Proposed update to Figure 176-2 to more accurately capture "Symbol Demultiplexing" function (in support of comment #420 against D1.3)

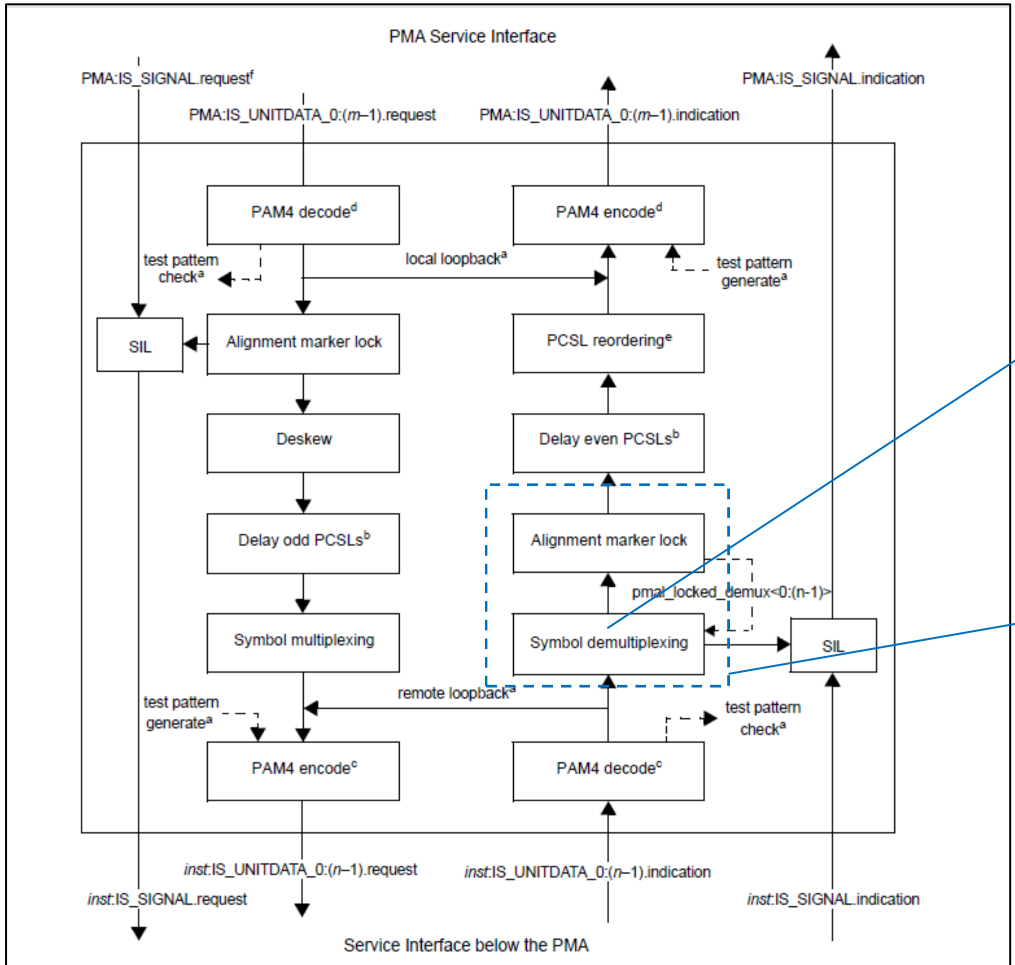
Gary Nicholl - Cisco

IEEE 802.3dj, Phoenix, Jan 2025

Introduction

- Comment #420 points out that in Figure 176-2 the "symbol demultiplexing" functional block is somewhat confusing as this block is really a "blind 20-bit demux and slip" function , and only truly represents a symbol demux when the 20-bit demux aligns with the 20-bit symbol-pair boundaries as confirmed by the subsequent “alignment marker lock" functional block.
- It is the combination of that "blind 20-bit demux and slip" and "alignment marker lock" functions that together performs the "symbol demux" .
- This presentation will propose an update to Figure 176-2 to only show a single "symbol demultipexing" block at this level, and then in the subsequent subclause which defines this function, have a separate diagram showing that the “symbol demultiplexing” block contains two sub blocks; a "blind 20-bit demux and slip“ and an "alignment marker lock".

