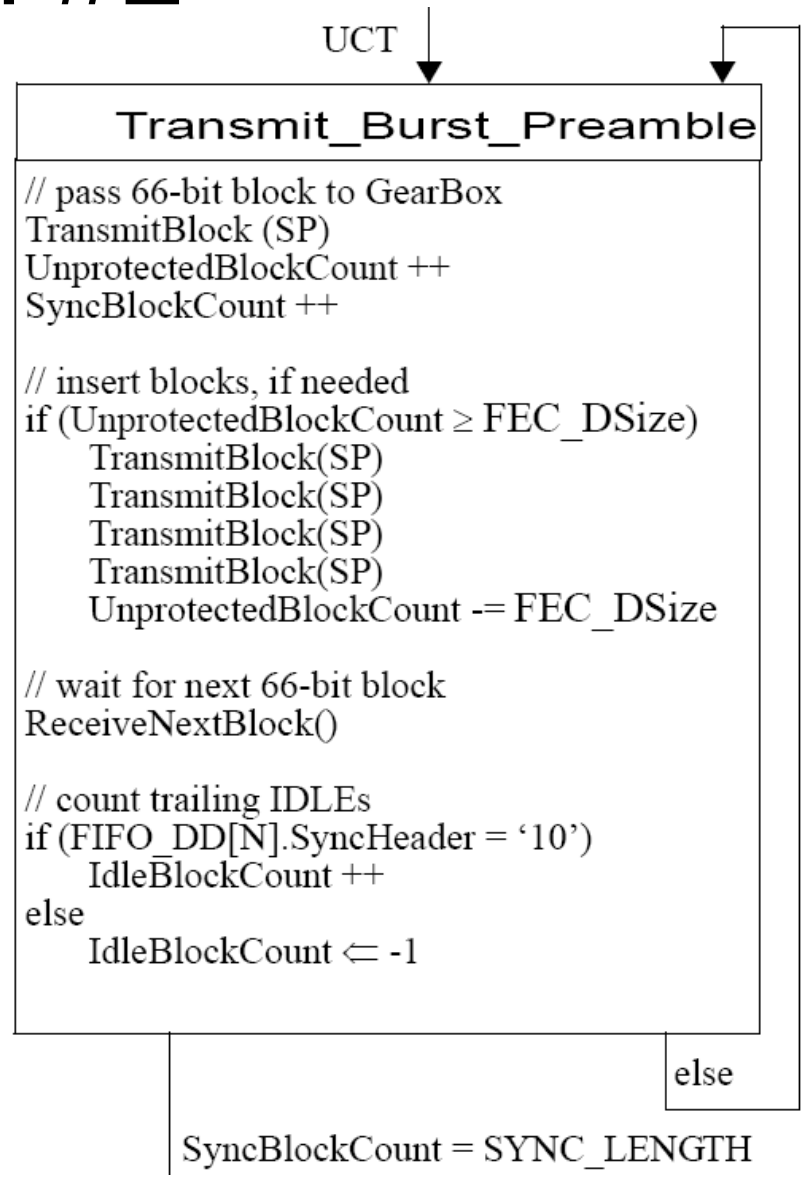
Problem #1

- ReceiveNextBlock() function is said to be a blocking function, i.e., “it does not return until a next block becomes available and is stored in the tail position in the FIFO_DD…”
- But 66-bit blocks arrive not periodically, but with big gaps, due to deleted idles
  - After 2000-byte frame, the gap may be as big as CEILING((2000 + preamble) /216) * 4 = 40 blocks
- During such gaps, the Data Detector will not output any data to the GearBox

