10GEPON ONU Upstream Stack

Additional Considerations and Bridge Proposal

Jeff Mandin

802.3av - Geneva

Agenda

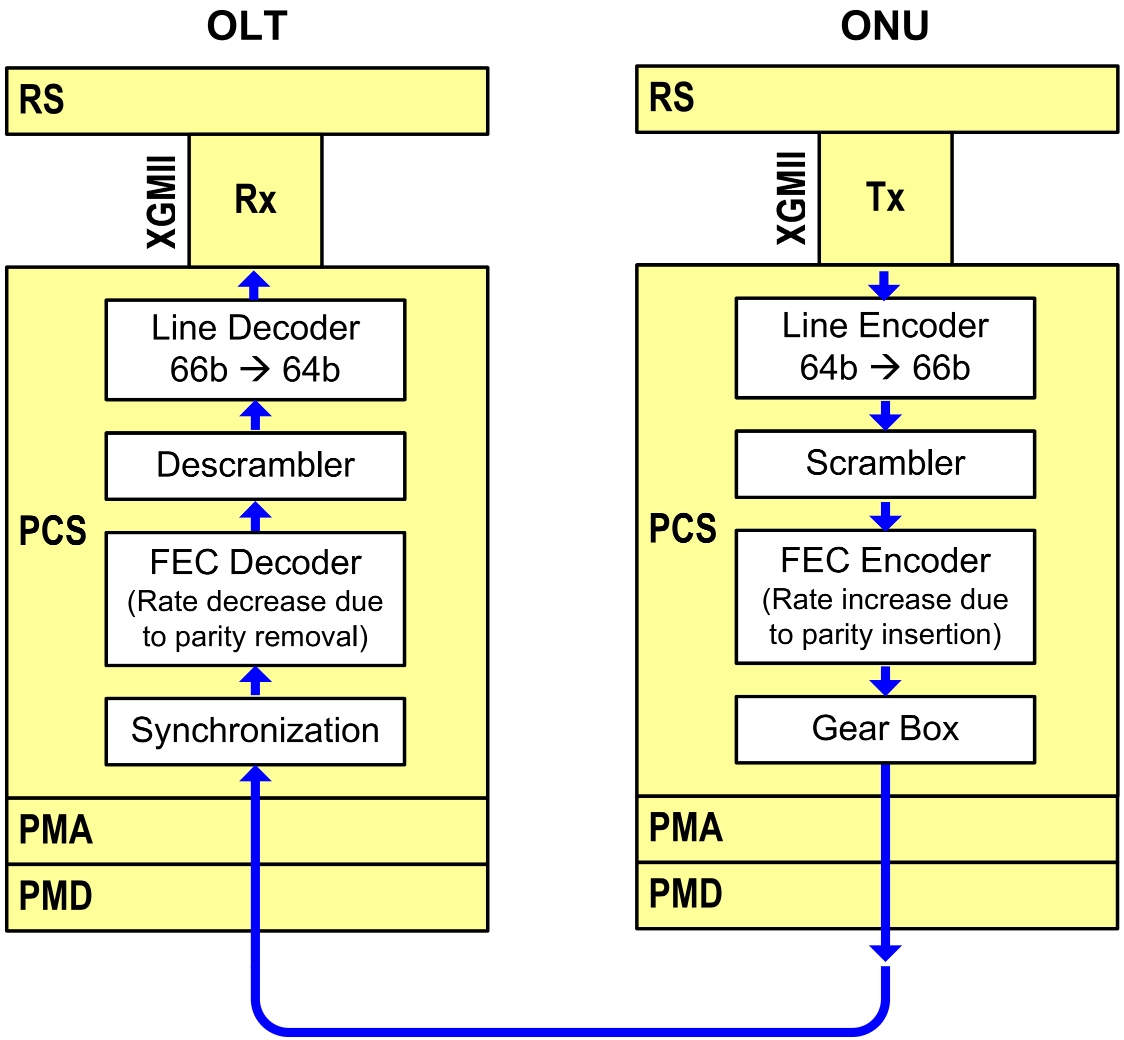
1. New considerations for ONU upstream stack
   - Synchronous Laser Control
   - Use of Ordered Sets for link control

2. Bridge Proposal for ONU upstream stack
   - Features
   - State Diagrams

10GEPON Stack (Baseline Slide)

Note: Data detector placement is an open issue and so the data detector (which activates laser on/off) is not shown.

Additional functional blocks may be added as state diagrams are refined.



