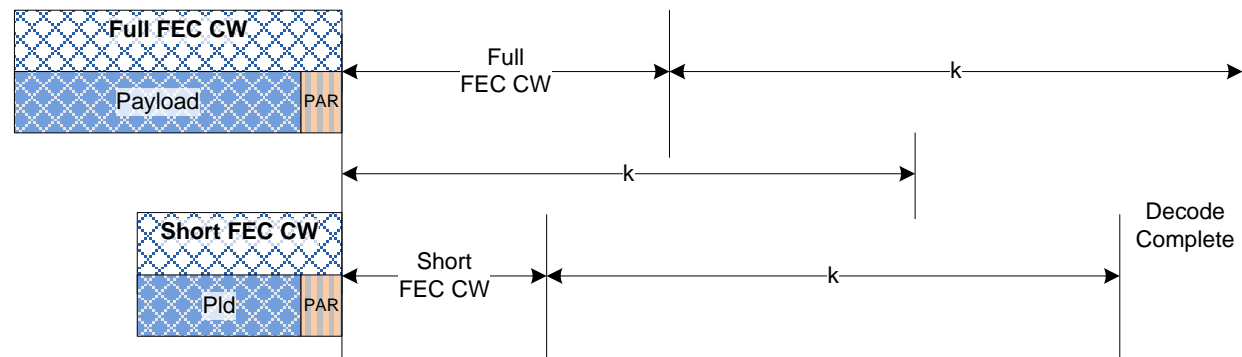


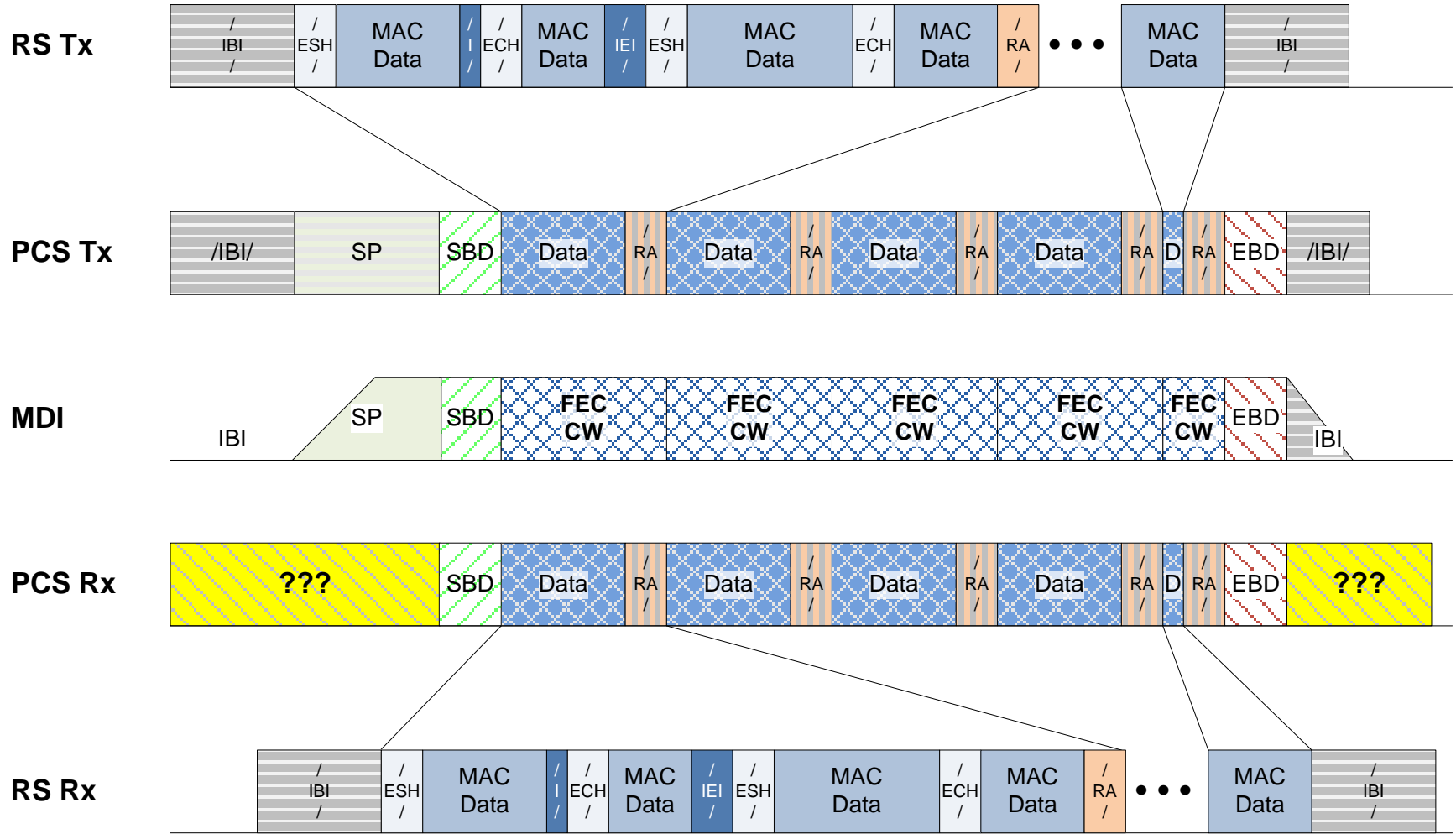
Nx25G-EPON Receive PCS

Motivation & assumptions

- ❑ Transmit side PCS is fairly stable and we now know what the framing and burst structure look like
- ❑ We need to stabilize the receive side
