Control Multiplexer, Idle Deletion, Data Detector, and GearBox for Upstream CNU Burst Transmission. Control Multiplexer

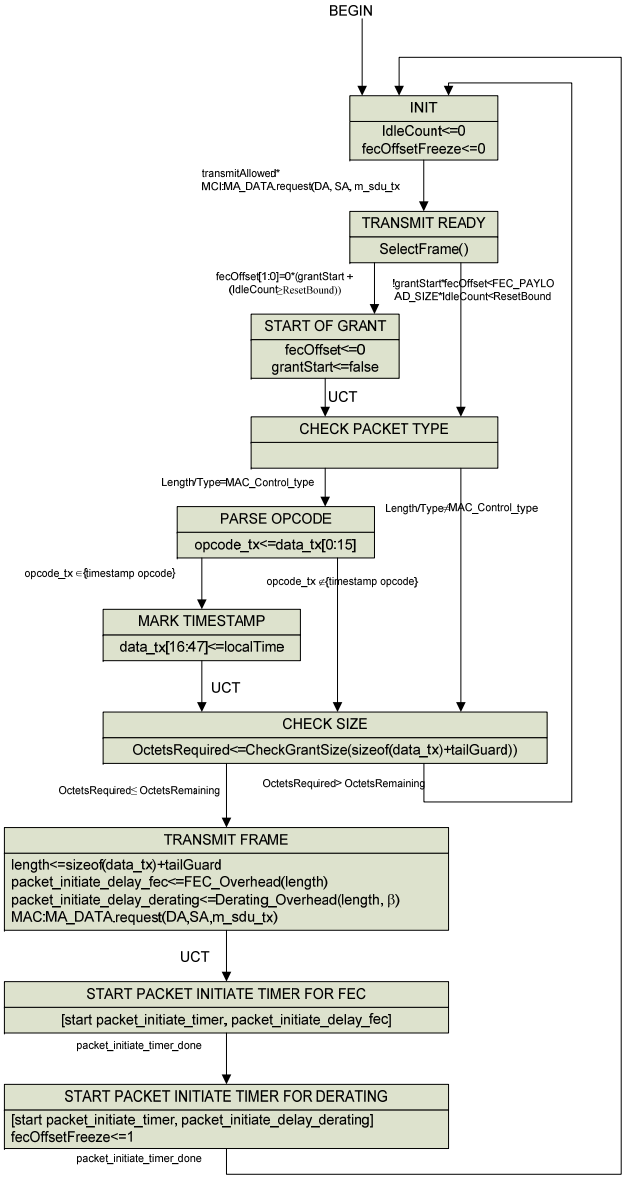
Jin Zhang, Xiaolin Che

Marvell

Functions of CNU Control Multiplexer

- Common for CLT and CNU control multiplexer
  - Select which data unit to send
    - For example, always control unit before data unit.
    - Implementation dependent
  - Force idle period after each data unit for:
    - Insertion of FEC parity
    - Insertion of blank period to adapt to PMD rate.
- CNU only:
  - Identify gap of idles and decide formation of bursts
  - Check if the remaining time is enough for the incoming MAC frame

CNU Control Multiplexer



Key parameters and functions:

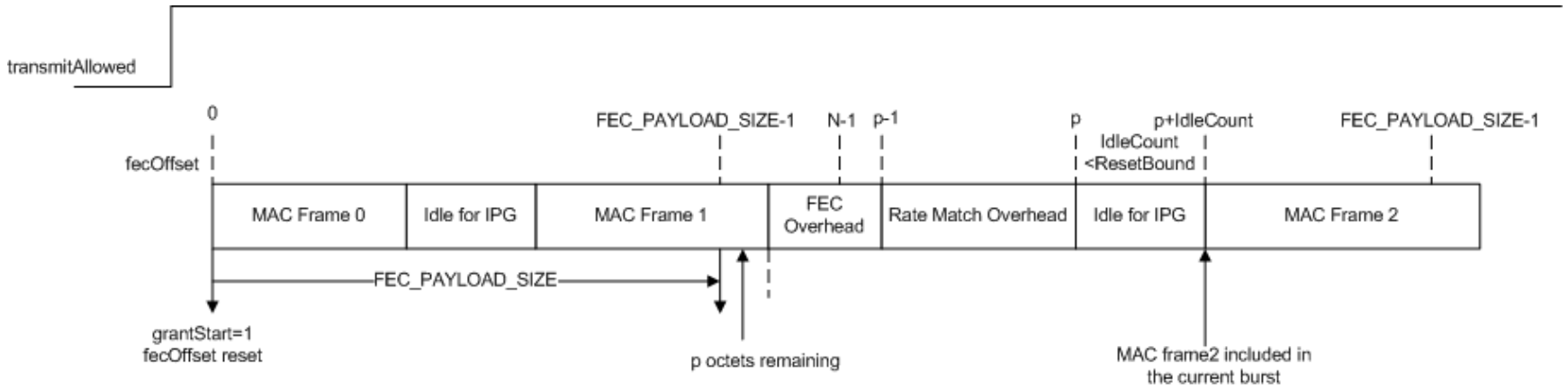
1. ResetBound
2. CheckGrantSize
3. FEC_Overhead and Derating Overhead

ResetBound

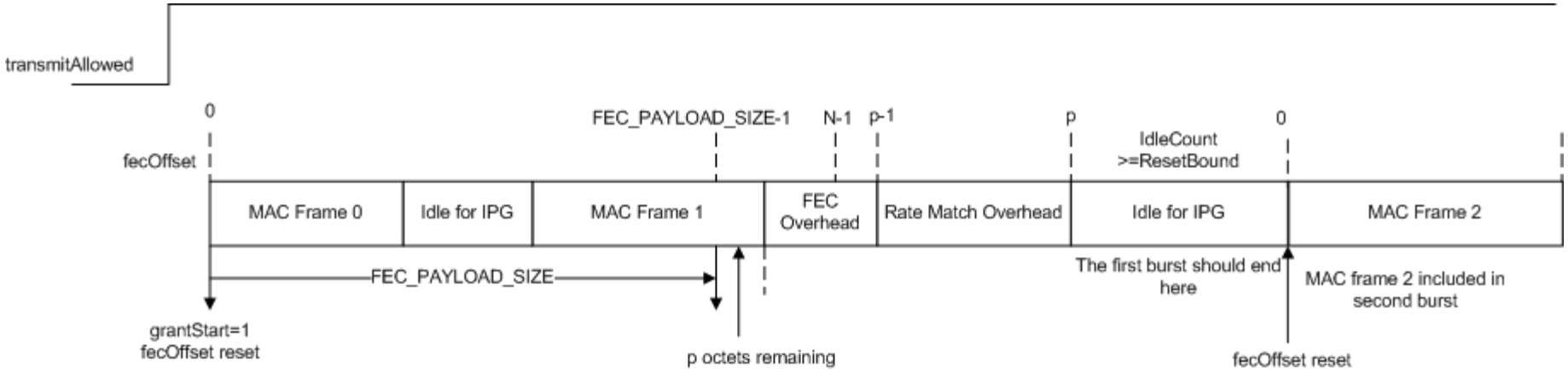
- “IdleCount” counts the natural idles between MAC frames
- Case 1: If IdleCount<ResetBound, it is more efficient to include the next frame in the current burst.
- Case 2:If IdleCount >= ResetBound, terminate the current burst, because this is more efficient.

Function of ResetBound

Case 1:



Case 2:



How to Determine ResetBound?

- Part 1: Cover the time used to transmit the remaining p octets from previous MAC frame

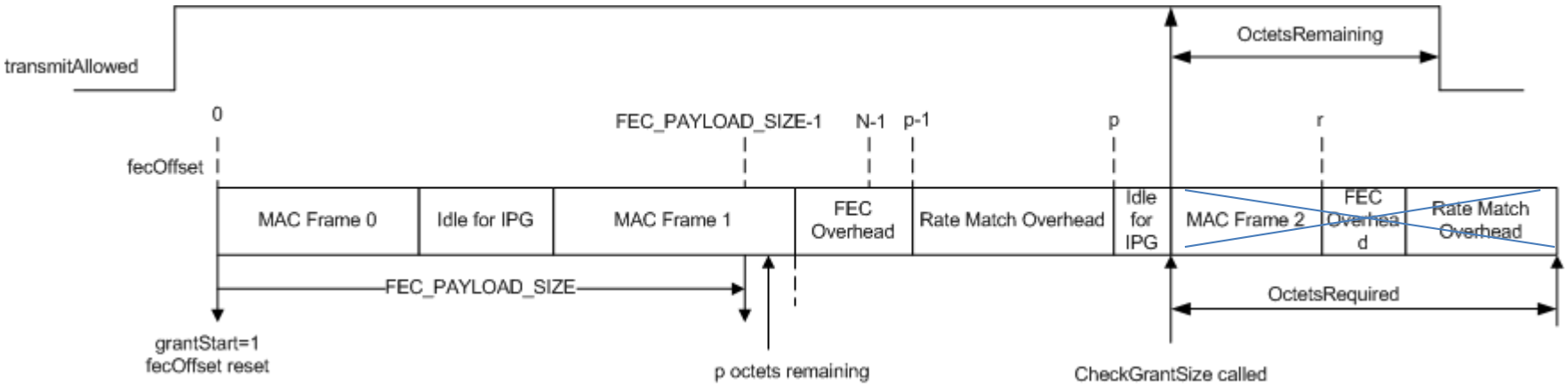
  p1 = f(p,β);//similar to CheckGrantSize function

  Should first decide how to transmit the remaining p octets, where 0<=p<FEC_PAYLOAD_SIZE

- Part 2: an upper bound of the time used to transmit end marker and start marker
  - Worst case: p2=10 bits * (Number of REs for start and end markers)* β/8
- ResetBound >= p1+p2,
  - For efficiency, we can set ResetBound as function of p
  - For simplicity, we can set ResetBound for the worst case of p.