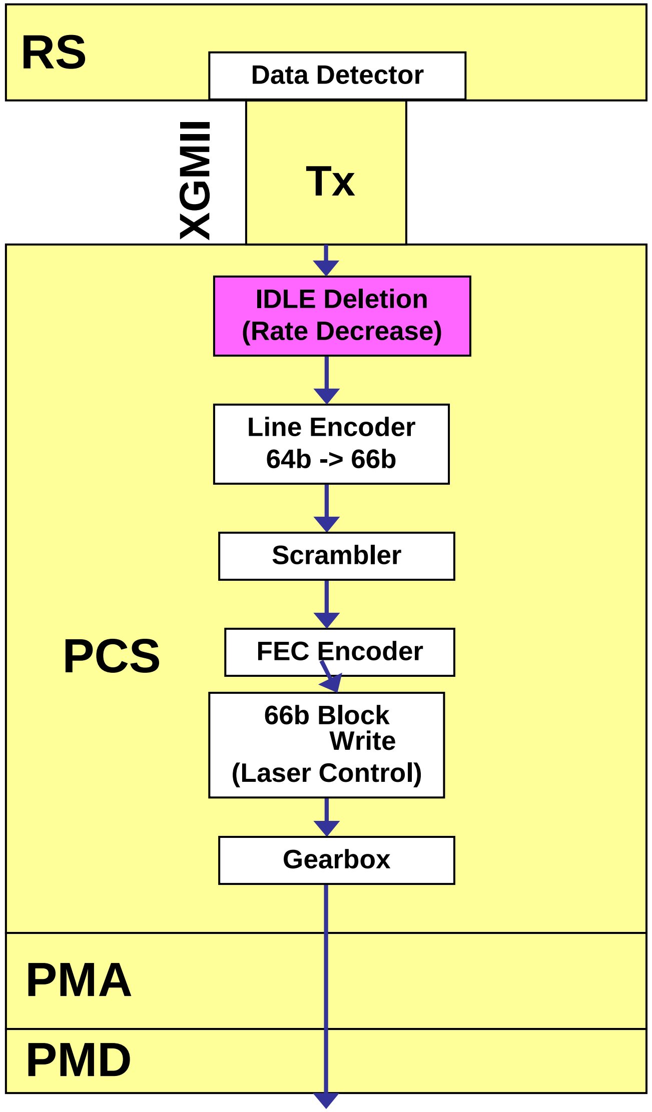
IDLE Deletion

- Deletes IDLEs from the data burst so that (using the example of RS [239,255]) the average rate equals 28/30 *10Gbps

- Deletes IDLEs in groups of 4 and ensures that observance of all IPG rules is maintained

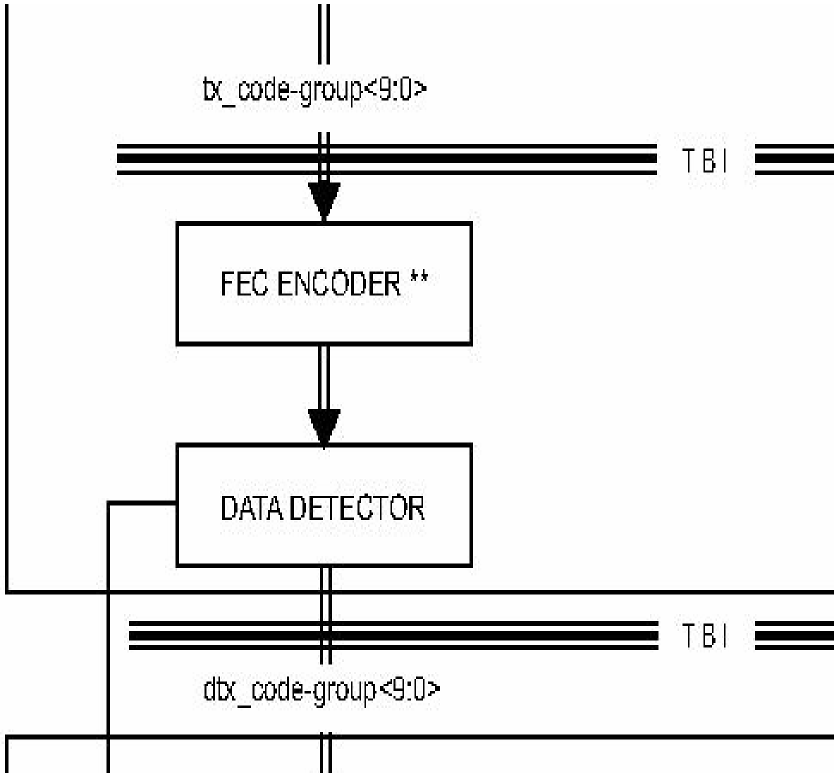
- 66b encoder still operates on tx_raw as described in clause 49.2.13