- ❑ Assumptions:
 - Delay due to FEC Decoding is constant (k plus the length of the decoded payload)
 - The constant FEC delay is large compared to the length of the FEC CW

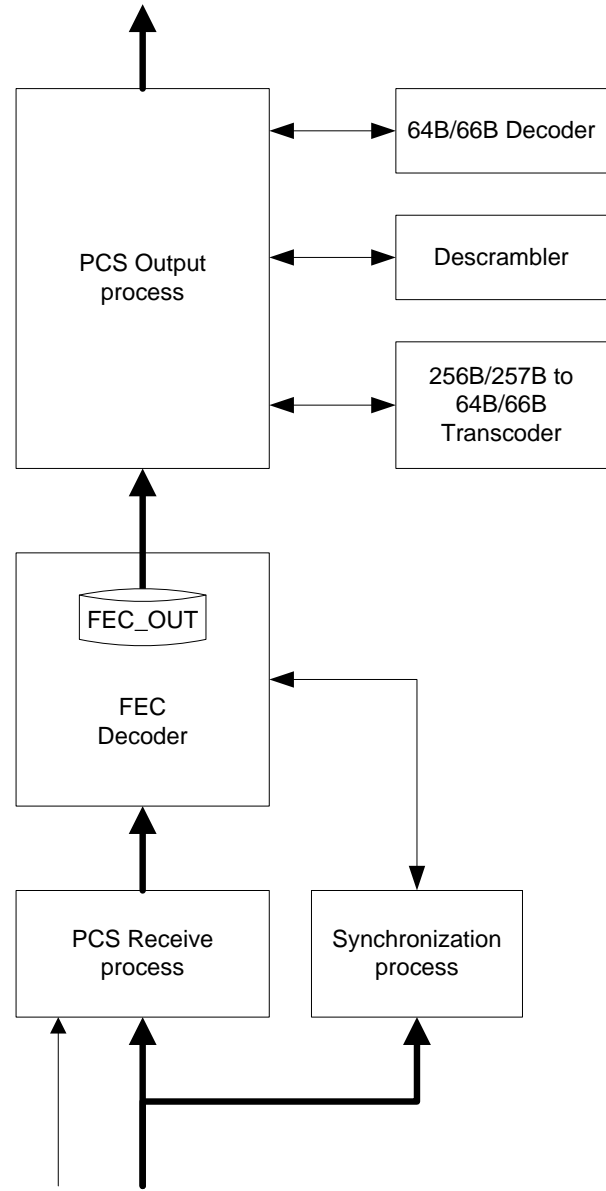
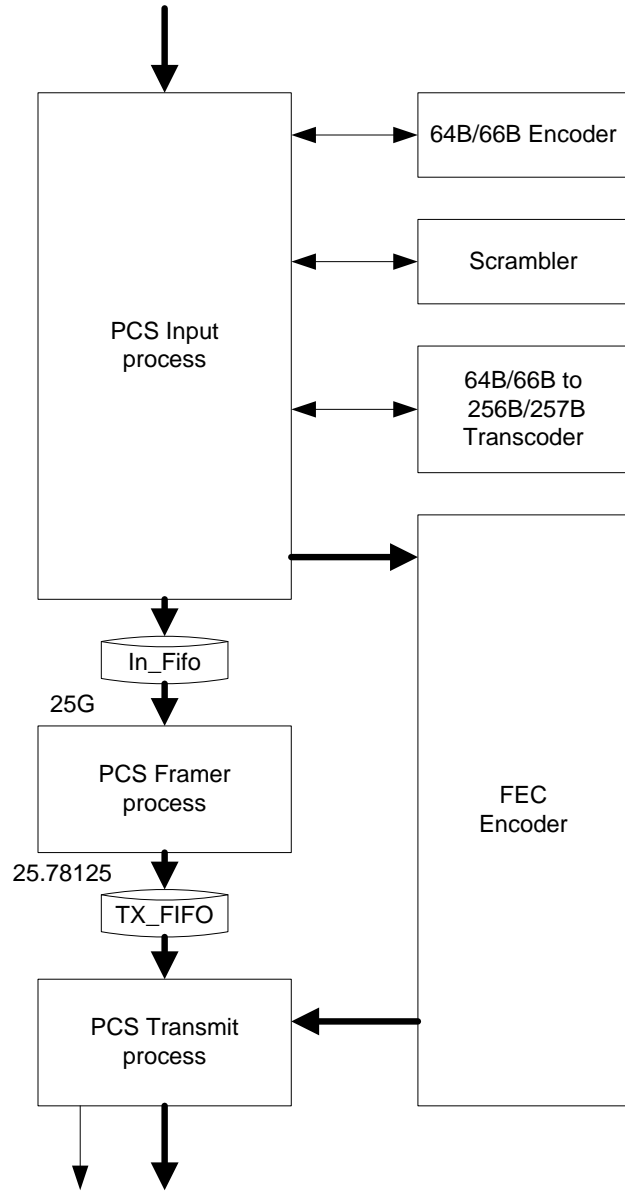