CheckGrantSize Function

- Purpose
  - whenever a new MAC frame is about to transmit, check if the current grant has enough time to allow for it



Calculate CheckGrantSize(length,β)

1. [n0, n1, n2] = FEC_LUT(fecOffset+length);
2. Psize = n0*FEC_PARITY_SIZE0+
   n1*FEC_PARITY_SIZE1+
   n2*FEC_PARITY_SIZE2;
3. OctetsRequired=length+Psize*β+(β-1)*(fecOffset+length);
4. Return OctetsRequired;

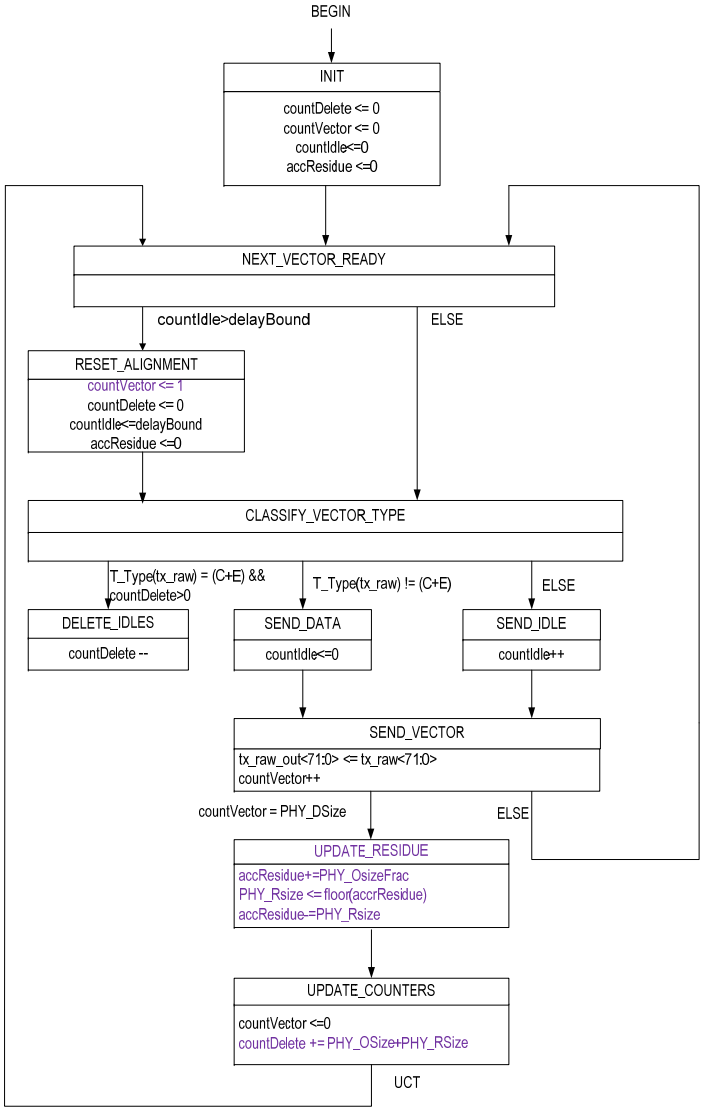
CNU Idle Deletion

- Only delete 72-bit idle vectors inserted from control multiplexer.
  - Delete idle vectors for FEC overhead
  - Delete idle vectors for rate adaptation overhead
- Should not delete idle vectors for inter-packet gap.
- In the long run, should supply exact amount of vectors for PMD rate.
- CNU Only: identify the end of burst, and reset to a transparent mode

Modification of CNU Idle Deletion

- Combine deletion of idles for FEC overhead and rate adaptation overhead into a unified diagram.
- Compensate for the residue of PHY_Osize.
- In the transparent mode, reset the FIFO size.

Modified CNU Idle Deletion



- accResidue: a fractional number to accumulate the residue.
- PHY_RSize: binary number, 1 or 0
- PHY_DSize: integer 220
- FEC_OSize: integer 29
- PHY_OSize:

$$\mathrm{PHY\_OSize} = \left\lfloor \frac{\mathrm{XGMII\_Rate} - \mathrm{PCS\_Rate}}{\mathrm{PCS\_Rate}} \times (\mathrm{PHY\_DSize} + \mathrm{FEC\_OSize}) \right\rfloor + \mathrm{FEC\_OSize}$$

- PHY_OsizeFrac: fractional part of the actual PHY_Osize. Its precision is implementation dependent.

$$\mathrm{PHY\_OSizeFrac} = \frac{\mathrm{XGMII\_Rate} - \mathrm{PCS\_Rate}}{\mathrm{PCS\_Rate}} \times (\mathrm{PHY\_DSize} + \mathrm{FEC\_OSize}) + \mathrm{FEC\_OSize} - \mathrm{PHY\_OSize}$$

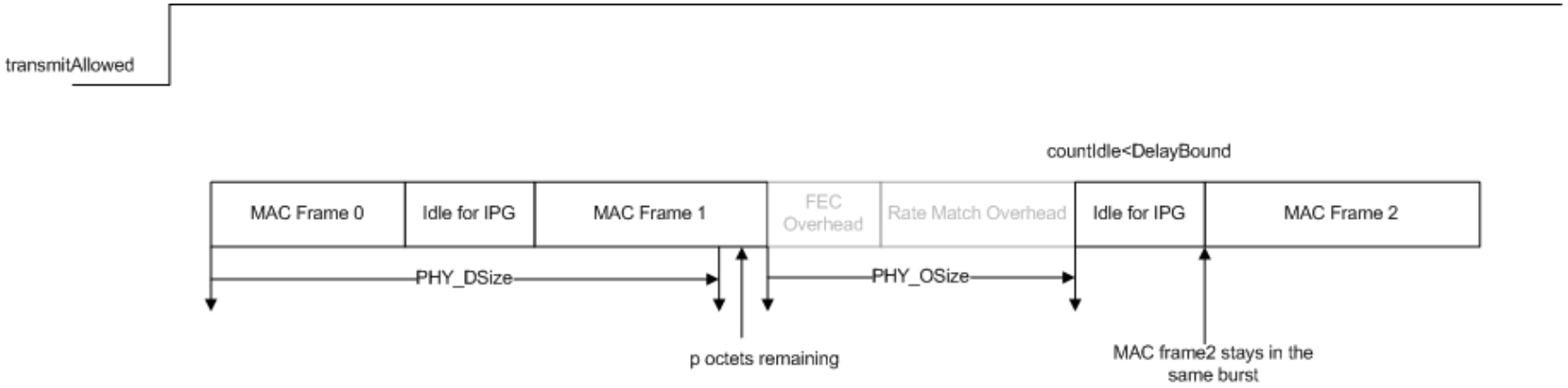
$$\mathrm{PCS\_Rate} = \mathrm{PMD\_Rate} \times \frac{64}{65}$$

$$\mathrm{PMD\_Rate} = \frac{\mathrm{PLCTotalBits}}{\mathrm{PLCTotalCycles}} \times \mathrm{OFDM\_SampleFreq}$$