- Aligns first data frame of the burst to the beginning of the 66b block



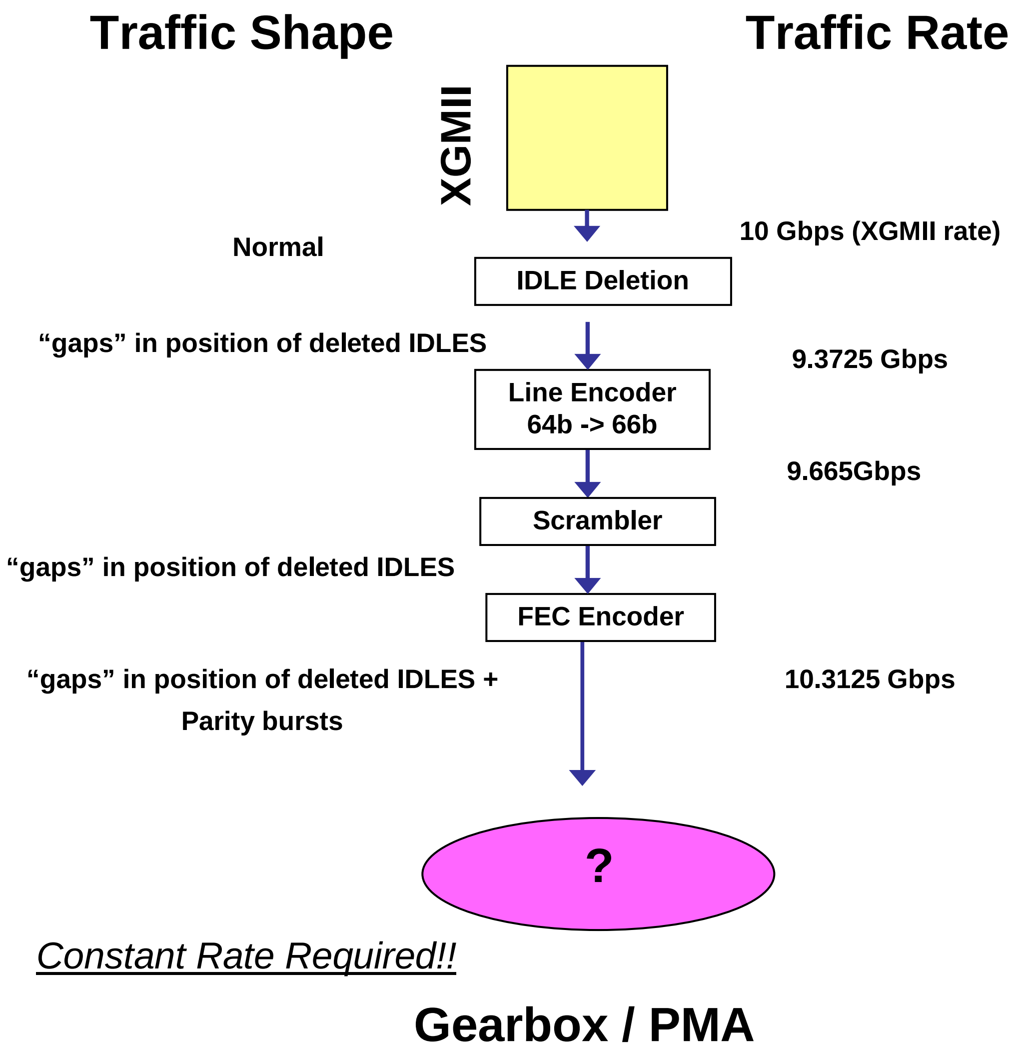
Transmission to PMA is synchronous

- Generally, the 802.3 specification does not describe timing of data passing between functional elements (not needed and much simpler!)
- Functional entities are *logical*, so timing is really undefined
- But for EPON (1G and 10G) we need to address this - as laser control and encoder state is timed by data arrival
- In GEPON, transmission loop is according to arrival on TBI



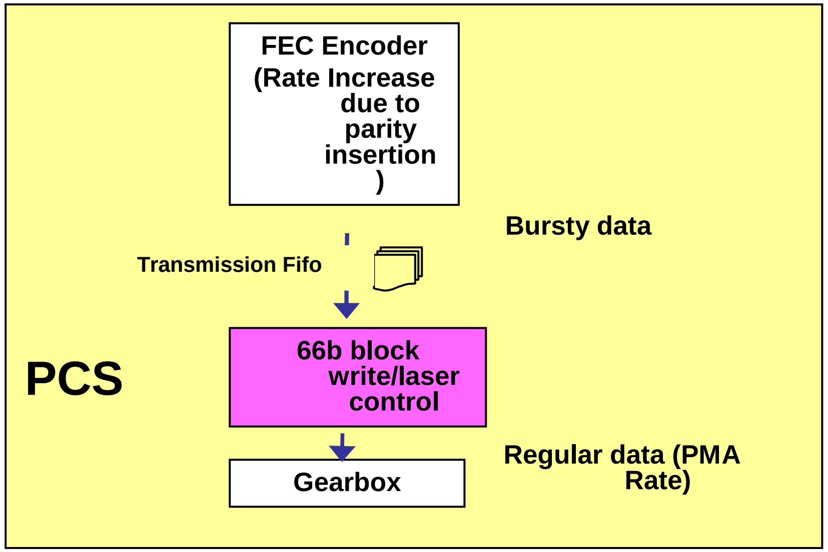
Rate and Shape of Tx Traffic

- Output of the FEC encoder is bursty
- Gearbox/PMA requires constant traffic rate
- What to do?