Current Figure 176-2 (D1.3)



inst: PMA or PMD or FEC or AUI
SIL: Signal indication logic

[a] Optional
[b] If 200GBASE-R, 400GBASE-R or 800GBASE-R
[c] If the sublayer below the PMA is an AUI or PMD
[d] If the sublayer above the PMA is a 1.6TAUI-16
[e] If the sublayer above the PMA is an 800GBASE-R BM-PMA
[f] PMA:IS_SIGNAL.request is not present when the sublayer above the PMA is a PCS or DTE XS

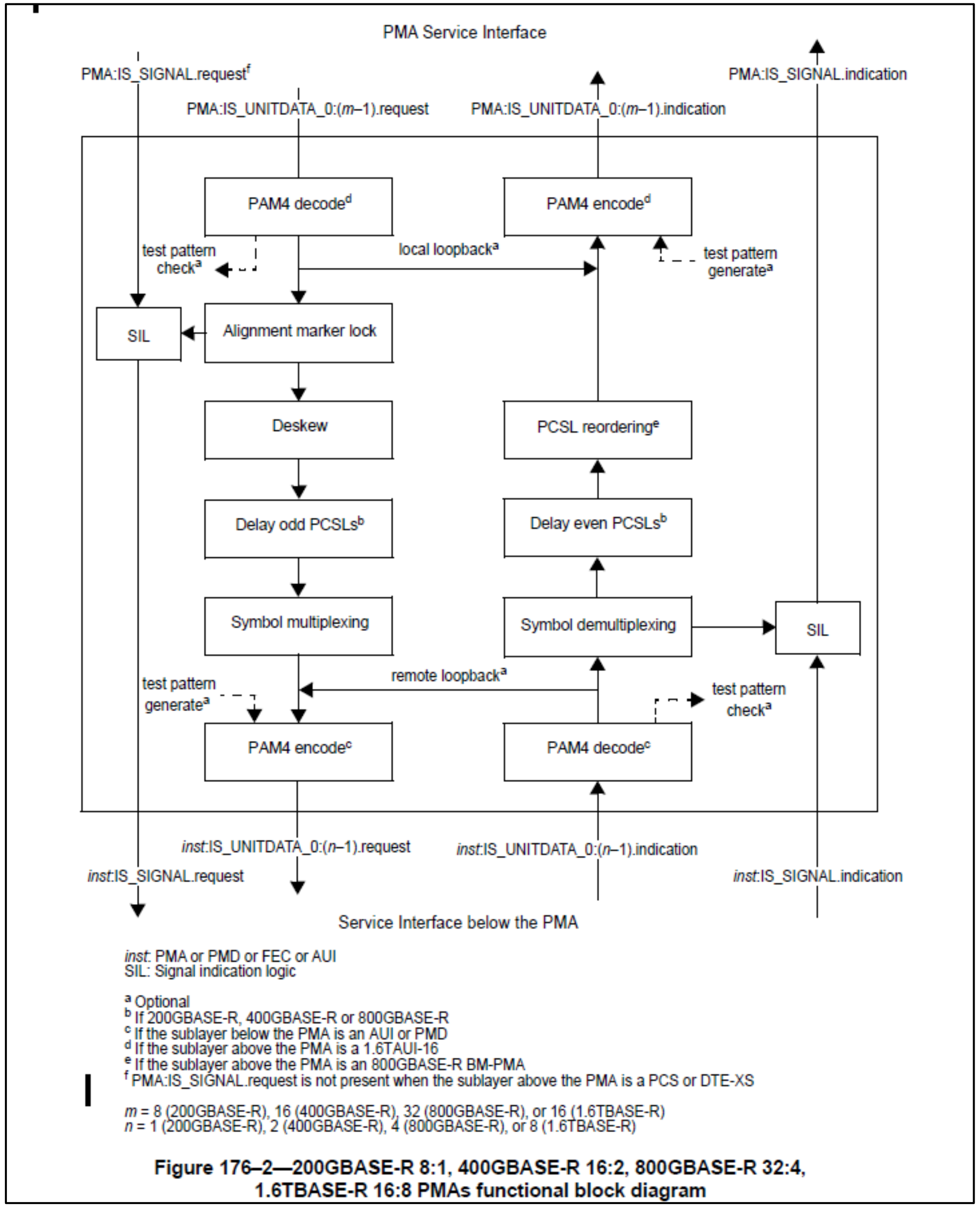
m = 8 (200GBASE-R), 16 (400GBASE-R), 32 (800GBASE-R), or 16 (1.6TBASE-R)
n = 1 (200GBASE-R), 2 (400GBASE-R), 4 (800GBASE-R), or 8 (1.6TBASE-R)