Review of burst



Proposal

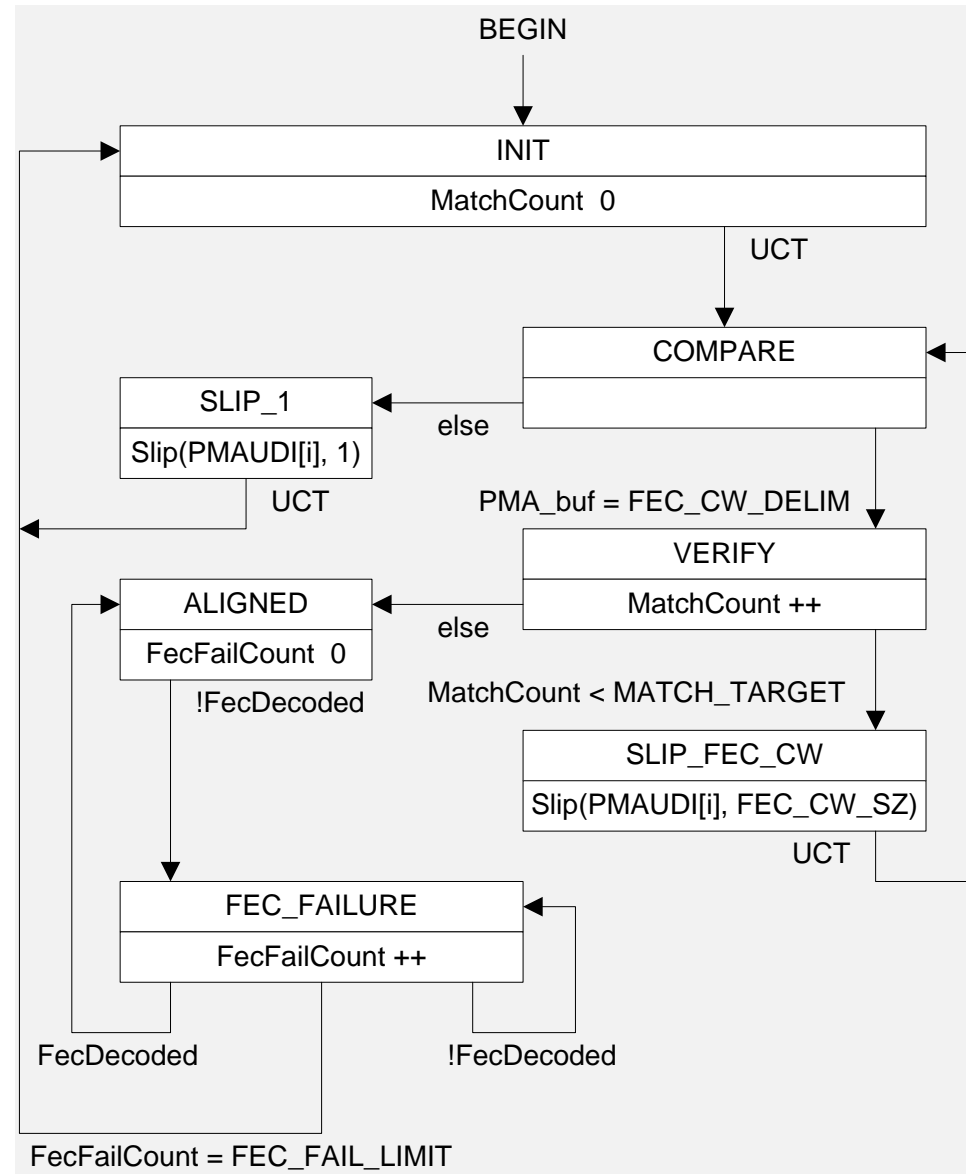
100G EPON



ONU Synchronization process (D1.2)