Block Write functional element

- Output of FEC encoder is “bursty” (ie. “gaps” in position of deleted IDLES + inserted parity)
- Non-regular output of FEC encoder is enqueued to <u>*transmission FIFO*</u>
- *Block Write* functional entity synchronously dequeues the buffered 66b blocks and writes to the gearbox
- Since *Block Write* is synchronous, only this entity can turn the laser on/off



Ordered Sets in 10G Ethernet

- Link Maintenance is one of the functions performed by Ordered Sets in 10G Ethernet (also in 10G Fibre Channel).
  - Eg. 10GBASE-X periodically replaces IDLE with control codes used by local functional elements

Table 48–4—Defined ordered_sets and special code-groups

Code	Ordered_Set	Number of code-groups	Encoding
||I||	**Idle**		**Substitute for XGMII Idle**
||K||	Sync column	4	/K28.5/K28.5/K28.5/K28.5/
||R||	Skip column	4	/K28.0/K28.0/K28.0/K28.0/
||A||	Align column	4	/K28.3/K28.3/K28.3/K28.3/

- **10GEPON Burst initialization is similar to these functions, except that it’s FEC and the laser state that are being controlled**
  - **Ordered Sets are thus a *simple*, convenient and precedented mechanism to use for communication between stack layers**

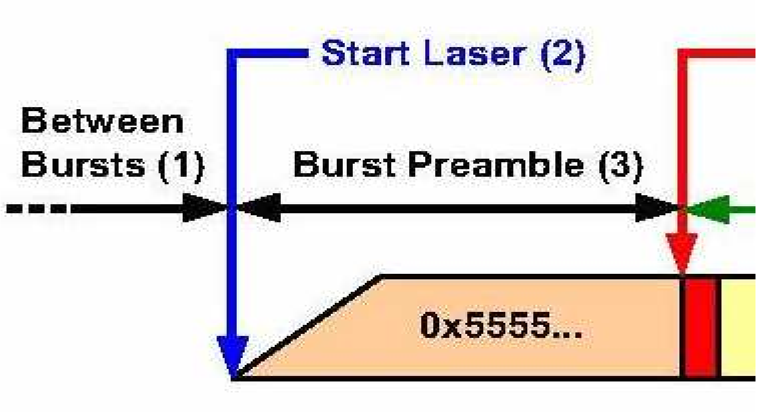
Using Ordered Sets for Upstream Burst Control

The familiar Data Detector functionality would now need to be located in the RS.

We might define Ordered Sets for upstream burst control as follows:

Burst Preamble		Encoded to the 0x555.. Preamble pattern
Start of Burst		Indicates that the subsequent codeword contains actual user data (and requires FEC protection).
End of Burst		Immediately follows the last codeword of actual data. Triggers laser off.