Figure 176–2—200GBASE-R 8:1, 400GBASE-R 16:2, 800GBASE-R 32:4, 1.6TBASE-R 16:8 PMAs functional block diagram

This block by itself does not perform the “symbol demultiplexing” function. This block is really a “blind 20-bit demux and slip” function.

It is the combination of these two blocks and the “pmal_locked_demux feedback signal between them, that performs the “symbol demultiplexing” function

Proposed update to Figure 176-2 (D1.4)



inst: PMA or PMD or FEC or AUI
SIL: Signal indication logic

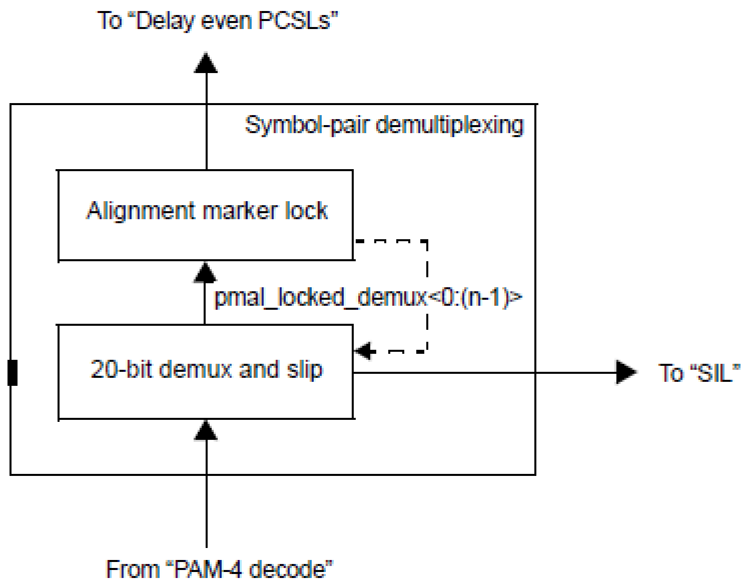
[a] Optional
[b] If 200GBASE-R, 400GBASE-R or 800GBASE-R
[c] If the sublayer below the PMA is an AUI or PMD
[d] If the sublayer above the PMA is a 1.6TAUI-16
[e] If the sublayer above the PMA is an 800GBASE-R BM-PMA
[f] PMA:IS_SIGNAL.request is not present when the sublayer above the PMA is a PCS or DTE-XS

m = 8 (200GBASE-R), 16 (400GBASE-R), 32 (800GBASE-R), or 16 (1.6TBASE-R)
n = 1 (200GBASE-R), 2 (400GBASE-R), 4 (800GBASE-R), or 8 (1.6TBASE-R)

Figure 176–2—200GBASE-R 8:1, 400GBASE-R 16:2, 800GBASE-R 32:4, 1.6TBASE-R 16:8 PMAs functional block diagram

Update to “Symbol demultiplexing” description (176.4.3.2)

- Update the symbol multiplexing description in 176.4.3.2 and 176.4.3.2.1 to indicate that symbol demultiplexing is performed using a “20-bit demux and slip” function in conjunction with an “alignment marker lock” function, which together locate the correct symbol demultiplex boundary and achieve symbol lock on a given input lane.
- One approach would be to keep the description in 176.4.3.2 high level with no implementation details, something like “The symbol demultiplexing function locates the correct symbol demultiplex boundary and achieves symbol lock on a given input lane”. The details of exactly how this is achieved using the “20-bit demux and slip” and “alignment marker lock” functions could then be described in 176.4.3.2.1, and include the block diagram shown on the right.