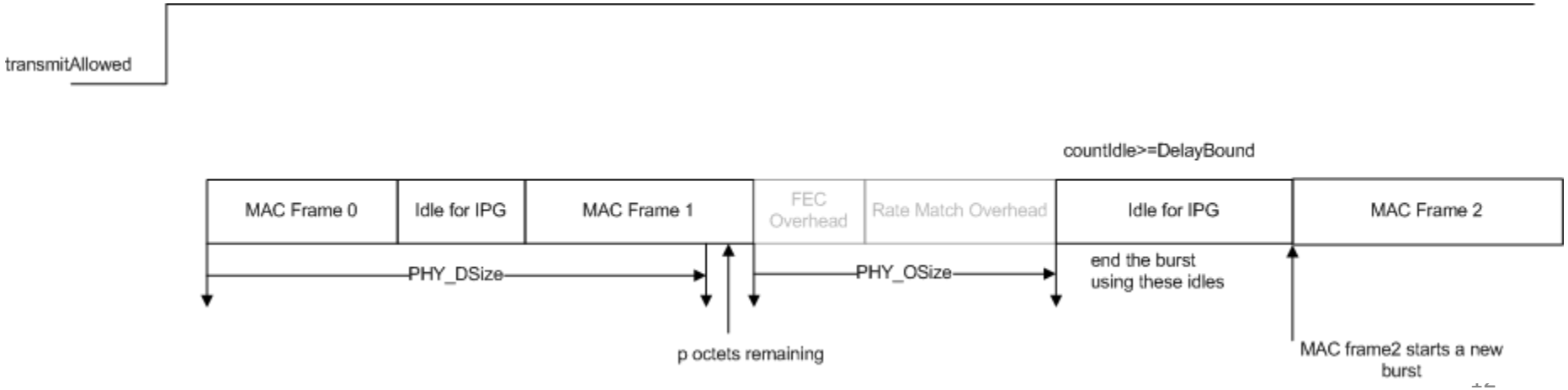
- DelayBound = ResetBound/8

Example

Case 1:



Case 2:



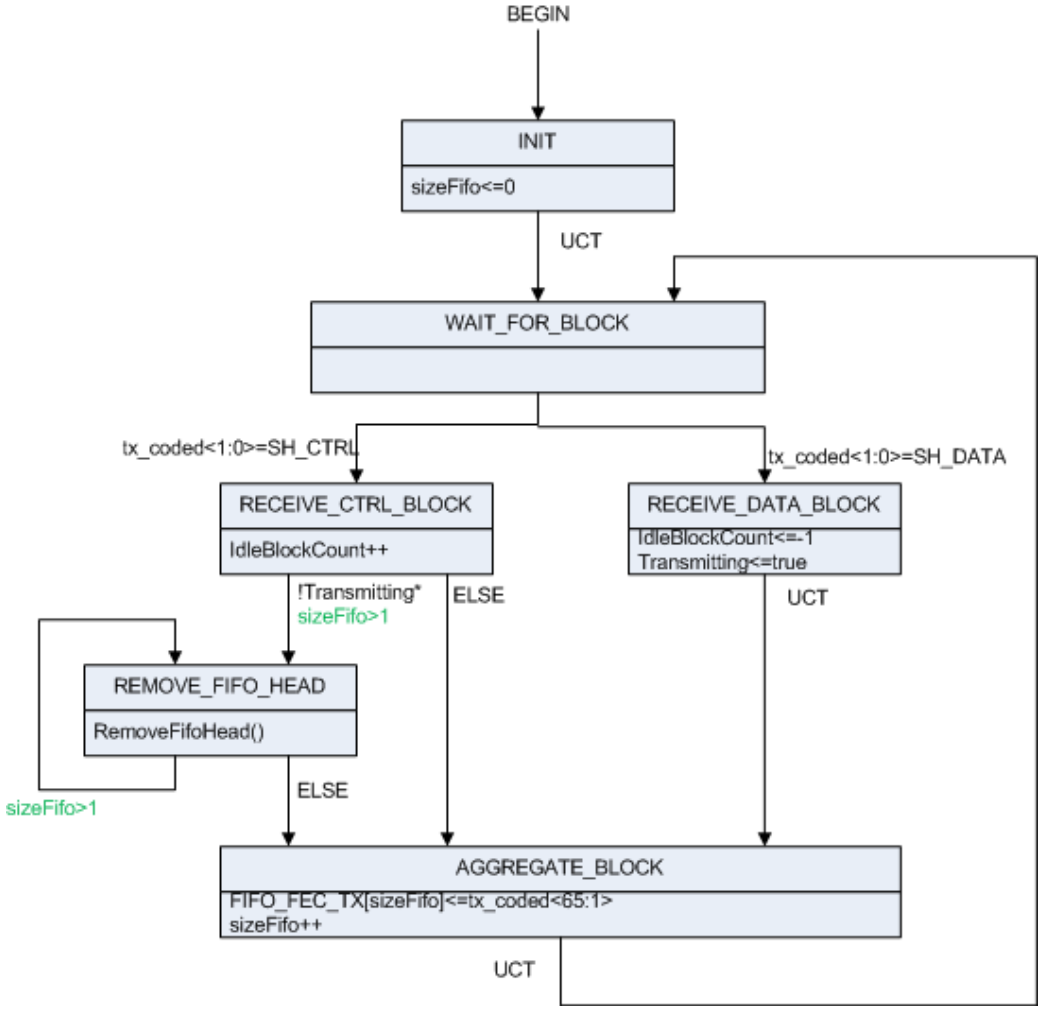
CNU Data Detector

- Control FEC encoder and insert parity.
- Formation of burst
  - Act on arrival of data vector and trigger the burst start.
  - Identify the arrival of idles indicating end of burst.
  - Upon terminating a burst, process the remaining data octets and insert FEC parities and rate adaptation overhead.

CNU Data Detector Input Process

- When no burst data is present, idle deletion is in a “transparent mode”, i.e. passing down all idle vectors.
- CNU data detector shall block all idles in “transparent mode”, except for one idle vector reserved for special use.
- CNU data detector starts to buffer data in FIFO when data vector is detected.

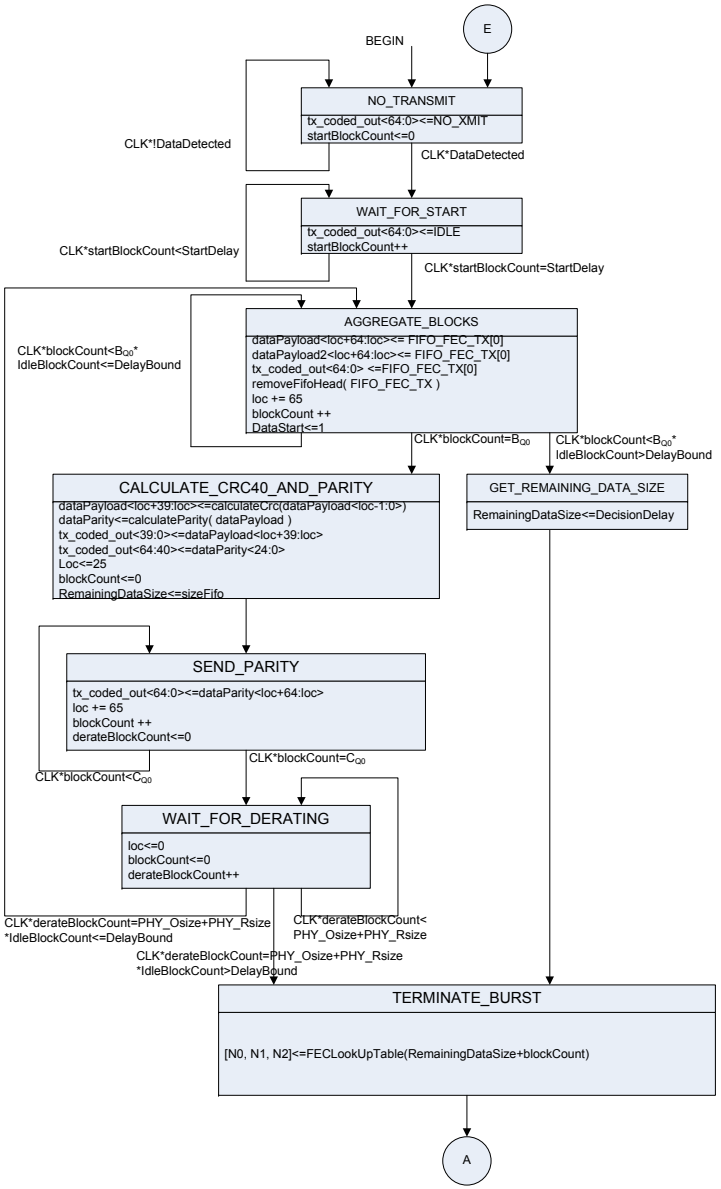
Proposed CNU Data Detector Input Process