Length of the Transmitted Burst Preamble

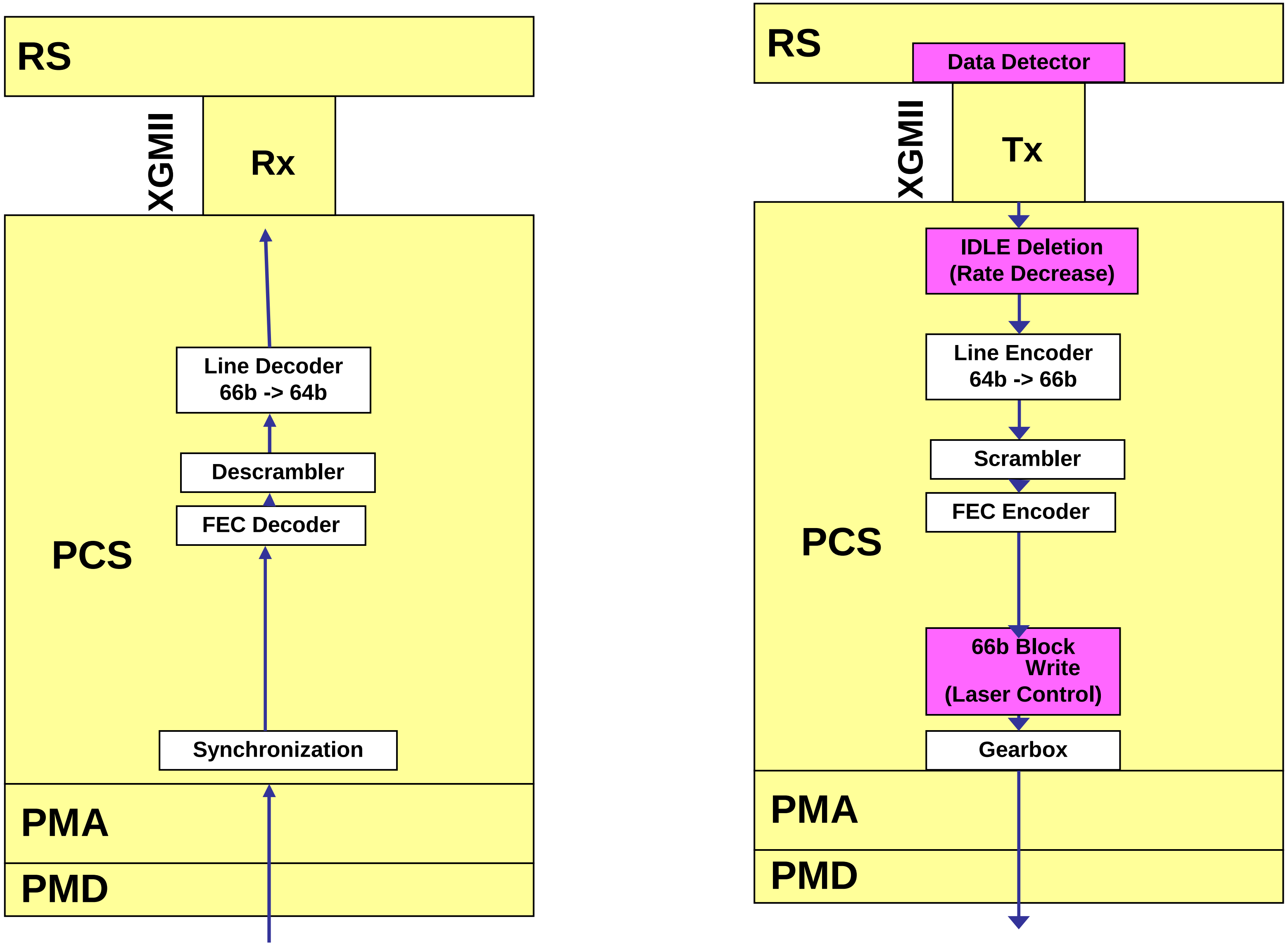


- Inserting extra bytes (or “false parity”) to the datastream to maintain PMA rate when FEC is off means that the number of codewords transmitted between laser-on and the delimiter will not be constant.

- Because of the variability in when the extra bytes are inserted, the length of the transmitted burst preamble will vary by 132 bits in the case of RS(239, 255)

- Whereas if we use an *Ordered Set* to indicate the burst preamble, we have no complex deletion and restoration schemes - so we avoid the problem altogether

Revised 10GEPON stack



Summary

1. The ONU stack model must include a new element which runs synchronously to perform laser control and write to the PMA
2. Ordered Sets enable communication between functional blocks and are a typical mechanism used in 10G ethernet for link maintenance.

   As such we should consider:

   - Signalling the stages of burst initialization using ordered sets
   - Locating the data detector functionality above the XGMII ie. in the RS

Bridge Proposal

1. **The following slides define a scheme for the ONU upstream stack The scheme incorporates the most recent discussions and TF input and comprises elements of the previous proposals:**
   - Stack awareness of burst phase: 3av_0703_mandin_1.pdf
   - Unified architecture + modified data detector state diagram:
     3av_0703_kramer_1.pdf
   - Burst signalling: Based on methods used link maintenance signaling in 10GBASE-X and related standards
2. **Goals:**
   1. Simplicity => blocks do what you would expect
   2. Completeness
   3. Transparency => avoid corner cases, hidden assumptions
3. **3 State diagrams:**
   1. Data Detector
   2. FEC
   3. Block Write(Laser Control)