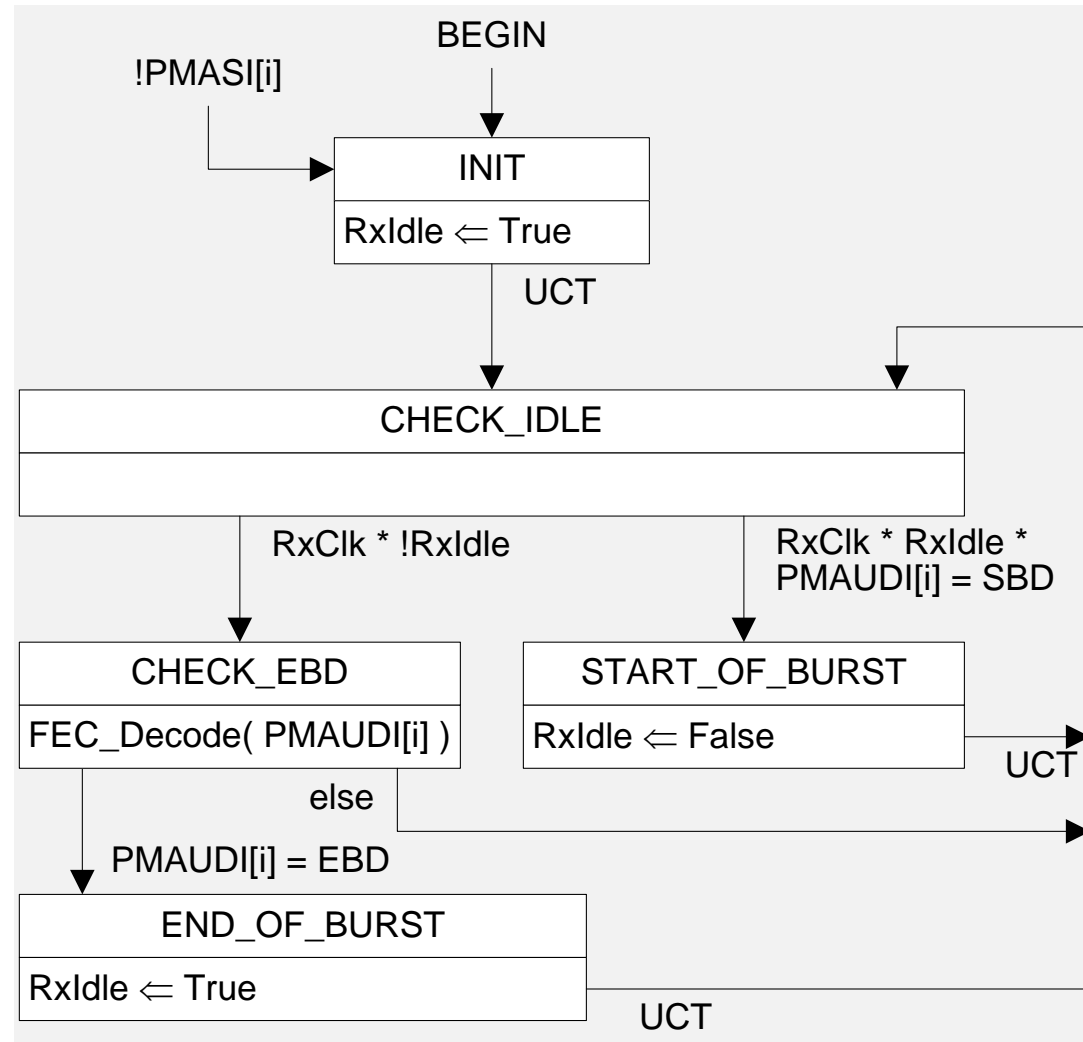
□ Per Draft 1.2

- Changed rx_buffer to PMAUDI[i] (alias for PMA_UNITDATA[i] <256:0>.indication)



PCS Receive process

- ❑ Runs at line rate (257-bit blocks)
- ❑ Detects start of burst & end of burst
- ❑ IDs everything between as data and sends to FEC Decoder
- ❑ Sends end of burst to FEC decoder (important for short CW)



PCS RX variables, constants, functions

Constants

EDB

TYPE: 256-bit vector

Value: 0x 00

The EBD constant holds the value of the end of burst delimiter.

Variables

RxCk

TYPE: Boolean

The clear on read RxCk variable is set to True for each 257-bit block loaded into the PMA_Buf.