Problem #2

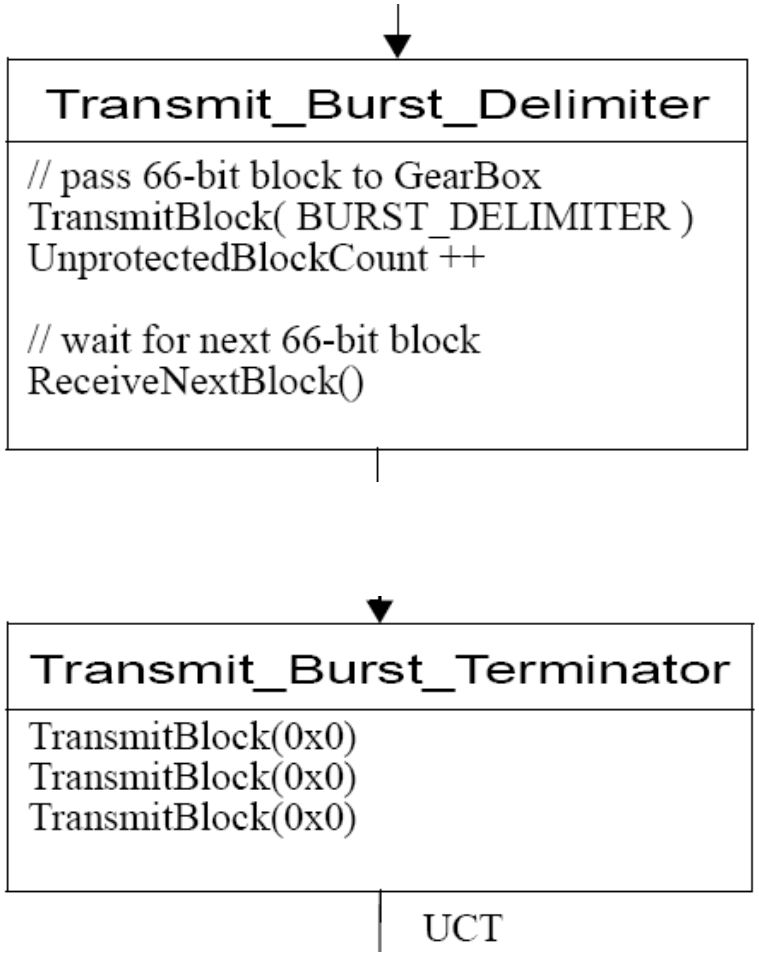
- In state TRANSMIT_BURST_PREAMBLE, number of transmitted blocks may exceed SyncBlockCount, since additional blocks are inserted, but are not accounted for.

- Longer burst preamble leads to
  - Potential collisions due to Longer ONU's transmission
  - Overflow of the FIFO_DD Queue, since more data will accumulate waiting for longer syncTime to pass.



Problems #3 and #4

- Idles are not counted in state Transmit_Burst_Delimiter
- Idle Blocks and Unprotected Blocks should also be counted in state Transmit_Burst_Terminator
- In Transmit_Burst_Terminator state, 3 blocks should also be removed from FIFO_DD, or else outgoing rate will not be correct.



What happened?

- The Initial Data Detector state machine was developed with the assumptions that data will arrive from the Scrambler with a reduced, but constant rate.
- The basic idea was that the Data Detector will use only the incoming data to properly recover output rate.
- Many other decisions were made since then and many other state machines introduced (CRS generation, Idle deletion), but the Data Detector was never updated.

Proposed Solution

- Make bursty input process independent of the constant rate output process.
  - Input process is clocked by the arriving blocks from the Scrambler.
  - Output process runs at the constant rate corresponding to the TX_CLK from the XGMII

State diagrams are presented next

92.0.0.0.1 Constants

BURST_DELIMITER
TYPE: 66-bit unsigned
A 66-bit value used to find the beginning of the first FEC codeword in the upstream burst.
Value: 0x 4 97 BA C4 69 F0 4C 88 FD (transmission bit sequence: 10 11101001 01011101 00100011 10010110 00001111 00110010 00010001 10111111)

FEC_DSize
See Subclause 92.2.2.1.1.

FEC_PSize
See Subclause 92.2.2.1.1.

SP
Type: 66-bit unsigned
The burst mode synchronization pattern.
Value: 0x4 55 55 55 55 55 55 55 55 (transmission bit sequence 1010...)

TERMINATOR_LENGTH
Type: 8-bit unsigned
Number of blocks containing zeroes that are transmitted at the end of each burst.
Value: 3

92.0.0.0.2 Variables

CLK
TYPE: boolean
This boolean is true on every negative edge of TX_CLK (See @@46.3.1.1@@) and represents instances of time at which a 66-bit block should be passed from Data detector to the GearBox. This variable is reset to false upon read.

DelayBound
This variable is defined in @@92.2.2.1.2@@.

FIFO_DD
TYPE: Array of 66-bit unsigned elements
A FIFO array used to store tx_coded<65:0> blocks while the parity is inserted and while burst preamble is generated (at the ONU only).

FifoSize
TYPE: 16-bit unsigned
Variable representing a number of elements stored in FIFO_DD.

SyncLength
TYPE: 16-bit unsigned
Required number of sync blocks per burst. The value of this variable is derived from the syncTime parameter passed from the OLT to the ONU. See @@93.3.3.2@@ for details.