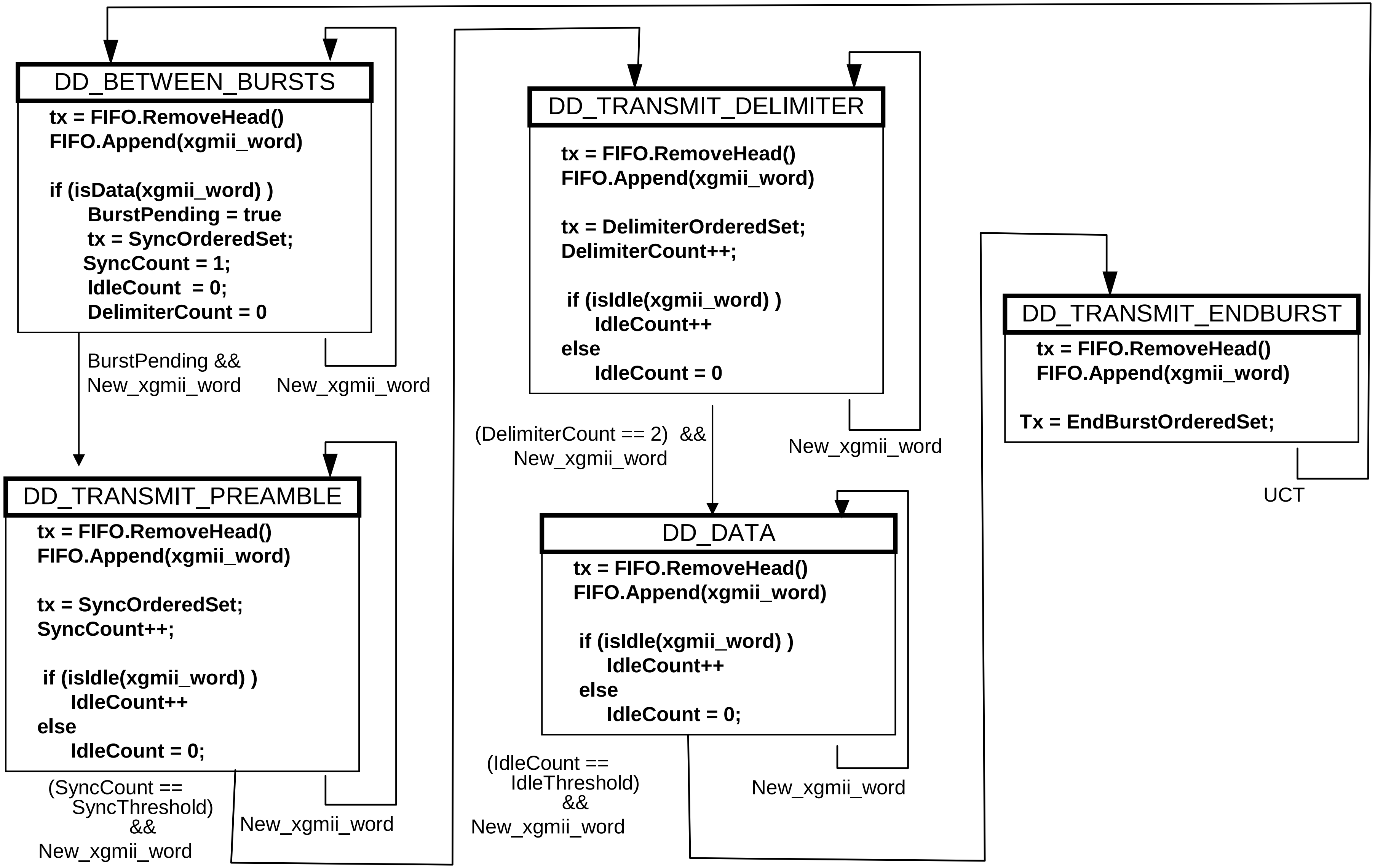
State Diagram Behaviour in IEEE

21.5.1 Actions inside state blocks

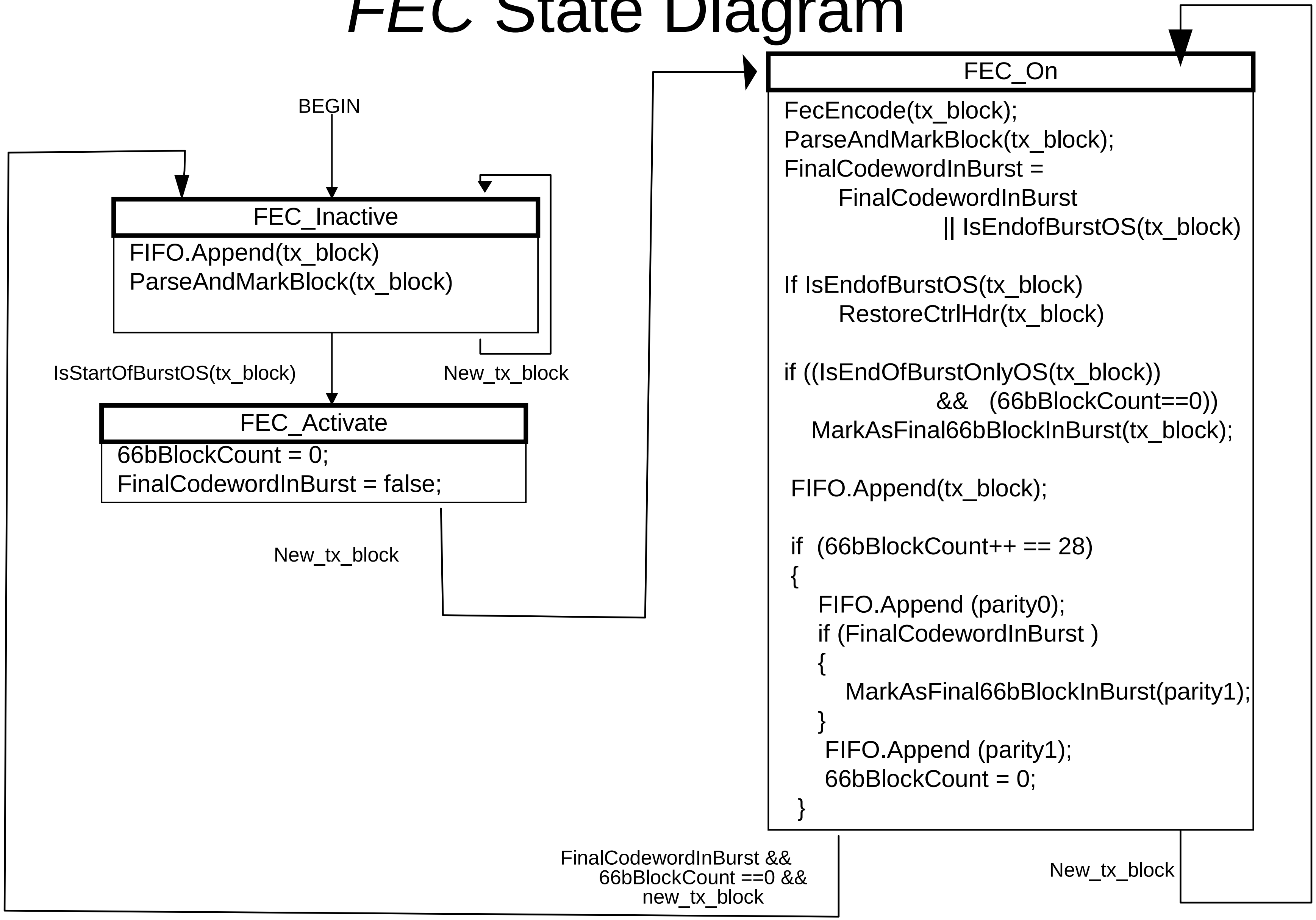
The actions inside a state block execute instantaneously. Actions inside state blocks are atomic (i.e., uninterruptible).

