# Alignment Marker Format Updates

Adrian Butter (on behalf of the P802.3bs Logic Ad Hoc team)

GLOBALFOUNDRIES®

# Contributors

- Mark Gustlin, Xilinx
- Pete Anslow, Ciena
- Ben Jones, Xilinx
- Jeff Slavick, Avago Technologies
- Eric Baden, Broadcom
- Juan Carlos Calderon, Inphi
- David Ofelt, Juniper Networks
- Tongtong Wang, Huawei
- Phil Sun, Credo Semiconductor

# Supporters

- Gary Nicholl, Cisco

# Introduction

- ➢ Alignment Marker format definition is one of the Logic Ad Hoc BTIs to resolve in support of a technically complete P802.3bs draft:

  - Sublayer delay constraints are TBD, same with skew limitations
  - Define 400G AM fields
    - Made progress, but some more work to do on this
  - Exact criteria for achieving AM lock
    - AMP_valid

- ➢ Since December 2015, several Logic Ad Hoc team members have made ***excellent contributions*** towards resolving this BTI! For example:
  - ○ Toward 400GbE AMs and PAM4 test pattern characteristics
  - ○ 400GbE AMs revised proposal
  - ○ Proposed AM Format
- ➢ Logic Ad Hoc team consensus has been attained on Alignment Marker formatting.
- ➢ Comment #3 was logged against P802.3bs D1.2 to support technical updates capturing this consensus.
  - ○ Comment #2 was also logged identifying an ancillary issue found during this activity.
- ➢ Editing details supporting these technical updates are contained in “butter_3bs_01_0316” to be presented during the 802.3 March plenary in Macau during the 802.3bs Logic Track meeting.
- ➢ This presentation identifies those updates with supporting material...

# D1.2 Comments Regarding Alignment Markers

*Cl* **119** *SC* **119.2.4.4** *P* **98** *L* **6** # 3

Butter, Adrian GLOBALFOUNDRIES

*Comment Type* **TR** *Comment Status* **X**

The alignment marker encodings in Table 119-1 contain many "TBDs". Further analysis of this alignment marker structure (with 64-bit common part and 56-bit unique part) reveals undesirable clock content which is reduced using a shorter alignment marker (with 48-bit common part and 48-bit unique part). To reduce the complexity of alignment marker processing logic for the shorter marker, as well as increase format compability of the shorter marker with that defined in 802.3bj, padding based on PRBS9 sequences is both interleaved with and appended to the marker. Refer to http://www.ieee802.org/3/bs/public/adhoc/logic/feb9_16/gustlin_01_0216_logic.pdf for details.

*Cl* **119** *SC* **119.2.4.5** *P* **100** *L* **32** # 2

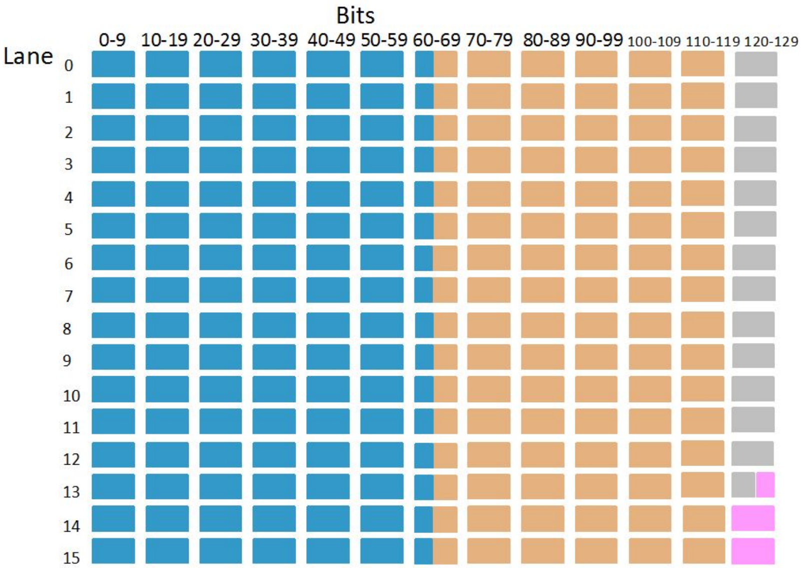
Butter, Adrian GLOBALFOUNDRIES

*Comment Type* **T** *Comment Status* **X**

There is no clear connection between variables tx_scrambled_am and tx_scrambled_am_j. Also, defining tx_scrambled_am as 257 bits does not align with the width implied in 119.2.4.4, page 97, line 25.