Transmitting
TYPE: boolean
Boolean variable indicating whether ONU is transmitting or not. At the ONU, the default value of Transmitting is false. At the OLT, this variable is always set to true.

tx_coded<65:0>
TYPE: 66-bit unsigned
66-bit block containing the output of the scrambler. The format for this vector is shown in Figure 49–7. The leftmost bit in the figure is tx_coded<0> and the rightmost bit is tx_coded<65>.

tx_coded_out<65:0>
TYPE: 66-bit unsigned
66-bit block containing the output of Data Detector being passed to the Gearbox. The format for this vector is shown in Figure 49–7. The leftmost bit in the figure is tx_coded<0> and the rightmost bit is tx_coded<65>.

92.0.0.0.3 Functions

RemoveFifoHead()
This function removes the the first block in FIFO_DD and decrements the variable FifoSize by 1.

```
RemoveFifoHead()
   {
      // shift FIFO_DD forward
      FIFO_DD[0] = FIFO_DD[1]
      FIFO_DD[1] = FIFO_DD[2]
         ...
      FIFO_DD[FifoSize-2] = FIFO_DD[FifoSize-1]
      FifoSize --
   }
```

92.0.0.0.4 Messages

PMD_SIGNAL.request(tx_enable)
This primitive is used to turn the laser on and off at the PMD sublayer. In the OLT, this primitive shall always take the value ON. In the ONU, the value of this variable is controlled by the Data detector state diagram (see Figure 92–18).

SCRAMBLER_UNITDATA.request(tx_coded<65:0>)
A primitive generated by the SCRAMBLER transmit process conveying the next 66-bit block to be transmitted.

SUDR
Alias for SCRAMBLER_UNITDATA.request(tx_coded<65:0>).

92.0.0.0.5 Counters

IdleBlockCount
TYPE: 32-bit unsigned
The number of consecutive non-data blocks ending with the most recently received block. The non-data blocks are represented by sync header 10 (binary).

ParityBlockCount
TYPE: 8-bit unsigned
The number of parity blocks transmitted in a current FEC codeword. After reaching the full parity size (FEC_PSize=4), this counter is reset to 0.

ProtectedBlockCount

TYPE: 8-bit unsigned

The number of blocks added to a payload of a current FEC codeword. After reaching the full payload size (FEC_DSize = 27), this counter is reset to 0.

SyncBlockCount

TYPE: 16-bit unsigned

The number of synchronization blocks transmitted in current burst.

92.0.0.0.6 State diagrams

The OLT and the ONU shall implement the Data Detector input process as depicted in Figure 92–1. The OLT shall implement the Data Detector output process as depicted in Figure 92–18. The ONU shall implement the Data Detector output process as depicted in Figure 92–18

.