After performing all the actions listed in a state block one time, the state block then continuously evaluates its exit conditions until one is satisfied, at which point control passes through a transition arrow to the next block. While the state awaits fulfillment of one of its exit conditions, the actions inside do not implicitly repeat.

Data Detector State Diagram



FEC State Diagram



FEC State Diagram Description (1)

```
ParseAndMarkBlock(tx_block)
{
     Descramble(tx_block);
     sIsBurstPreambleOrderedSet = (tx_block == TX_BURST_PREAMBLE_ORDERED_SET);
     sIsStartOfBurstOrderedSet = (tx_block == TX_START_OF_BURST_ORDERED_SET);
     sIsEndOfBurstOnlyOrderedSet = (tx_block == TX_END_OF_BURST_ORDERED_SET_LOW);
     sIsEndOfBurstOrderedSet = ((tx_block == TX_END_OF_BURST_ORDERED_SET_LOW) ||
                         ( tx_block == TX_END_OF_BURST_ORDERED_SET_HI) );
    if (sIsEndOfBurstOrderedSet)
               tx_block.hdr = 00;
   if (sIsPreamble || sIsStartOfBurst  || sIsEndOfBurstOnly)
               tx_block.hdr = 11;
   else
               Rescramble(tx_block);
}
```

ParseAndMarkBlock() gives the 2bit header of the 66b block a temporary value:

- BurstPreamble and StartOfBurst, and EndOfBurst (occuring by itself) Ordered Sets marked with 11

- EndOfBurst ordered set (following data) marked with 00

BurstPreamble, StartOfBurst, EndOfBurst (occuring by itself) do not need to be re-scrambled.

FEC State Diagram Description (2)

Functions

- IsBurstPreambleOS(tx_block)

  Returns true iff. (tx_block.hdr == 11) and the payload contains 2 instances of BURST_PREAMBLE_ORDERED_SET

- IsStartOfBurstOS(tx_block)

  Returns true iff. (tx_block.hdr == 11) and the payload contains 2 instances of START_OF_BURST_ORDERED_SET

- IsEndOfBurstOnlyOS(tx_block)

  Returns true iff. (tx_block.hdr == 11) and the payload contains TX_END_OF_BURST_ORDERED_SET_LOW

- IsEndOfBurstOS(tx_block)

  Returns true iff. the supplied block are marked with 00

- RestoreCtrlHdr(tx_block)

  Resets tx_block.hdr to original value of ‘10’

- MarkAsFinal66bBlockInBurst(tx_block)

  Sets tx_block.hdr to ’00’

FEC State Diagram Description (3)

States

FEC_Inactive

- In this state the received 66b block is simply queued to the *transmission FIFO*
- After a block containing 2 StartOfBurst Ordered Sets is processed, the state transitions to *FEC_Activate*