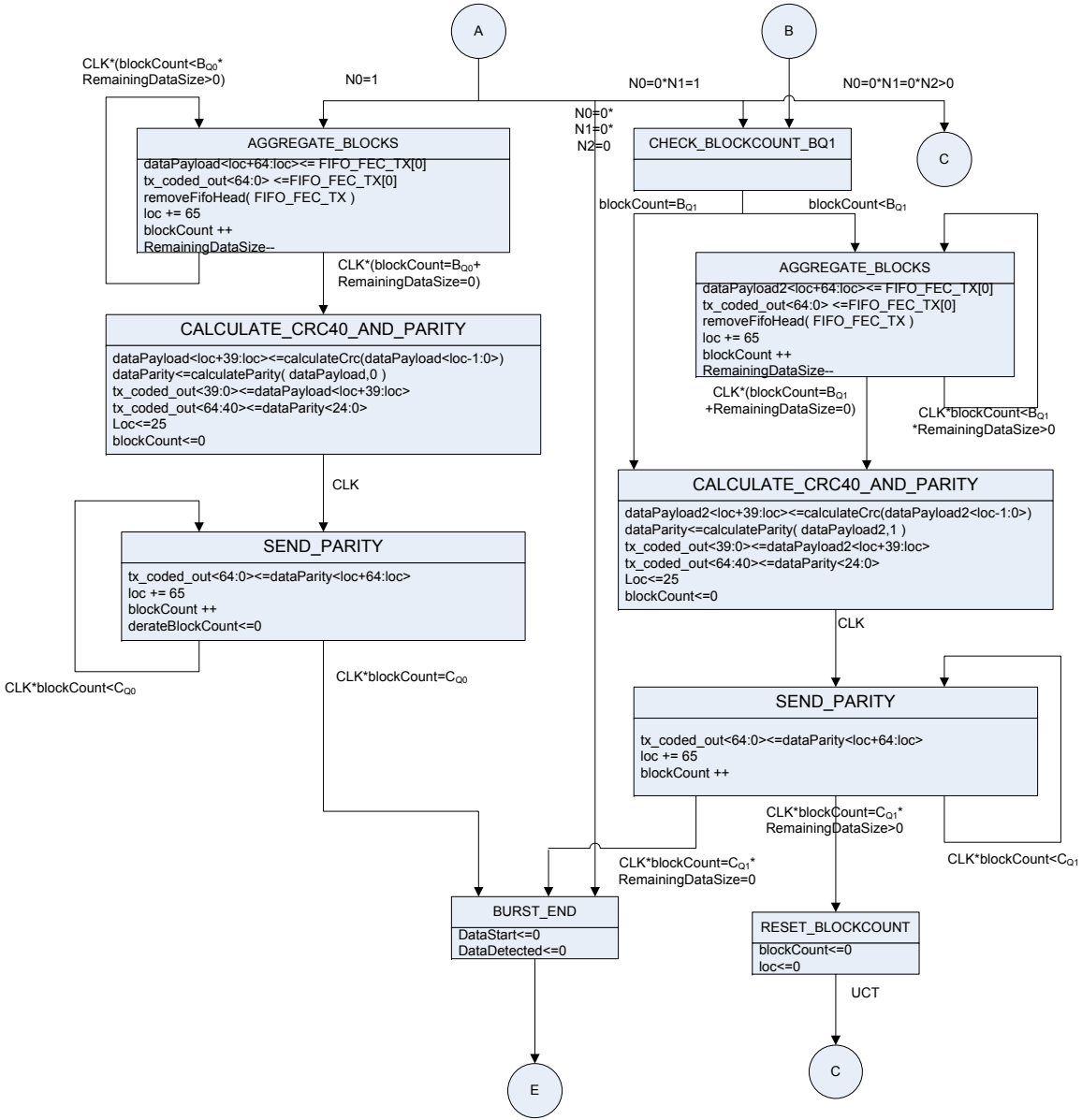
CNU Data Detector Output Process

- Insert FEC parities
  - 3 encoders to consider
  - Two options were on debate.[ref: kliger_3bn_02_0314.pdf]
    - Two options have similar TX memory. (not critical)
    - Option 1 has more TX latency: 1.6~6.4us
    - Option 2 has higher implementation complexity, mainly on the memory scheduling and sharing at the receiver.
    - The decision is basically between certain TX latency and possible receiver complexity.
  - In this contribution we show how data detector based on option 1 works
  - Disclaimer: This contribution does not preclude any future possibility of simplification of coding scheme, for example, using 2 instead of 3 FEC codes. In case that TX latency turns out to be a bottleneck during further investigation, we do not preclude other options.