RxIdle

TYPE: Boolean

When True the RxIdle indicates that the PCS Receive process is in the idle state and is not receiving FEC encoded data.

SBD

TYPE: 256-bit vector

The variable holds the value of SP2 or SP3, depending on the most recently provisioned synchronization pattern (see {142.2.2.2?}).

Functions

FEC_Decode()

The FEC_Decode function to passes one 257-bit block to FEC decoder.

Add the following definition to Synchronizer variables removing rx_buffer.

PMASI[i]

Alias for PMA_SIGNAL[i].indication

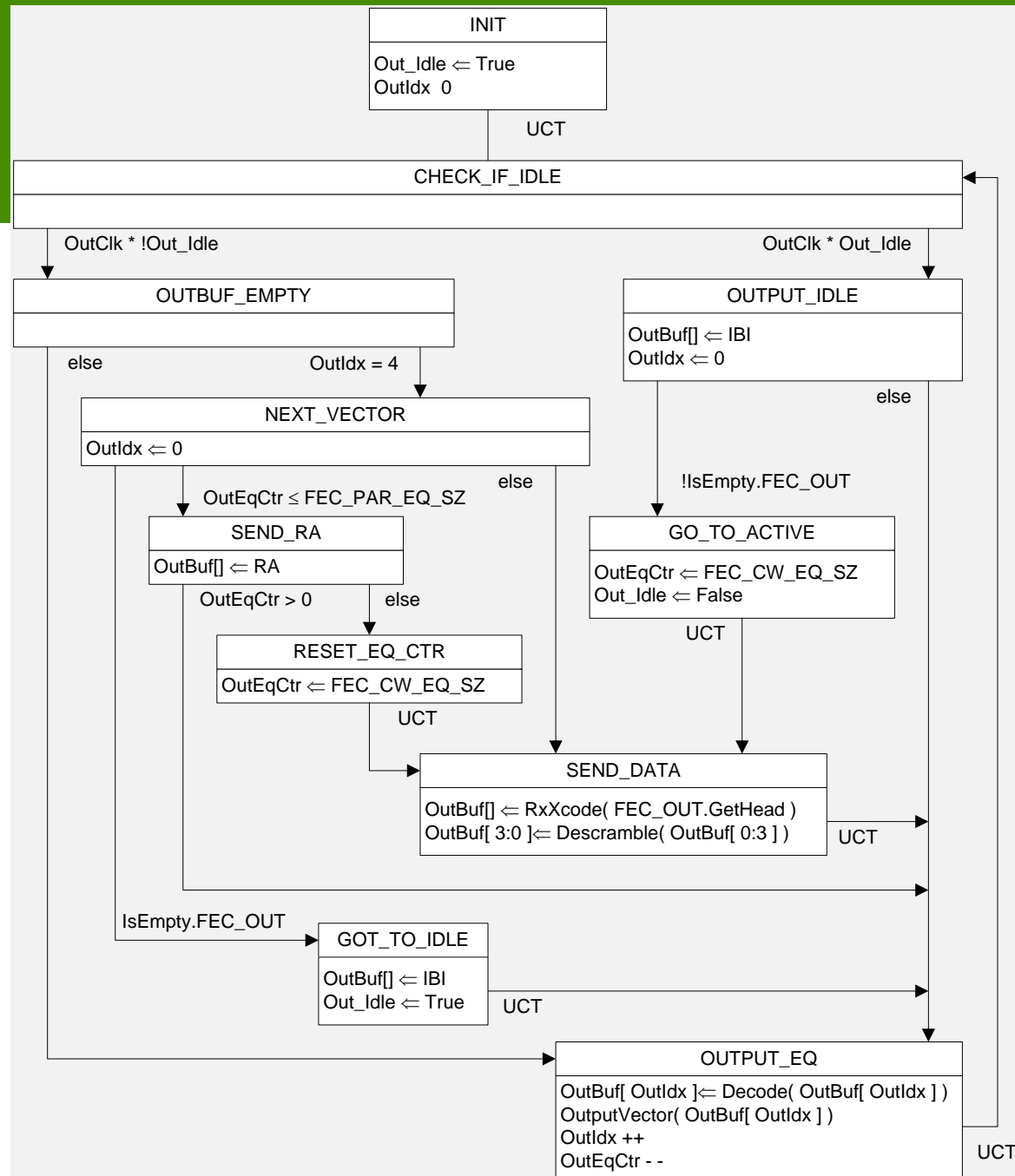
PMAUDI[i]

