Note to Editor: line in dark red font are FYI and not to be included in the draft

143.3.3.1 ~~Conventions~~

~~143.3.3.2~~ State diagram conventions

The body of this standard comprises state diagrams, including the associated definitions of variables, constants, and functions. In case of any discrepancies between a state diagram and descriptive text, the state diagram prevails. The notation ++ after a variable indicates it is to be incremented by 1. The notation – – after a variable indicates it is to be decremented by 1. The notation –= after a variable indicates that the counter value is to be decremented by the following value. The notation += after a variable indicates that the counter value is to be incremented by the following value. Code examples given in this clause adhere to the style of the "C" programming language. The vector notations used in the state diagrams for bit vector use 0 to mark the first received bit and so on (for example data<15:0>), following the conventions of 3.1 for bit ordering.

143.3.3.2 Application-specific parameter definitions

Some constants and variables in this sub-clause have characteristics that are application specific. For Nx25G-EPON specific parameter definitions refer to 143.4.1.3.

Editor's Note (to be removed prior to publication) in the future, references to other applications-specific parameters are to be added in this subclause.

143.3.3.3 Constants

Location: Pg 108 Cl 143.3.3.3 Line 52. Change as marked.

~~ADJ_BLOCK_SIZE~~
~~TYPE: {TBD}~~
~~Value: {TBD}~~
~~{ description }~~

ADJ_BLOCK_SIZE
TYPE: integer
Value: application specific (see 143.3.3.2)
The ADJ_BLOCK_SIZE constant represents the block size (in EQs) that is used to adjust the rate between the MAC and the PHY in the MCRS-based device.

Location: Pg 109 Cl 143.3.3.3 Add new definition.

NUM_CH
TYPE: integer
Value: application specific (see 143.3.3.2)
The NUM_CH constant represents the number of channels supported by an MCRS-based device.

Location: Pg 109 Cl 143.3.3.3 Line 19. Change as marked.

~~RATE_ADJ_SIZE~~
~~TYPE: {TBD}~~
~~Value: {TBD}~~
~~{ description }~~

RATE_ADJ_SIZE

TYPE: integer

Value: application specific (see 143.3.3.2)

The RATE_ADJ_SIZE constant represents the number of EQs within the ADJ_BLOCK_SIZE block during which the MAC transmission is deferred. The effective MAC rate is equal to <nominal MA rate> × (1- RATE_ADJ_SIZE / ADJ_BLOCK_SIZE).

Location: Pg 109 Cl 143.3.3.4 Line 25. Change as marked.

ch

TYPE: ~~2-bit~~ integer

The *ch* variable represents the index of a specific xMII channel ~~or~~bound to an instance of MCRS Transmit or an MCRS Receive process . The values of *ch* range from 0 to (NUM_CH – 1). Within each instance of MCRS Transmit or MCRS Receive process, the ~~corresponding ENV_TX buffer, or ENV_RX buffer column.~~ value of *ch* remains constant.

Location: Pg 109 Cl 143.3.3.4 Line 34. Change as marked.

ENV_TX~~[c][r]~~

TYPE: array of 72-bit ~~binary array~~ vectors

The *ENV_TX* buffer is used to transfer information between the MCRS Input Process and the MCRS Transmit Process. ~~Each~~In this buffer, each cell, represented by the variables *ENV_TX[c][r]*, ~~in this buffer~~ stores one EQ (a 72- bit vector) of information. ~~The buffer has N columns (c) and two rows (r). The~~ The number of columns in ENV_TX buffer is ~~dependent on the~~NUM_CH (see 143.3.3.3). The maximum number of ~~channels supported. For 100 Gb/s devices N = 4, for 50 Gb/s devices N = 2, and for 25 Gb/s devices N = 1.~~rows is 64, as determined by the size of EPAM field in Envelope Header (see 143.3.2). For some applications, fewer rows may be sufficient (see application-specific *ENV_TX* definition in 143.3.3.2). The buffer is filled ~~in a cyclic pattern row-by-row. The source LLID for each cell is determined by the~~ sequentially by the MCRS~~_CTRL[].request() primitive.~~ Input process and is emptied in parallel by NUM_CH instances of MCRS Transmit process. For additional details, refer to 143.2.5.3.

Location: Pg 110 Cl 143.3.3.4 Line 16. Add the following:

rCol

TYPE: integer

The *rCol* variable represents the *ENV_TX* buffer column currently being read by the MCRS Transmit process. Each column corresponds to a separate transmission channel, i.e., a separate xMII interface.