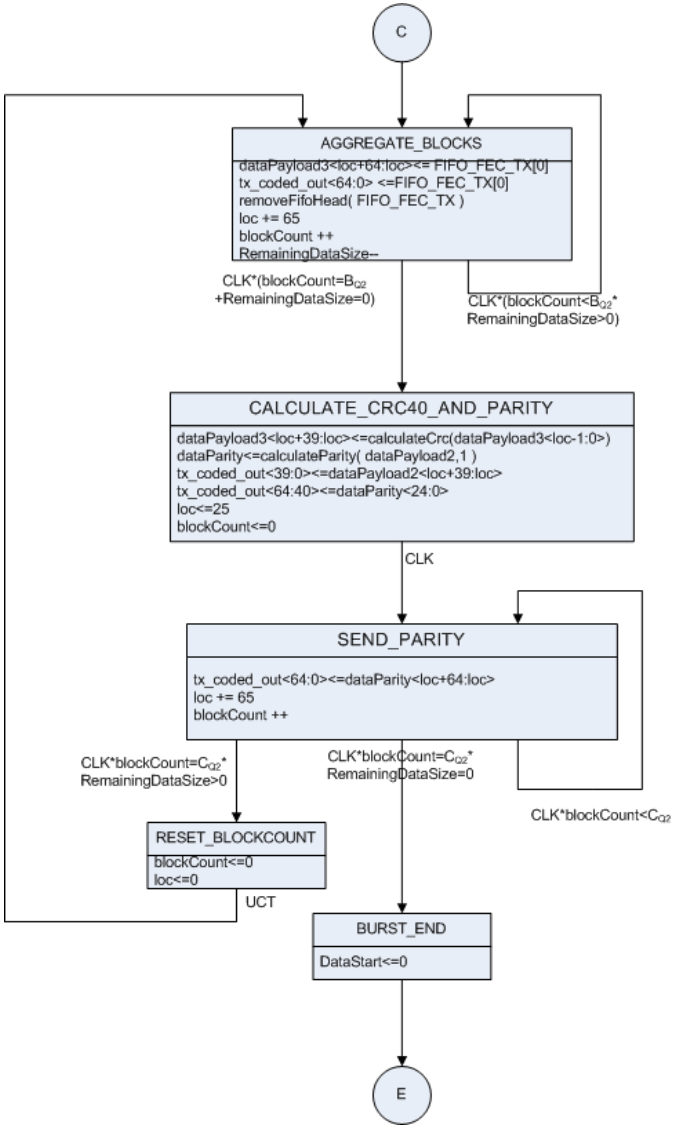
CNU Data Detector Diagram – I



CNU Data Detector Diagram – II



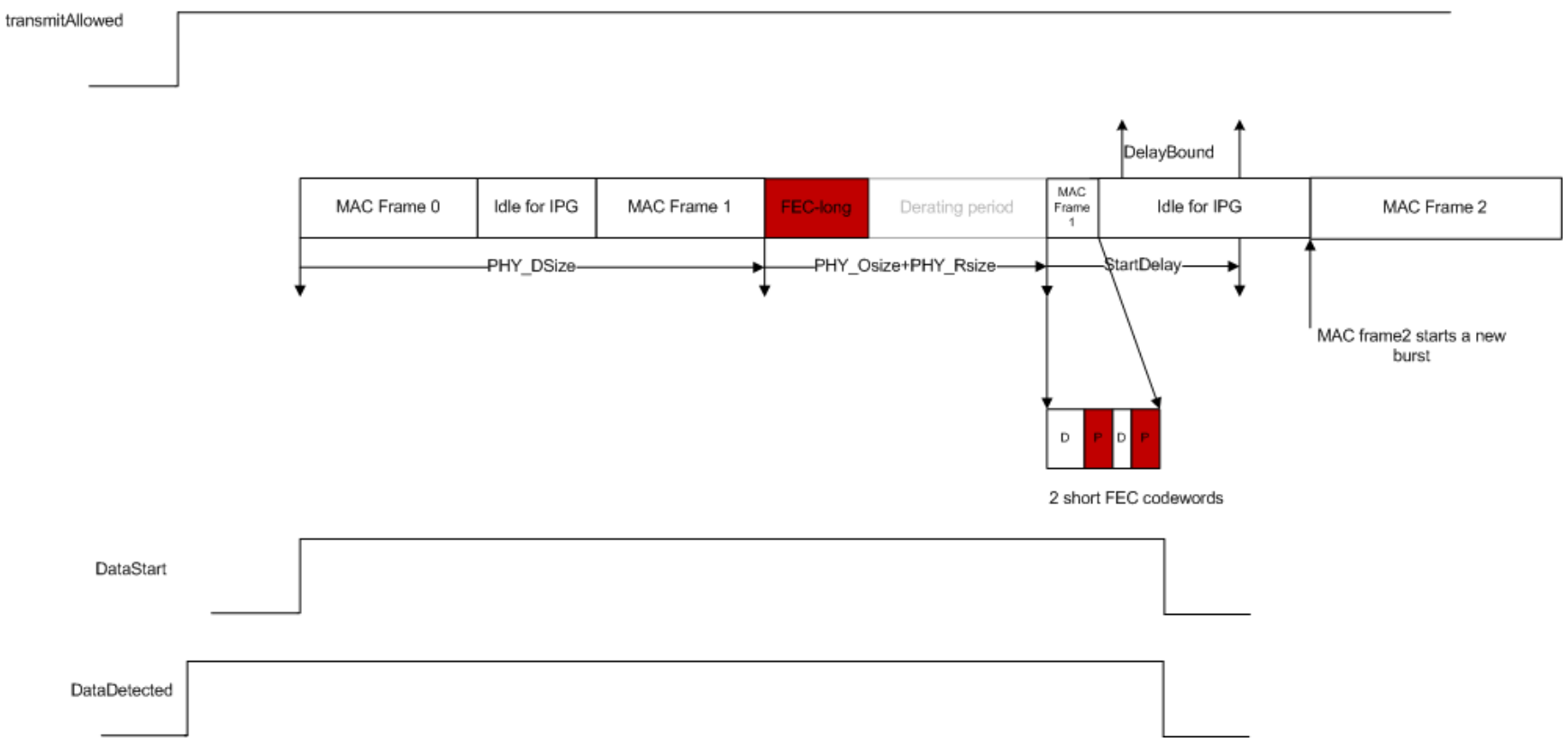
CNU Data Detector Diagram – III



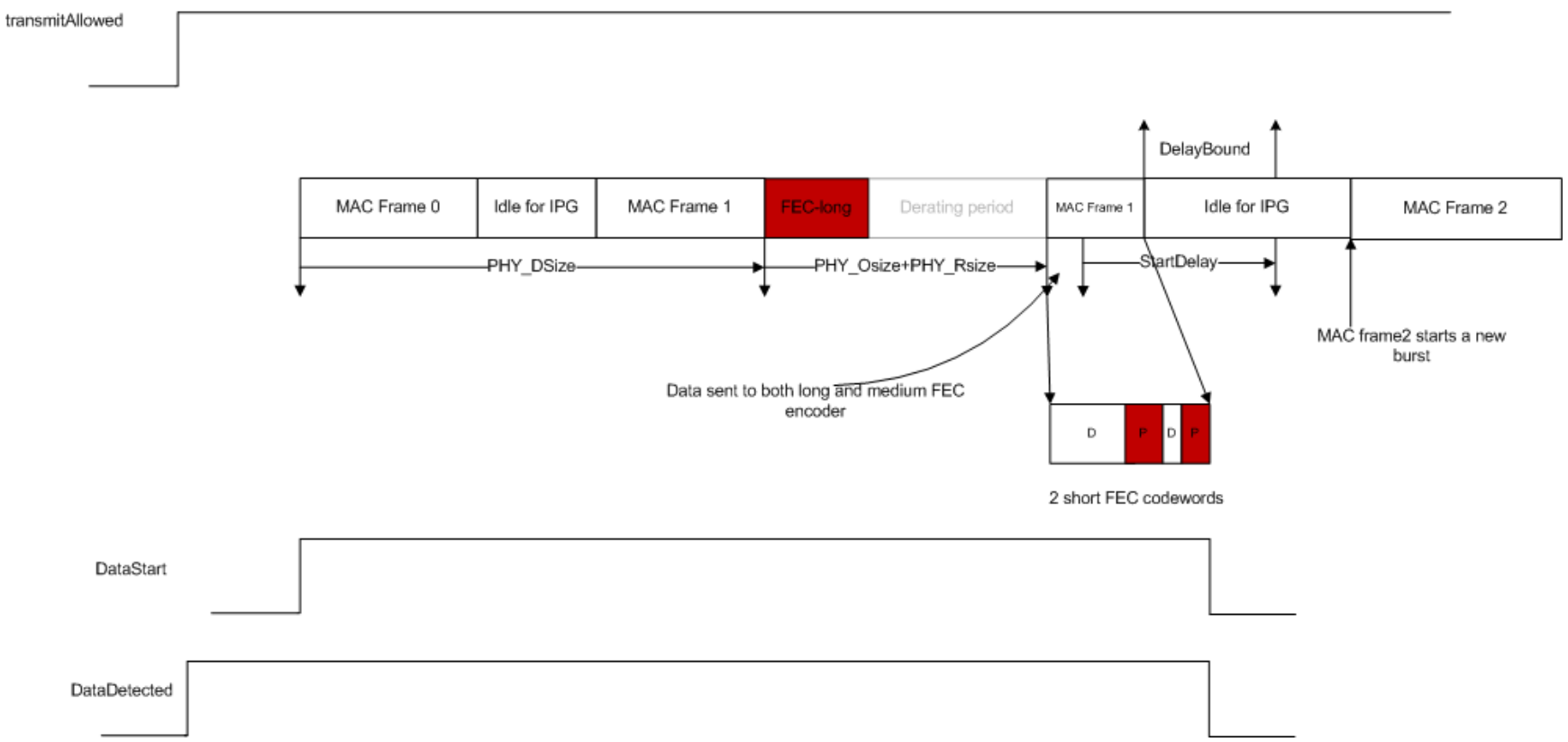
Parameters and Functions

- StartDelay: Initial delay to get FIFO filled. StartDelay=DecisionDelay+DelayBound
- DecisionDelay: Additional delay for FEC encoder
- DelayBound: the threshold of number of consecutive idle vectors for terminating a burst.
- dataPayload: encoder memory buffer for long FEC code
- dataPayload2: encoder memory buffer for medium FEC code
- dataPayload3: encoder memory buffer for small FEC code.
- [N0, N1, N2]<=FECLookUpTable(size): return the number of long, medium, shoft codewords given the size of remaining data.
- PHY_Osize and PHY_Rsize: same as in idle deletion
- DataStart: set to 1 when data detector output vectors during a burst

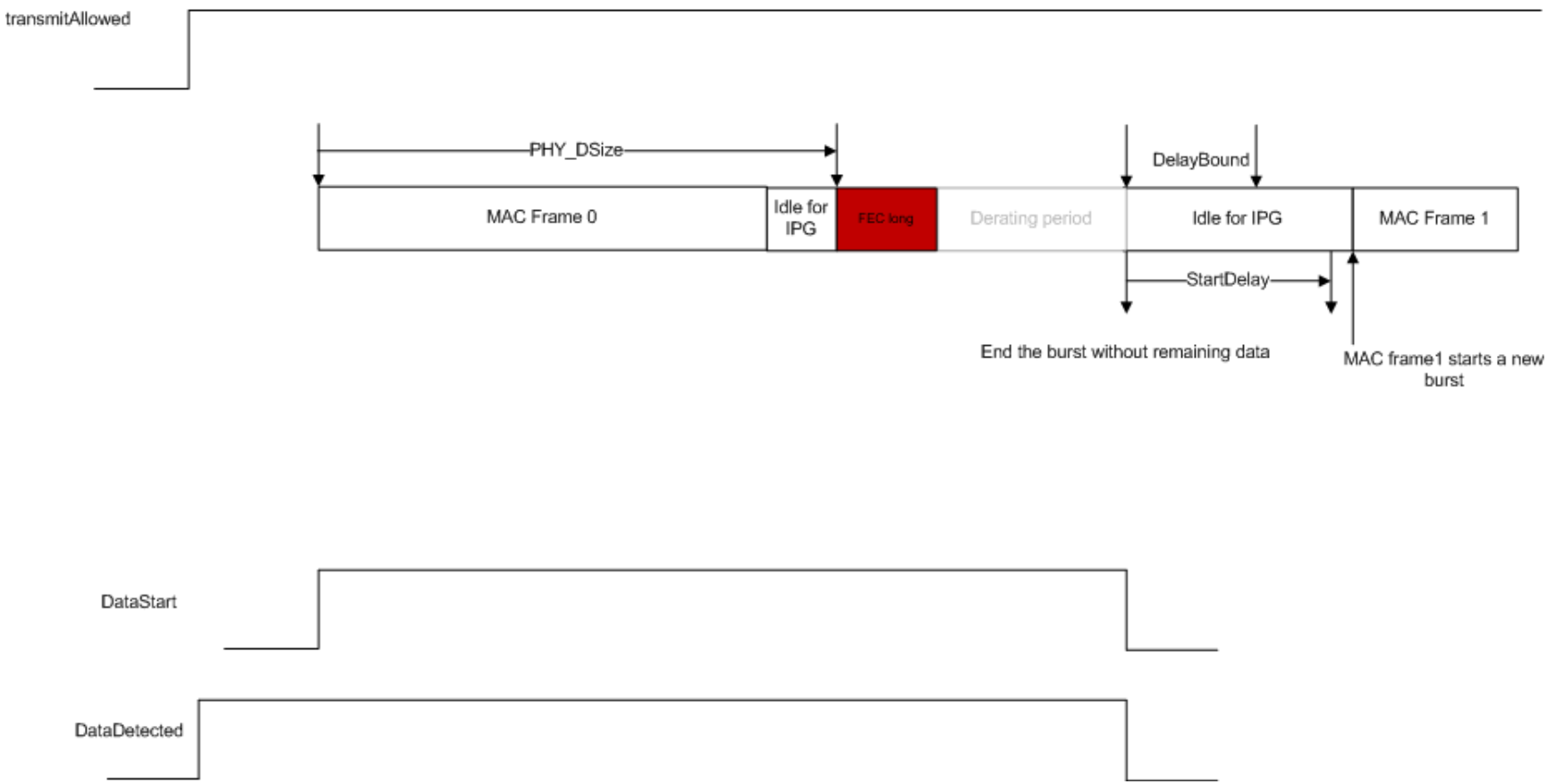
Example(1)