Alias for

PMA_UNITDATA[i]<256:0>.indication

PCS Output process

- ❑ Runs at EQ rate
- ❑ If idle outputs IBI & looks for data from FEC Decoder
- ❑ We FEC Decoder fills FEC_OUT switches to active
- ❑ Outputs FEC payload as EQs
- ❑ Outputs FEC parity placeholder
- ❑ Continues until FEC Output buffer is empty, then shifts to back to idle



Output constants & variables

Constants

FEC_CW_EQ_SZ

TYPE: integer

Value: 256

The FEC_CW_EQ_SZ represents the size of a FEC codeword in EQs

FEC_PAR_EQ_SZ

TYPE: integer

Value: 32

The FEC_PAR_EQ_SZ represents the size of the FEC parity word in EQs

IBI

TYPE: 66-bit vector

Value: 0x2-0A-0A-0A-0A-0A-0A-0A-0A

The IBI constant represents a 66-bit encoded word of Inter-Burst Idle EQ.

RA

TYPE: 66-bit vector

Value: 0x2-09-09-09-09-09-09-09-09

The RA constant represents a 66-bit encoded word of the rate adjustment EQ.

Variables

FEC_OUT

TYPE: array of 257-bit blocks

The FEC_OUT buffer holds one decoded payload from FEC decoder.

Variables (cont)

OutEqCtr

TYPE: integer

The OutEqCtr variable represents the number of EQs output from the PCS from a single FEC codeword.

Out_Idle

TYPE: Boolean

When True the Out_Idle variable indicates the PCS Output process is in the idle state and not outputting Envelope data.

OutIdx

TYPE: integer

The OutIdx variable is an index to the OutBuf variable and has a value of between 0 and 3, inclusive.

OutClk

TYPE: Boolean

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

OutBuf[0:3]

TYPE: array of 66-bit vectors

This buffer holds four 66-bit vectors of 64B/66B encoded data.

Output functions

100G EPON

Functions

OutputVector()

This function inputs one EQ (72-bit block) and outputs two 36-bit vectors over the xMII. Control is not returned to the calling state until after the second 32-bit vector is sent.

RxXcode()

This function inputs one 256B/257B encoded block, transcodes it into four 64B/66B encoded blocks and returns the result.

Decode()

This function input one 64B/66B encoded block, performs the decoding function as described in 49.2.11 and Figure 49-17, and returns the resultant 72-bit vector.