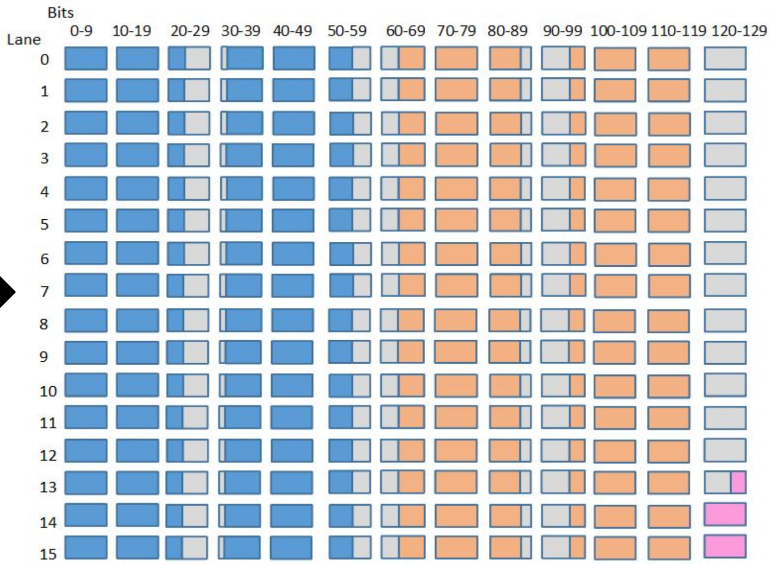
# Objectives

Change alignment marker format from:




- Common marker (64 bits)
- Unique marker (56 bits)
- Pad (136 bits)
- Next 257-bit block

… to:




- Common marker (48 bits)
- Unique marker (48 bits)
- Pad (520 bits)
- Next 257-bit block

# Objectives

Map alignment markers to FEC codewords in the following manner:

| Lane \ Bits | 0-9 | 10-19 | 20-29 | 30-39 | 40-49 | 50-59 | 60-69 | 70-79 | 80-89 | 90-99 | 100-109 | 110-119 | 120-129 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | m513 | m505 | m497 | m489 | m481 | m473 | m465 | m457 | m449 | m441 | m433 | m425 | m417 |
| 1 | m513 | m505 | m497 | m489 | m481 | m473 | m465 | m457 | m449 | m441 | m433 | m425 | m417 |
| 2 | m512 | m504 | m496 | m488 | m480 | m472 | m464 | m456 | m448 | m440 | m432 | m424 | m416 |
| 3 | m512 | m504 | m496 | m488 | m480 | m472 | m464 | m456 | m448 | m440 | m432 | m424 | m416 |
| 4 | m511 | m503 | m495 | m487 | m479 | m471 | m463 | m455 | m447 | m439 | m431 | m423 | m415 |
| 5 | m511 | m503 | m495 | m487 | m479 | m471 | m463 | m455 | m447 | m439 | m431 | m423 | m415 |
| 6 | m510 | m502 | m494 | m486 | m478 | m470 | m462 | m454 | m446 | m438 | m430 | m422 | m414 |
| 7 | m510 | m502 | m494 | m486 | m478 | m470 | m462 | m454 | m446 | m438 | m430 | m422 | m414 |
| 8 | m509 | m501 | m493 | m485 | m477 | m469 | m461 | m453 | m445 | m437 | m429 | m421 | m413 |
| 9 | m509 | m501 | m493 | m485 | m477 | m469 | m461 | m453 | m445 | m437 | m429 | m421 | m413 |
| 10 | m508 | m500 | m492 | m484 | m476 | m468 | m460 | m452 | m444 | m436 | m428 | m420 | m412 |
| 11 | m508 | m500 | m492 | m484 | m476 | m468 | m460 | m452 | m444 | m436 | m428 | m420 | m412 |
| 12 | m507 | m499 | m491 | m483 | m475 | m467 | m459 | m451 | m443 | m435 | m427 | m419 | m411 |
| 13 | m507 | m499 | m491 | m483 | m475 | m467 | m459 | m451 | m443 | m435 | m427 | m419 | m411 |
| 14 | m506 | m498 | m490 | m482 | m474 | m466 | m458 | m450 | m442 | m434 | m426 | m418 | |
| 15 | m506 | m498 | m490 | m482 | m474 | m466 | m458 | m450 | m442 | m434 | m426 | m418 | |