transmitAllowed
DelayBound
MAC Frame 0
Idle for IPG
MAC Frame 1
FEC-long
Derating period
MAC Frame 1
Idle for IPG
MAC Frame 2
PHY_DSize
PHY_Osize+PHY_Rsize
StartDelay
MAC frame2 starts a new burst
D
P
D
P
2 short FEC codewords
DataStart
DataDetected

Example (2)


transmitAllowed
DelayBound
MAC Frame 0
Idle for IPG
MAC Frame 1
FEC-long
Derating period
MAC Frame 1
Idle for IPG
MAC Frame 2
PHY_DSize
PHY_Osize+PHY_Rsize
StartDelay
MAC frame2 starts a new burst
Data sent to both long and medium FEC encoder
D
P
D
P
2 short FEC codewords
DataStart
DataDetected

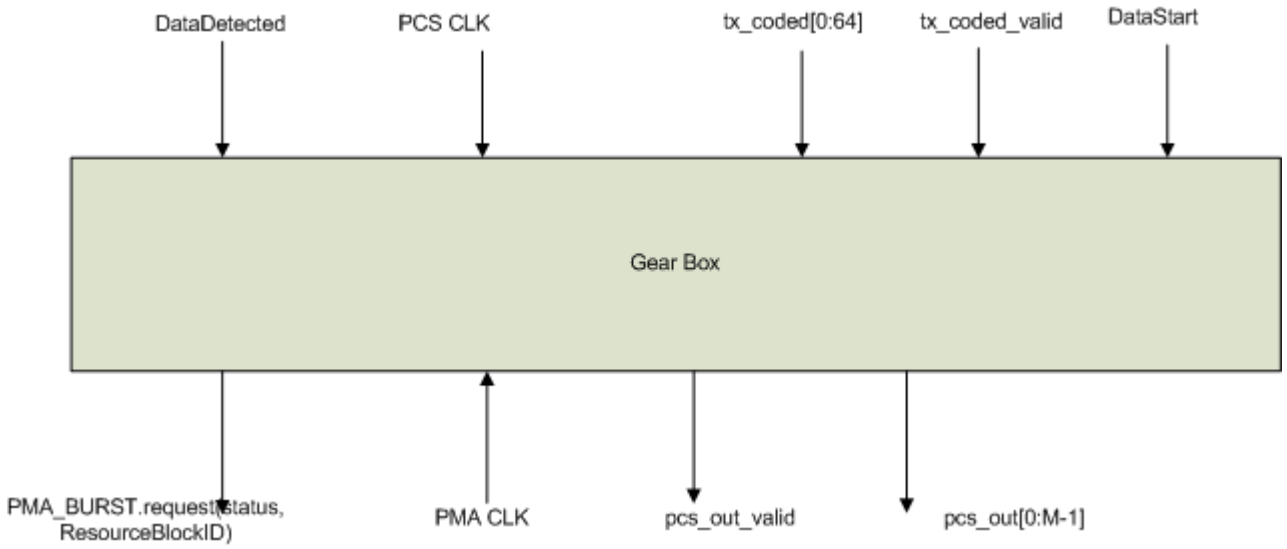
Example (3)


transmitAllowed
PHY_DSize
DelayBound
MAC Frame 0
Idle for IPG
FEC long
Derating period
Idle for IPG
MAC Frame 1
StartDelay
End the burst without remaining data
MAC frame1 starts a new burst
DataStart
DataDetected

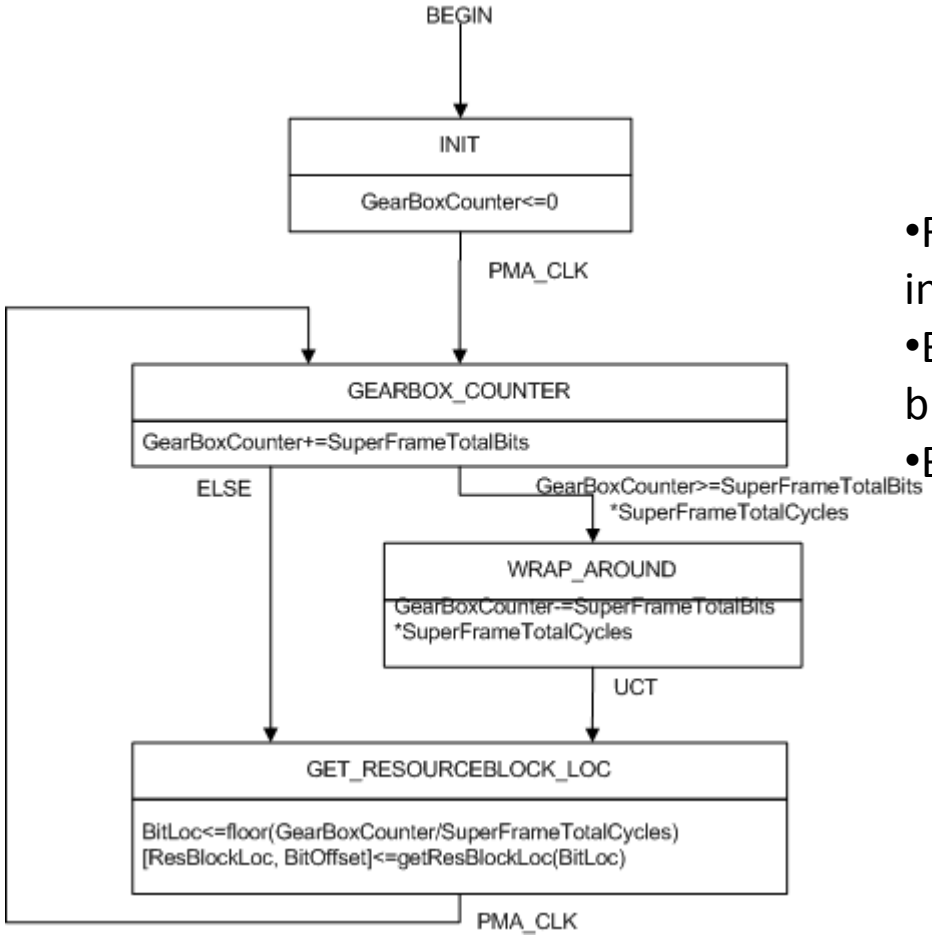
CNU Gear Box

- Unlike CLT, the CNU gearbox plays an important role in defining the 2-D burst.
- In all CNUs, the gearbox has a common counter that sweeps through a superframe at a constant bit rate, i.e. sweeps through superFrameTotalBits within SuperFrameTotalCycles.
- Upon receiving DataDetected from data detector, the gearbox locks in a snapshot of the common counter and passes down to PMA a primitive to indicate the first RB position of a burst
- Upon receiving DataStart, the gearbox input process starts buffering data detector output.
- Upon receiving zero for DataStart, the gearbox shall empty its buffer and notify PMA burst end using a primitive