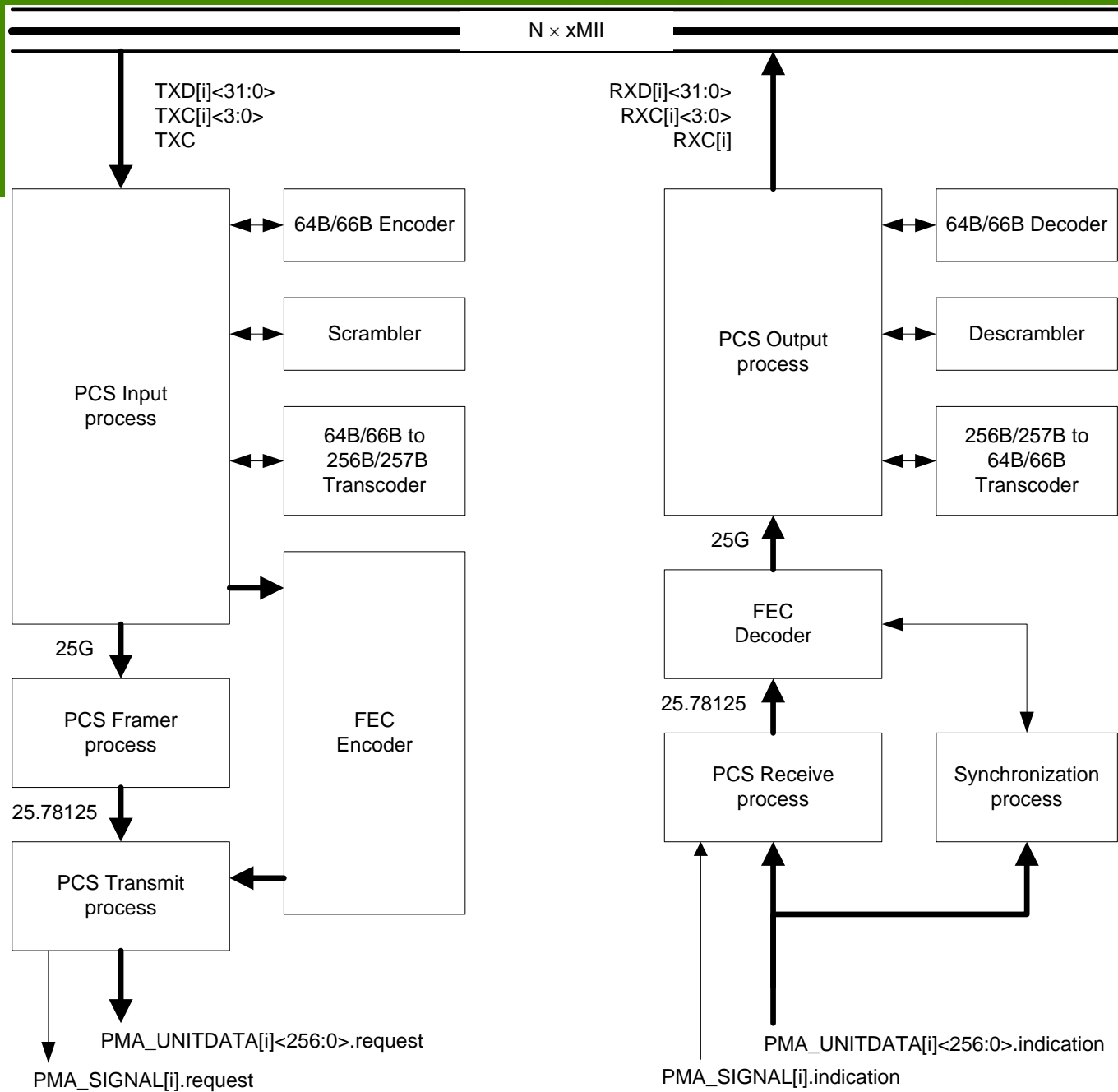
Descramble()

This function inputs one 257-bit vector, performs the descrambling function described in 49.2.10 and returns the result.

Fifo.GetHead (see {142.2.2.5.3?})

PCS FBD

Update to
Figure 142-2



THANK YOU

1
IBI
2
IBI
Data
3
IBI
Data
Data
4
IBI
Data
...
Parity
5
IBI
Data
...
Parity
Data
6
IBI
Data
...
Parity
Data
Parity
IBI