mNNN (red) 10-bit Message symbol NNN of FEC codeword A.

mNNN (green) 10-bit Message symbol NNN of FEC codeword B.

# Comment #3 Update - 119.2.4.4 Paragraph 1

## 119.2.4.4 Alignment marker mapping and insertion

In order to support deskew and reordering of the 16 individual PCS lanes at the receive PCS, alignment markers are added periodically for each PCS lane. The alignment marker for each PCS lane is a ~~unique 120~~ fixed 96-bit block. The alignment markers for all PCS lanes are inserted as a group, aligned to the beginning of ~~a~~ two FEC blocks, and interrupt any data transfer that is already in progress. The 96-bit alignment marker for each PCS lane is interleaved with 24 pad bits to achieve alignment marker field positioning which mimics that defined in 91.5.2.6. An additional 136-bit pad is appended to the alignment markers to yield the equivalent of eight 257-bit blocks. The pad bits shall be set to a free running PRBS9 pattern, defined by the polynomial $x^9 + x^5 + 1$, which advances by 520 bits between alignment marker mapping and insertion sequences. The initial value of the PRBS9 pattern generator may be any pattern other than all zeros. The pad shall not be checked on receive.

# Comment #3 Update - 119.2.4.4 Paragraph 2

Room for the alignment markers is created by the transmit PCS (see 119.2.4.1). Special properties of the alignment markers are that they are not scrambled, do not conform to the encoding rules as outlined in Figure 82-5 and are not transcoded. This is possible because the alignment markers are added after encoding, transcoding, and scrambling, and removed before descrambling, transcoding, and 64B/66B decoding. The alignment markers are not scrambled in order to allow the receiver to find the alignment markers, deskew the PCS lanes, and reassemble the aggregate stream before descrambling is performed. The alignment markers themselves are formed from a known pattern that is defined to be balanced and with many transitions and therefore scrambling is not necessary. ~~The group of alignment markers shall be inserted so they appear every 163 840 257-bit blocks. The variable tx_scrambled_am is created by inserting the group of alignment markers in the variable tx_scrambled. Alignment marker mapping and repetition rate are shown in Figure 119–5 and Figure 119–6.~~

# Comment #3 Update - 119.2.4.4 Paragraph 3, Figure 119-4

The format of ~~the~~ each PCS lane's alignment marker~~s~~ with 24 pad bits interleaved is shown in Figure 119–4. There is a portion that is common across all alignment markers (designated as CM$_0$ to CM$_{\sout{7}5}$), and then a unique portion per PCS lane (designated as UM$_0$ to UM$_{\sout{6}5}$). Common synchronization logic independent of the received PCS lane number can be used with the common part of the alignment marker.