CNU Gearbox Input/Output

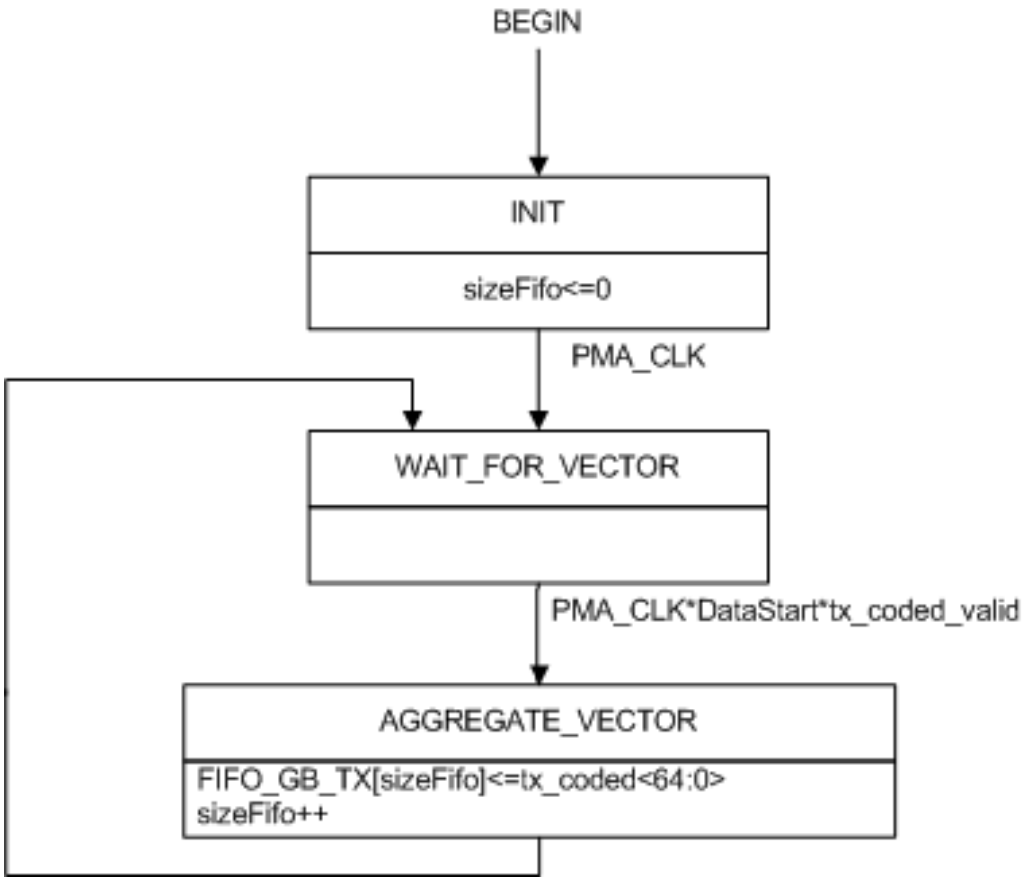


Gearbox Counter

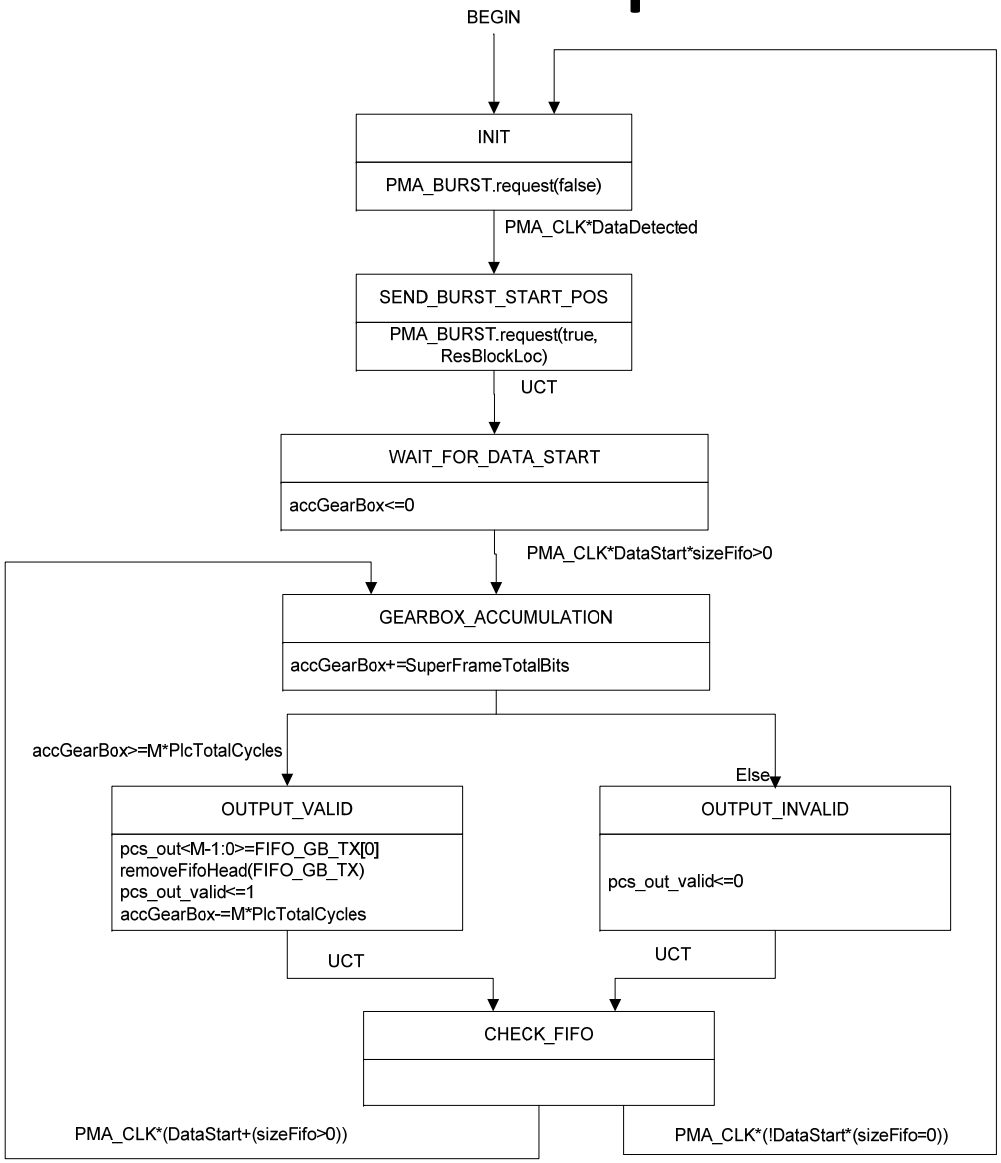


- ResBlockLoc: location of resource block in one superframe
- BitOffset: bit offset within a resource block
- BitLoc: location of the current bit.