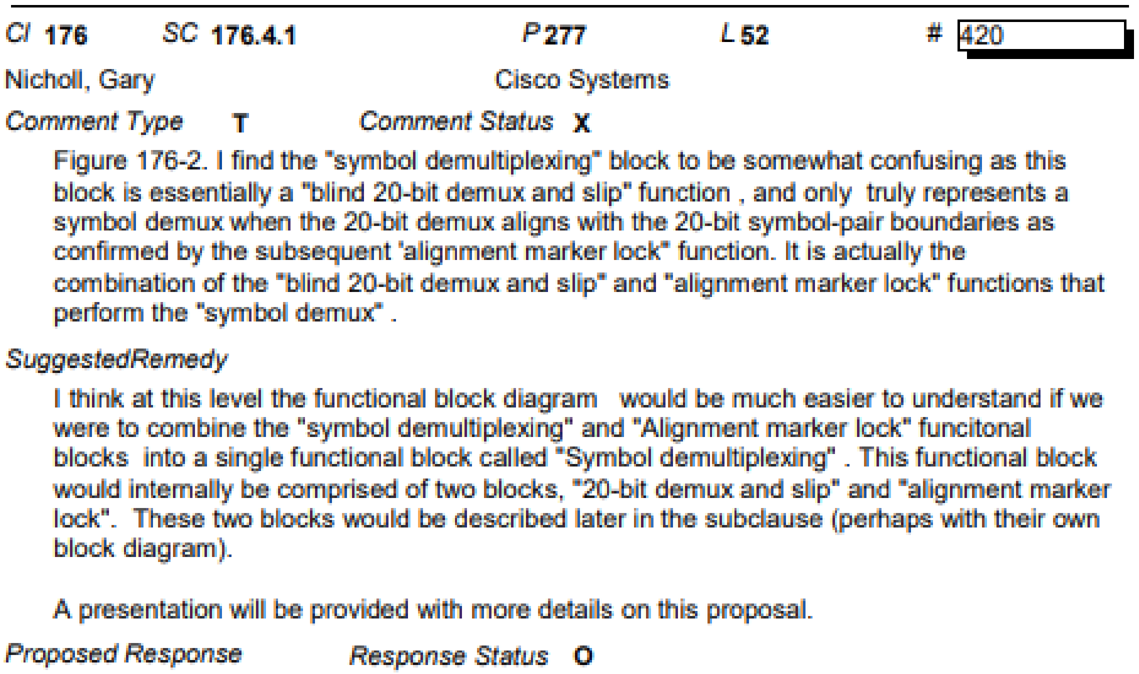


Other changes

- Make similar changes in Figure 176–13

Thanks

Comment #420



Cl **176** *SC* **176.4.1** *P* **277** *L* **52** # 420

Nicholl, Gary Cisco Systems

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

Figure 176-2. I find the "symbol demultiplexing" block to be somewhat confusing as this block is essentially a "blind 20-bit demux and slip" function , and only truly represents a symbol demux when the 20-bit demux aligns with the 20-bit symbol-pair boundaries as confirmed by the subsequent 'alignment marker lock" function. It is actually the combination of the "blind 20-bit demux and slip" and "alignment marker lock" functions that perform the "symbol demux" .

SuggestedRemedy

I think at this level the functional block diagram would be much easier to understand if we were to combine the "symbol demultiplexing" and "Alignment marker lock" funcitonal blocks into a single functional block called "Symbol demultiplexing" . This functional block would internally be comprised of two blocks, "20-bit demux and slip" and "alignment marker lock". These two blocks would be described later in the subclause (perhaps with their own block diagram).

A presentation will be provided with more details on this proposal.

Proposed Response *Response Status* **O**