Bit Position: 0 | 23 24 | 31 32 | 55 56 | 63 64 | 87 88 | 95 96 | 119

| {CM0, CM1, CM2} | Pad | {CM3, CM4, CM5} | Pad | {UM0, UM1, UM2} | Pad | {UM3, UM4, UM5} |
|---|---|---|---|---|---|---|

Common Marker = {CM0, CM1, CM2, CM3, CM4, CM5}; Unique Marker = {UM0, UM1, UM2, UM3, UM4, UM5}

**Figure 119-4 – Alignment marker format**

**<Replaces current Figure 119-4>**

# Comment #3 Update - 119.2.4.4 Paragraph 4, Table 119-1

The content of the alignment markers <u>without 24 pad bits interleaved</u> shall be as shown in Table 119–1. The contents depend on the PCS lane number and the octet number, with the first ~~64~~ <u>48</u> bits being identical across all alignment markers to allow for common synchronization across lanes. The format shown in Table 119–1 is how the alignment markers appear on the PCS lanes at the PMA service interface. In the FEC codewords, they appear in a permuted format due to the codeword interleaving that occurs before FEC codewords are distributed to PCS lanes.

**Table 119-1 – 400GBASE-R Alignment marker encodings**

| PCS lane number | Encoding | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | {CM0,CM1,CM2,CM3,CM4,CM5,UM0,UM1,UM2,UM3,UM4,UM5} | | | | | | | | | | | |
| 0 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x9E, | 0xEB, | 0x27, | 0x61, | 0x14, | 0xD8 |
| 1 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x50, | 0x74, | 0x88, | 0xAF, | 0x8B, | 0x77 |
| 2 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0xB4, | 0xB7, | 0xEA, | 0x4B, | 0x48, | 0x15 |
| 3 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0xE4, | 0xFB, | 0xF1, | 0x1B, | 0x04, | 0x0E |
| 4 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0xDC, | 0x58, | 0xEE, | 0x23, | 0xA7, | 0x11 |
| 5 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0xBD, | 0xA9, | 0xBF, | 0x42, | 0x56, | 0x40 |
| 6 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x97, | 0x67, | 0x77, | 0x68, | 0x98, | 0x88 |
| 7 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x24, | 0x35, | 0xA5, | 0xDB, | 0xCA, | 0x5A |
| 8 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x57, | 0x64, | 0x51, | 0xA8, | 0x9B, | 0xAE |
| 9 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x28, | 0xF9, | 0x3E, | 0xD7, | 0x06, | 0xC1 |
| 10 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0xCB, | 0xD1, | 0xAD, | 0x34, | 0x2E, | 0x52 |
| 11 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x5E, | 0x1E, | 0x38, | 0xA1, | 0xE1, | 0xC7 |
| 12 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x19, | 0x98, | 0xF9, | 0xE6, | 0x67, | 0x06 |
| 13 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x84, | 0xEC, | 0x20, | 0x7B, | 0x13, | 0xDF |
| 14 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x13, | 0xA4, | 0xED, | 0xEC, | 0x5B, | 0x12 |
| 15 | 0x9A, | 0x4A, | 0x26, | 0x65, | 0xB5, | 0xD9, | 0x3F, | 0x8A, | 0xBE, | 0xC0, | 0x75, | 0x41 |

<Replaces current Table 119-1>

# Comment #3 Update - 119.2.4.4 Paragraph 5+ (New Content)

The alignment marker mapping function operates on a group of 16 alignment markers. Let am_x<119:0> be the alignment marker for PCS lane x, x=0 to 15, where bit 0 is the first bit transmitted. The alignment markers shall be mapped to am_mapped<1919:0> in a manner that yields the same result as the following process.

For x=0 to 15, am_x<119:0> is constructed as follows.

a) am_x<23:0> is set to $CM_0$, $CM_1$, and $CM_2$, as shown in Figure 119-4 (bits 23:0) using the values in Table 119-1 for PCS lane number x.

b) am_x<31:24>={PRBS9<2*x+97:2*x+96>, PRBS9<6*x+5:6*x>}, as shown in Figure 119-4 (bits 31:24) is an 8-bit pad value of PRBS9 pattern bits, where bit 6*x is the first PRBS9 bit output of the 8-bit pad.

c) am_x<55:32> is set to $CM_3$, $CM_4$, and $CM_5$, as shown in Figure 119-4 (bits 55:32) using the values in Table 119-1 for PCS lane number x.