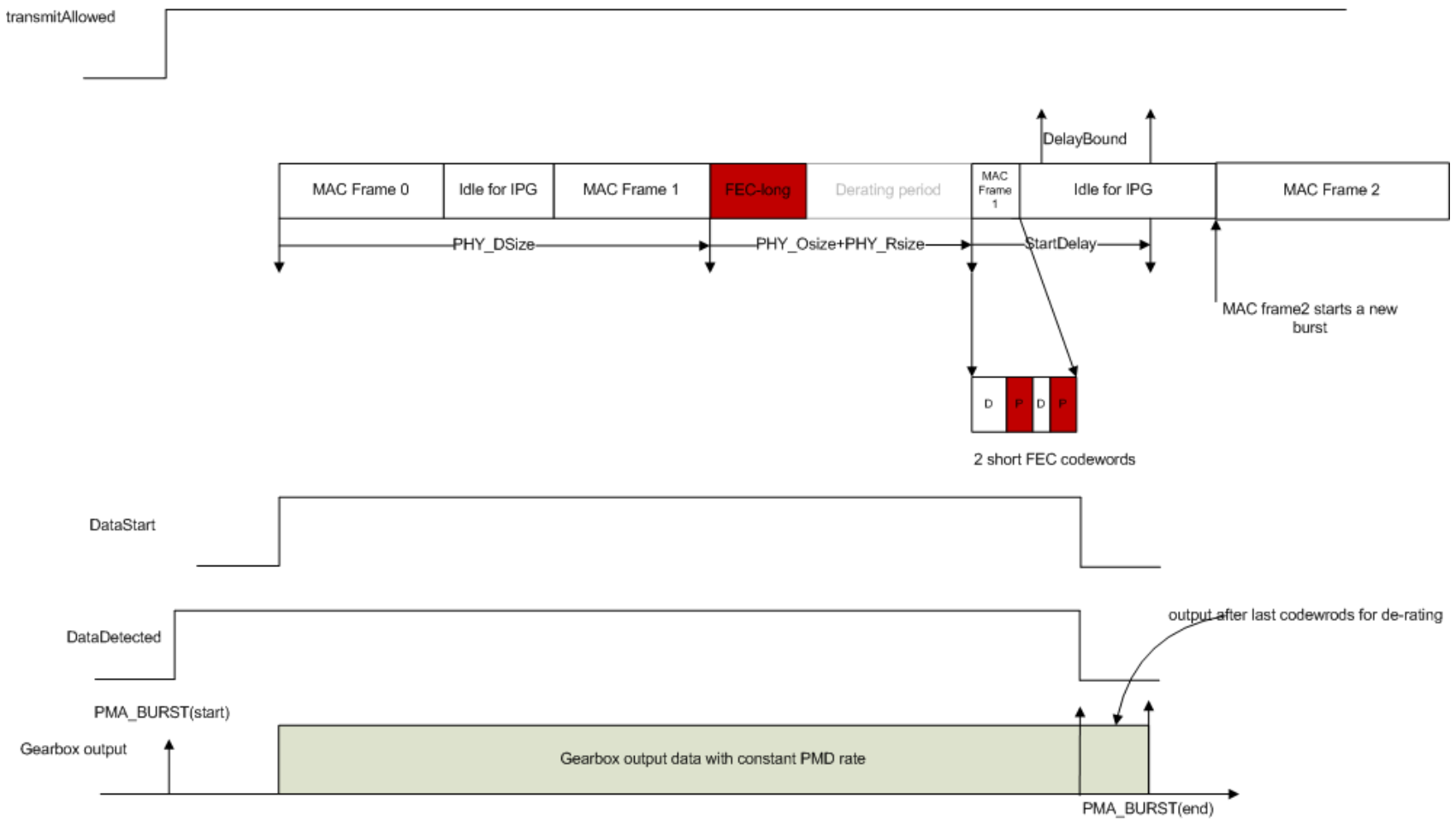
Gearbox Input Process



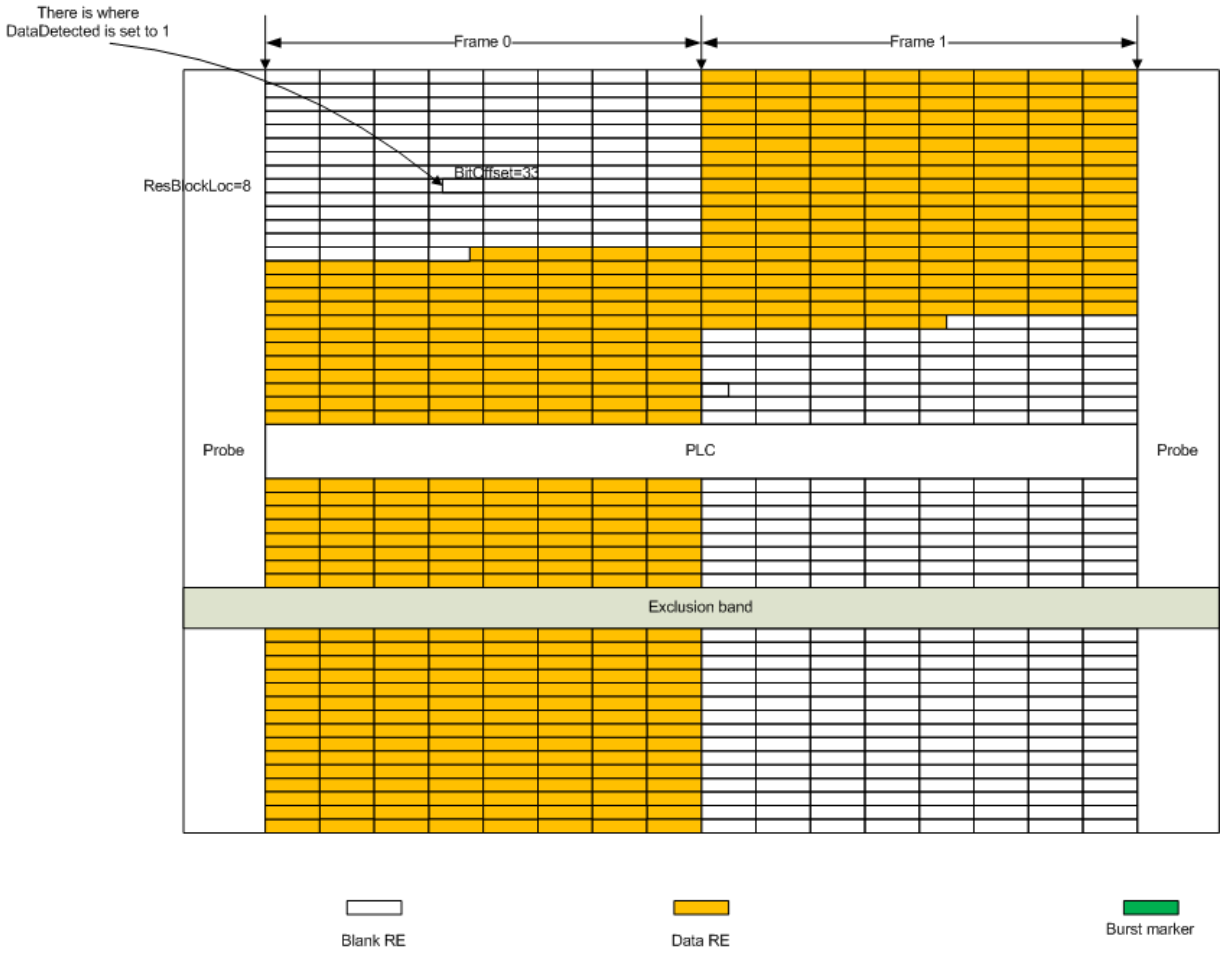
Gearbox Output Process



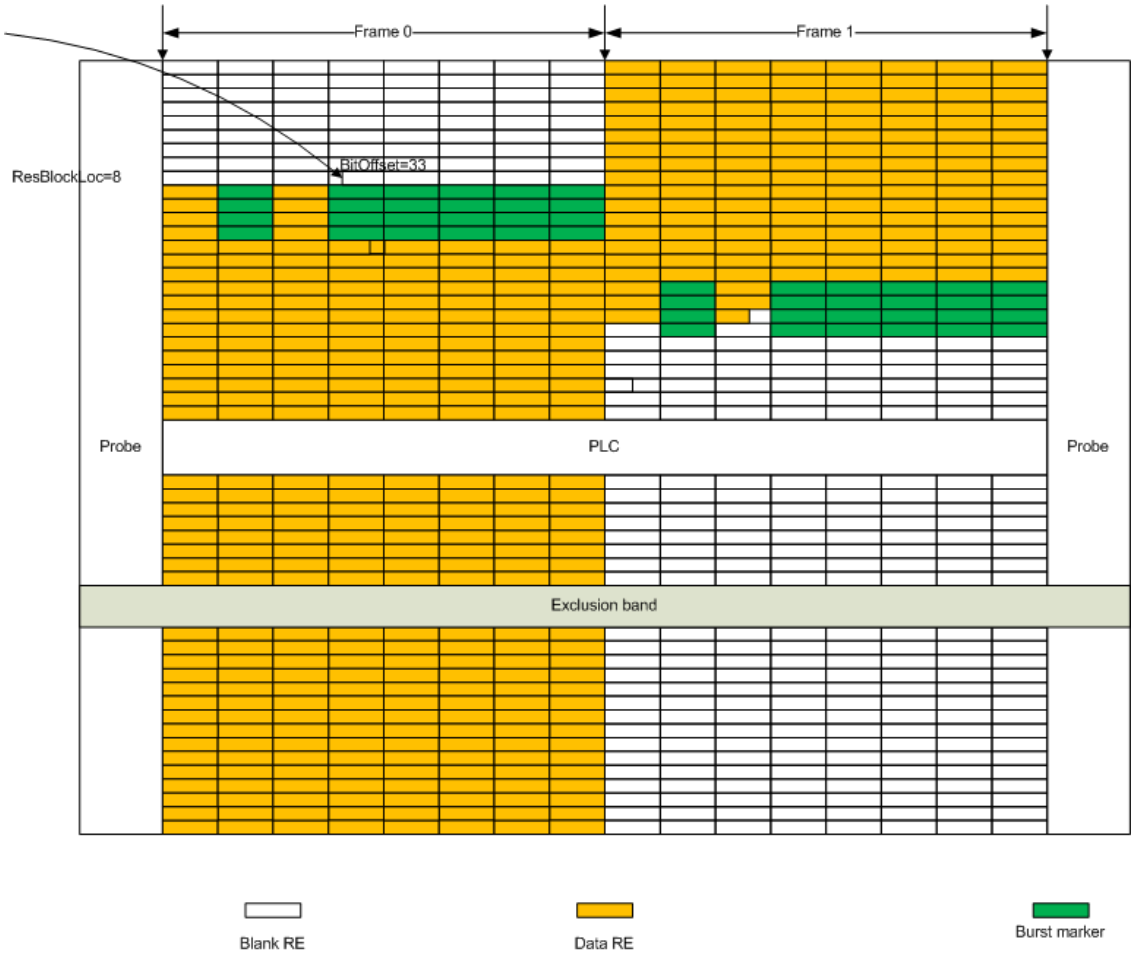
Gearbox Example



How Does PMA Map Data?



After Reposition



PMA Mapping Process

- Upon receiving PMA_BURST.request(start, ResBlockLoc), locate the first resource block, fill in start marker.
- Upon receiving burst data, place (after possible interleaving) the data at the first available position.
- Upon receiving PMA_BURST.request(end), find out the position of end marker with minimum burst end offset.