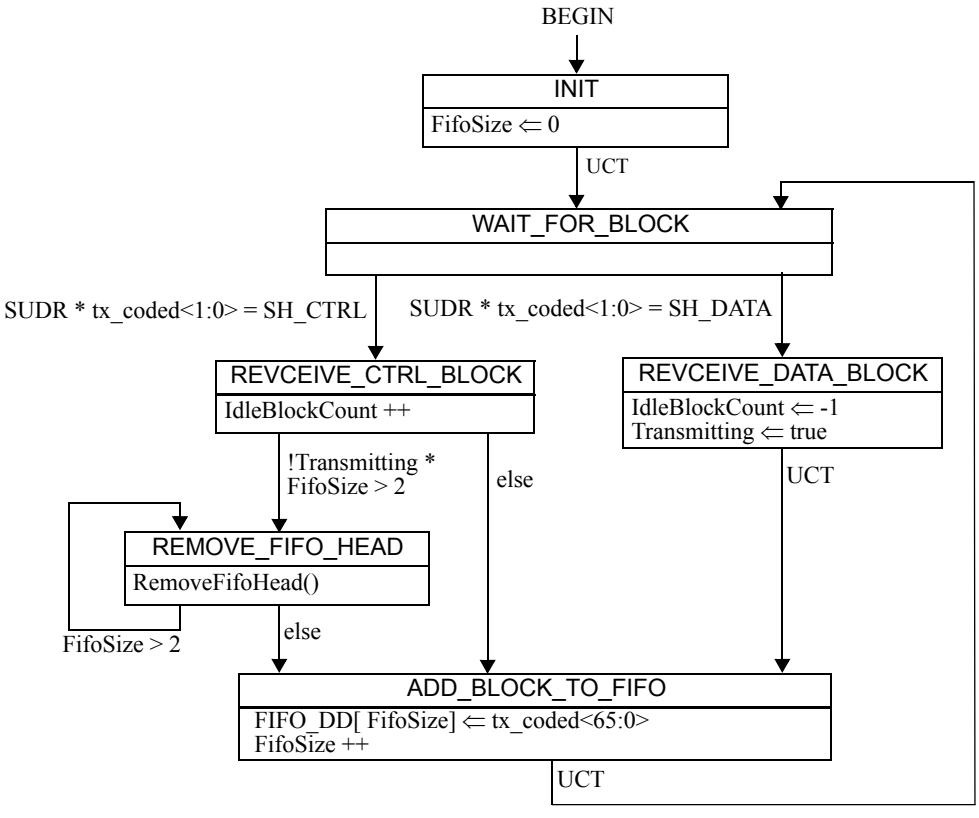


Figure 92–1—Data Detector, input process state diagram

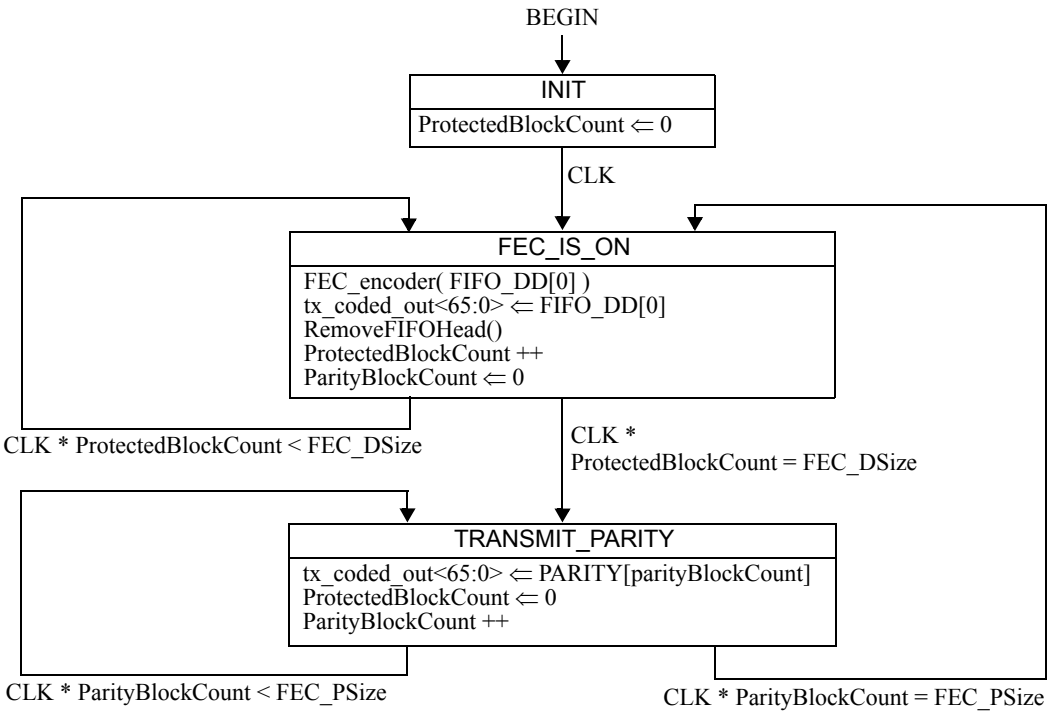


Figure 92–2—OLT Data Decoder, output process state diagram

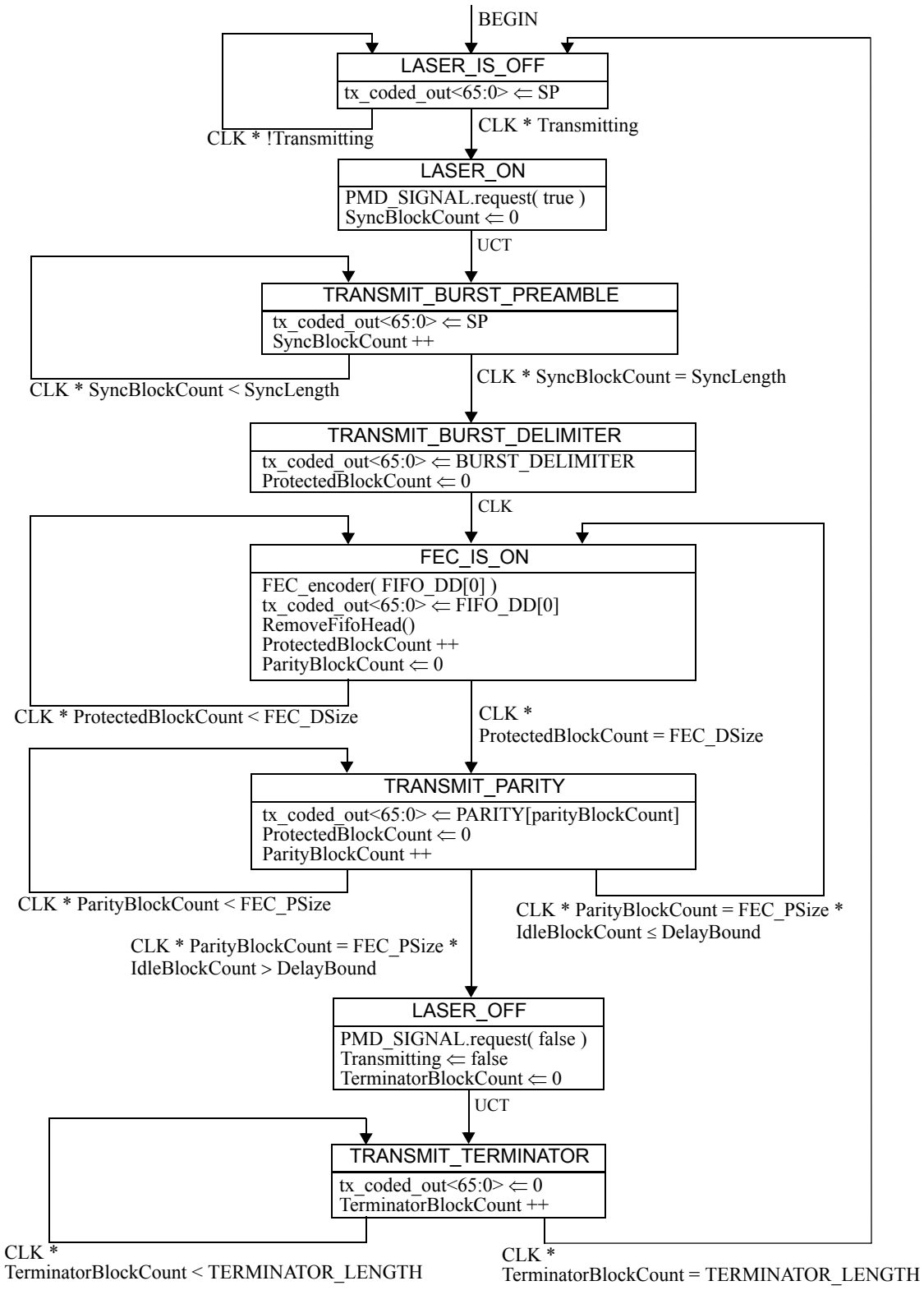


Figure 92–3—ONU Data Decoder, output process state diagram

Description of the Input Process

- Input process waits for a 66-bit block to arrive from the Scrambler.
- It counts the continuous run of Idle blocks (represented by IdleBlockCount).
  - If this value exceeds DELAY_BOUND, this means that the entire FIFO_DD contains only idles.
  - If arriving block is data block, the IdleBlockCount is reset to -1. This is because when a T-block received, IdleBlockCount will increment by one and show a true number of idles as 0.

- If the input block is a data block, Transmitting value is set to true to signal the output process to start turning the laser on (if it is currently off).

Input Process (continued)

- The input process only keeps in the queue the data that will go into the FEC protected section of the burst.
- Between bursts, the Input process only keeps 3 blocks in the queue.
  - When a control block arrives, the head block is purged and the new block is appended.
- When the first data block arrives, the three blocks in the queue will be
  - FIFO_DD[0] = 1st protected IDLE
  - FIFO_DD[1] = 2nd protected IDLE
  - FIFO_DD[2] = block containing /S/
  - FIFO_DD[3] = just arrived data block

Input Process in the OLT

- In the OLT, the Transmitting variable should be initialized to true.
- The same state diagram can be used in the OLT (the REMOVE_FIFO_HEAD state will never be entered)

Output Process (in ONU)

- Output process maintains the same states it had before, but state transitions now occur on edges of TX_CLK.
- The output process reads data from the FIFO_DD only in the FEC_IS_ON state.
- All other states generate output blocks on their own and at right times.

Output Process (in OLT)

- Output process in the OLT only has states FEC_IS_ON and TRANSMIT_PARITY. These two states are identical to the corresponding ONU states.

Conclusion

- The state machines where modeled and verified to work properly.
  - The Data Detector state machine in the OLT introduces delay variability of 1.6 TQ
  - In the ONU, the variability is 3.2 TQ.