d) am_x<63:56>={PRBS9<4*x+195:4*x+192>, PRBS9<4*x+131:4*x+128>}, as shown in Figure 119-4 (bits 63:56) is an 8-bit pad value of PRBS9 pattern bits, where bit 4*x+128 is the first PRBS9 bit output of the 8-bit pad.

e) am_x<87:64> is set to $UM_0$, $UM_1$, and $UM_2$, as shown in Figure 119-4 (bits 87:64) using the values in Table 119-1 for PCS lane number x.

f) am_x<95:88>={PRBS9<6*x+293:6*x+288>, PRBS9<2*x+257:2*x+256>}, as shown in Figure 119-4 (bits 95:88) is an 8-bit pad value of PRBS9 pattern bits, where bit 2*x+256 is the first PRBS9 bit output of the 8-bit pad.

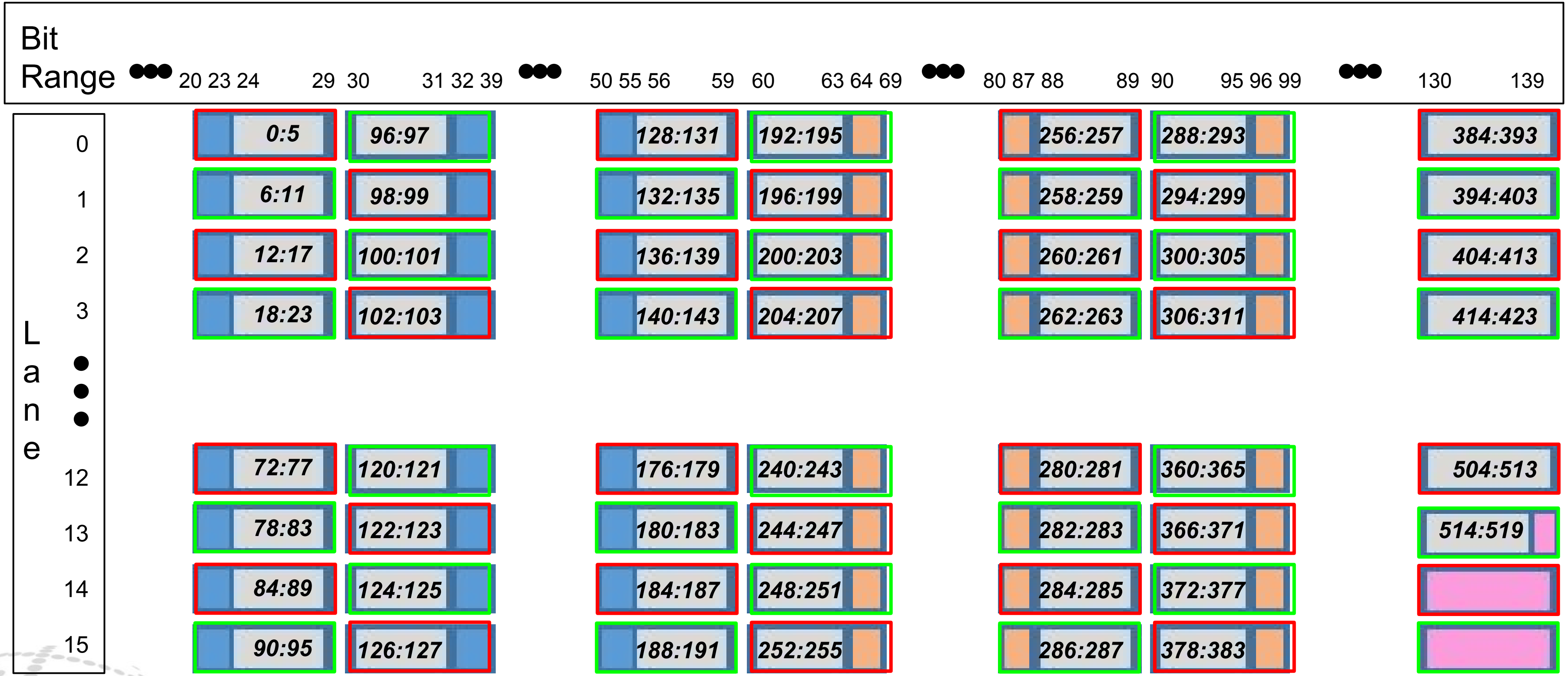
g) am_x<119:96> is set to $UM_3$, $UM_4$, and $UM_5$, as shown in Figure 119-4 (bits 119:96) using the values in Table 119-1 for PCS lane number x.