FEC_Activate

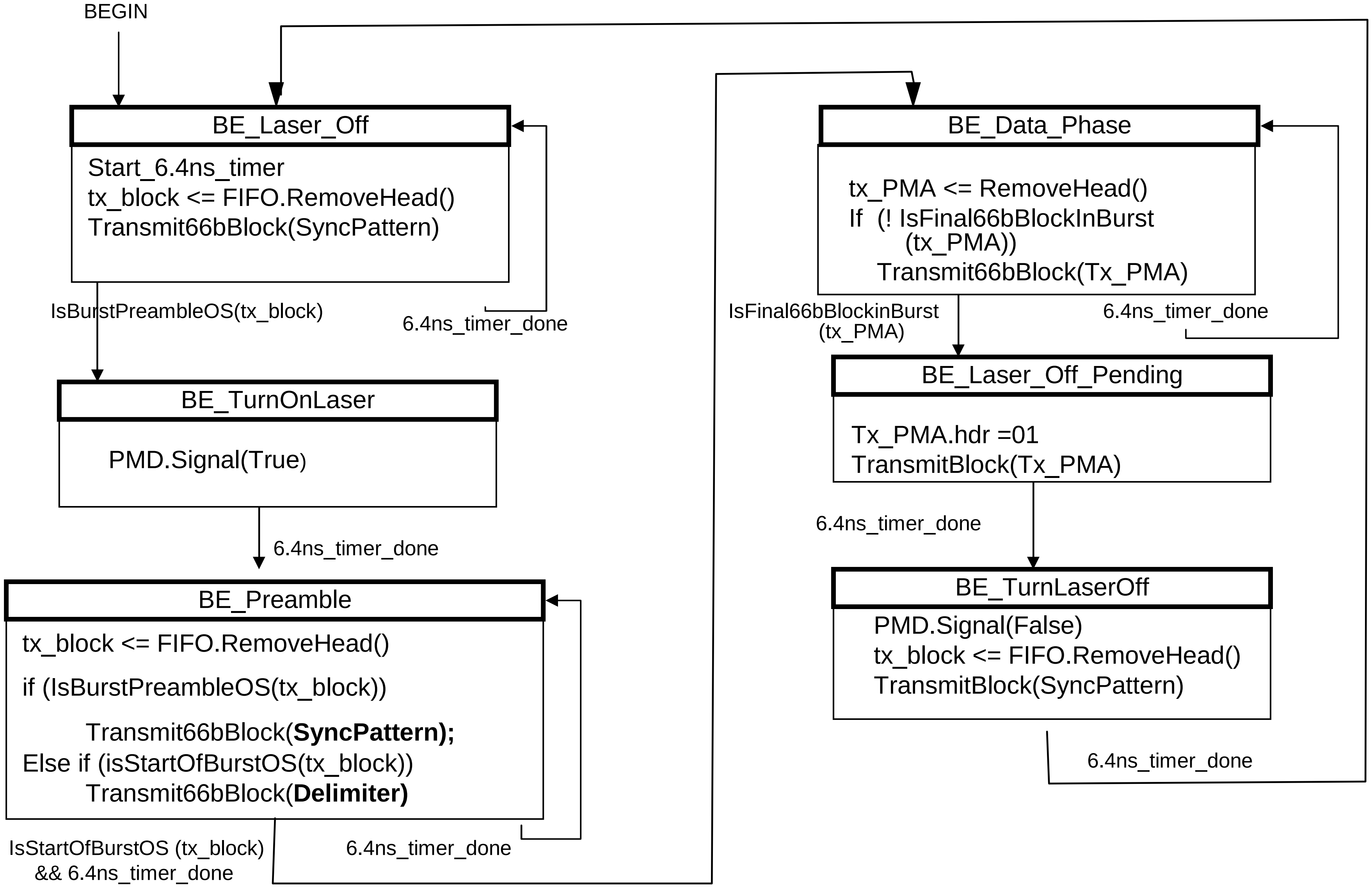
- A transitional state which initializes the FEC variables and waits for the next 66b block

FEC State Diagram Description (4)

FEC_On

- Each 66b block is supplied to the FEC encoder before being enqueued to the transmission FIFO
- If the 66b block included an EndOfBurst ordered set, we note that this FEC Codeword must be the last one in the burst by setting the flag *FinalCodewordInBurst*
- After 28 (for this example) 66b blocks have been enqueued, the 2 parity blocks are also appended to the transmission FIFO.
  - If at the time of writing parity the *FinalCodewordInBurst* flag is set, then the 2bit header of the second parity word is set to ’00’ which is a signal to the *BlockWrite* module that the burst has completed
  - In this case, state transitions to FEC_Inactive after the final parity words are enqueued to the transmission FIFO

State Diagram for *Block Write/Laser Control*



BlockWrite/LaserControl State Diagram Description (1)

Constants

- **SyncPattern**: the 66bit 1010… pattern
- **Delimiter**: the 66bit Burst Delimiter pattern

Variables

- Tx_PMA: contains the 66b value to be written to the gearbox/PMA

Functions

- Transmit66bBlock(tx_block) - passes tx_ block on for forwarding by the gearbox/PMA.
- IsFinal66bBlockInBurst(tx_block) – returns true iff. (txblock.hdr == 00).

BlockWrite/LaserControl State Diagram Description (2)

BE_Laser_Off

- In this state the Sync (0x5555..) Pattern is transmitted every 6.4 ns and a single 66b block is removed from the transmission FIFO
- Receipt of the BurstPreamble Ordered Set signals the beginning of the burst and triggers the transition to the next state

BE_Turn_On_Laser

- The laser is turned on and the next state is entered immediately

BE_Preamble

- In this state, a 66b block is dequeued (expected values are BurstPreamble and StartOfBurst Ordered Sets) and is replaced with the 0x555… or 66 bit delimiter pattern as appropriate.

State Descriptions for *BlockWrite* State Diagram (3)

BE_Data_Phase

- In this state a single dequeued block is transmitted.
- However if the 2bit header was marked as 00 by the FEC layer to signal that it is in fact the final block in a burst, the next state is entered instead.

BE_Laser_Off_Pending

- The 2 bit header (marked 00 by FEC) is restored to its original value (ie. 01 as it must be a control block), the block is transmitted to the gearbox, and the next state is entered after completion of a 6.4ns cycle

BE_Turn_Laser_Off

- Turns the laser off and cycles back to LaserOff state.