Location: Pg 110 Cl 143.3.3.4 Line 17. Change as marked.

rRow

TYPE: 6-bit integer

The *rRow* variable ~~rRow~~ represents the row in the *ENV_TX* buffer currently being read by the MCRS Transmit ~~Process~~process. The value of this variable is synchronized to *wRow* and is equal *wRow* - 1.

Location: Pg 110 Cl 143.3.3.4 Line 31. Change as marked.

wCol

TYPE: ~~2-bit~~ integer

The *wCol* variable represents the *ENV_TX* buffer column currently being written by the MCRS Input ~~Process~~process. Each column corresponds to a separate transmission channel, i.e., a separate xMII interface.

wRow

TYPE: 6-bit integer

The variable *wRow* represents the *ENV_TX* buffer row index currently being written by the MCRS Input ~~Process~~process. The value of *rRow* is synchronized to this variable and is equal to *wRow* - 1.

Location: Pg 116 Cl 143.3.4.3 Line 5. Change as marked.

ENV_RX~~[c][r]~~

TYPE: array of 72-bit ~~binary array~~vectors

The *ENV_RX* buffer is used to transfer information between the MCRS Receive ~~Process~~process and the MCRS Output ~~Process~~process. In this buffer, each cell, represented by the ~~variables~~variable *ENV_RX[c][r]*, ~~in this buffer~~ stores one EQ (a 72- bit vector) of information. ~~The buffer has N columns (c) and M rows (r). The~~ The number of columns in *ENV_RX* buffer is ~~dependent on the~~ NUM_CH (see 143.3.3.3). The maximum number of ~~channels supported. For 100 Gb/s devices N = 4, for 50 Gb/s devices N = 2, and for 25 Gb/s devices N = 1. The value of M~~ rows is 64, as determined by the size of EPAM field in Envelope Header (see 143.3.2). For some applications, fewer rows may be sufficient (see application-specific ~~but must be greater than or equal to the maximum value of EnvPam.~~*ENV_RX* definition in 143.3.3.2). The buffer is filled in ~~a cyclic pattern row-by-row~~parallel by ~~the~~NUM_CH instances of MCRS Receive process and is emptied sequentially by the MCRS Output process. For additional details, refer to 143.2.5.3.

Location: Pg 116 Cl 143.3.4.3 Line 31. Change as marked.

rCol

TYPE: ~~2-bit~~ integer

The *rCol* variable represents the *ENV_RX* buffer column currently being read by the MCRS Output ~~Process~~process. Each column corresponds to a separate reception channel, i.e., a separate xMII interface.

rRow

TYPE: 6-bit integer

The *rRow* variable represents the *ENV_RX* buffer row index currently being read by the MCRS Output ~~Process.~~process.

Location: Pg 116 Cl 143.3.4.3 Line 52. Add the following:

wCol

TYPE: integer

The *wCol* variable represents the *ENV_RX* buffer column currently being written by the MCRS Receive process. Each column corresponds to a separate reception channel, i.e., a separate xMII interface.

wRow

TYPE: 6-bit integer

The *wRow* variable represents the *ENV_RX* buffer row index currently being written by the MCRS Receive process.

Location: Pg 122 Cl 143.4.1.3 Line 13. Add the following:

143.4.1.3 Nx25G-EPON application-specific parameters

143.4.1.3.1 Constants

ADJ_BLOCK_SIZE

Value: 257

NUM_CH

Value: 1 for devices supporting only 10 Gb/s or 25 Gb/s operation over a single channel;
2 for devices supporting 50 Gb/s operation over two channels.

RATE_ADJ_SIZE

Value: 33

143.4.1.3.2 Transmit variables

ENV_TX

Since there is no timing jitter or channel skew to be removed at the transmitting device, the size of *ENV_TX* buffer can be reduced to only two rows. If this optimization is implemented, the variables *rRow* and *wRow* are represented by 1-bit integers.

143.4.1.3.3 Receive variables

ENV_RX

In a typical Nx25G-EPON deployment scenario, the maximum timing jitter and channel skew are expected to be low enough to allow implementations of *ENV_RX* buffer with only 32 rows, as opposed to the default 64 rows. If such an optimization is implemented, the variables *rRow* and *wRow* are represented by 5-bit integers.