# Comment #3 Update - 119.2.4.4 Paragraph 5+ (New Content)

As an example, the ~~:is sent as (left most bit sent first) lane marker for 400GBASE-R lane number 0~~ variable am_0 is constructed as follows.

~~10000011 00010110 10000100 00101111 01111100 01111001 01111011 11010000 TBD~~
am_0 = {0xD81461, {PRBS9(293:288), PRBS9(257:256)}, 0x27EB9E, {PRBS9(195:192), PRBS9(131:128)}, 0xD9B565, {PRBS9(97:96), PRBS9(5:0)}, 0x264A9A}

# PRBS9 Padding Organization




*x:y* = PRBS9 bit range for the current Alignment Marker insertion operation (bit 0 generated first).

10-bit Message symbol of FEC codeword A.

10-bit Message symbol of FEC codeword B.

# Comment #3 Update - 119.2.4.4 Paragraph 5+ (New Content)

The variable am_mapped is then derived from 10-bit interleaving the group of 16 alignment markers am_x per the following procedure.

```
For all k=0 to 11
        For all j=0 to 7
                if even(k)
                        am_mapped<160*k+20*j+9:160*k+20*j> = am_{2*j}<10*k+9:10*k>
                        am_mapped<160*k+20*j+19:160*k+20*j+10> = am_{2*j+1}<10*k+9:10*k>
                else
                        am_mapped<160*k+20*j+9:160*k+20*j> = am_{2*j+1}<10*k+9:10*k>
                        am_mapped<160*k+20*j+19:160*k+20*j+10> = am_{2*j}<10*k+9:10*k>
```

The additional 136-bit pad is appended to variable am_mapped as follows.

am_mapped<2055:1920> = PRBS9<519:384>

In this expression, PRBS9<384> is the first PRBS9 bit output of the 136-bit pad.

# Comment #3 Update - 119.2.4.4 Paragraph 5+ (New Content)

The group of alignment markers shall be inserted so they appear every 163 840 257-bit blocks. The variable tx_scrambled_am<10279:0> is constructed in one of two ways. Let the set of vectors tx_scrambled_i<256:0> represent consectutive values of tx_scrambled<256:0>. For a block with alignment markers inserted:

tx_scrambled_am<2055:0> = am_mapped<2055:0>
for all i=0 to 31
    tx_scrambled_am<257*i+2312:257*i+2056> = tx_scrambled_i<256:0>

For a block without alignment markers:

for all i=0 to 39
tx_scrambled_am<257*i+256:257*i> = tx_scrambled_i<256:0>

Alignment marker mapping and repetition rate are shown in Figure 119–5 and Figure 119–6.

# Comment #2 Update - 119.2.4.5

## 119.2.4.5 Pre-FEC Distribution

Two Reed-Solomon FEC codewords are interleaved before data is distributed to the PCS lanes to improve error correction capability. Data is distributed to two 5140-bit message blocks ($m_A$ and $m_B$ are both arrays of 514 10-bit symbols) by performing a 10-bit round robin distribution of the tx_scrambled_am~~<256:0>~~ data as follows.

~~For all j=0 to 39, tx_temp<10279:0> shall be constructed as follows:~~

~~tx_temp<(257j+256):(257j)> = tx_scrambled_am_j<256:0>~~

For all i=0 to 513~~, mA<513:0> and mB<513:0> shall be constructed as follows:~~

$m_A$<(513-i)> = ~~tx_temp~~ tx_scrambled_am<(20*i+9):(20*i)>

$m_B$<(513-i)> = ~~tx_temp~~ tx_scrambled_am<(20*i+19):(20*i+10)>

# Thanks!