IEEE P802.11
Wireless LANs

|  |
| --- |
| CR for 36.3.13.5 (Segment Parser) |
| Date: 9/29/2021 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Sigurd Schelstraete | MaxLinear |  |  | sschelstraete@maxlinear.com |
|  |  |  |  |  |

Abstract

This submission proposes resolutions for the following 29 CIDs:

* 4575, 5492, 6826, 7296, 5493, 5494, 6828, 4576, 6827, 7292, 6829, 7303, 7010, 7288, 7290, 7295, 7297, 7298, 7291, 7293, 7289, 7294, 7299, 7300, 7302, 7304, 7305, 7306, 7657

# Introduction

This submission proposes resolutions for the following 29 CIDs:

* 4575, 5492, 6826, 7296, 5493, 5494, 6828, 4576, 6827, 7292, 6829, 7303, 7010, 7288, 7290, 7295, 7297, 7298, 7291, 7293, 7289, 7294, 7299, 7300, 7302, 7304, 7305, 7306, 7657

All CIDs relate to section 36.3.13.5 (Segment parser).

Proposed text changes are relative to 802.11be D1.1.

(Note: D1.2 shows no changes in this section relative to D1.1, other than possible numbering update of the Equations)

# Proposed Resolutions

## CIDs 4575, 5492, 6826, 7296

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Page** | **Comment** | **Proposed Change** | **Resolution** |
| 4575 | 486.63 | In Equation (36-70), the l in summation of m\_l from l=0 to L-1 (l is not l\_0) and l in k'/ml should be differentiated. | Replace l in summation of m\_l from l=0 to L-1 with l'. | REVISEDUse different index notation for summation.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 5492 | 486.63 | l=0 to l=L-1, use a different letter from l. As l represents specfic subblock index | as in comment | REVISEDUse different index notation for summation.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 6826 | 486.63 | equation 36-70, first term on the RHS summation of m(subscript l), l being indicative of current segment should not be used to indicate the summation range | instead of l use i for subscript and summation range | REVISEDUse different index notation for summation.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 7296 | 486.63 | m should be a function of l | change m to m(l) in (36-70) | REJECTEDm implicitly depends on k and l. Formula (36-70) makes the dependency clear.To be consistent with practice elsewhere, it is preferred not to make this change. |

### Discussion

Formula (36-70) in D1.0 looks like this:



In (36-70), *l* is being used both as a summation index and as the value of the frequency subblock index (*l*=0, 1, …, *L*-1).

### Resolution

Correct equation to:

## CID 5493

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5493 | 487.06 | l=0 to l=L-1, use a different letter from l. As l represents specfic subblock index | as in comment | REVISEDUse different index notation for summation.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |



Change text to:

 is bit of a block of bits and

## CID 5494, 6828, 4576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5494 | 487.32 | l=0 to l=L-1, use a different letter from l. As l represents specfic subblock index | as in comment | REVISEDUse different index notation for summation.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 6828 | 487.32 | Equation 36-71, for summation range m(subscript l) is used, l being segment of interest in determining yk,l,u , using l to indicate the summation range is confusing | instead of l use i for subscript and summation range | REVISEDUse different index notation for summation.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 4576 | 487.63 | In equation (36-71), the l in summation of m\_l from l=0 to L-1 (l is not l\_0) and l in k'/ml should be differentiated. | Replace l in summation of m\_l from l=0 to L-1 (l is not l\_0) with l'. | REVISEDUse different index notation for summation.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 NOTE: comment refers to equation (36-71), which is on 487.32. It is assumed the line reference in the comment is incorrect and should be 32 instead of 63. |



Change text to:

## CID 6827, 7292

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6827 | 487.09 | description of mi not provided | use ml/mi instead of ml alone | REVISEDMake changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1  |
| 7292 | 486.01 | Proportional ratio m\_l is mentioned without any definition or explanation. | Introduce Proportional ratio m\_l before use | REVISEDMake changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |



Both notation and are used in the formula above, so better to include both in the list of parameters.

Also, “proportional ratio” is not the correct term to use, sine is actually an absolute number of bits assigned to a given output block for each round of the round-robin parser.

Updated text proposal:



## CID 6829, 7303

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6829 | 487.55 | Figure 36-57 - reference to equations in figure is misplaced | insted of 34-x2 use 36-70 and intead of 34-x3 use 36-71 | REVISEDReplace “Equation (34-x2)” with “Equation (36-70)”.Replace “Equation (34-x3)” with “Equation (36-71)”. |
| 7303 | 487.55 | Figure has text saying "Proportional round robin parser processing based on (34-x2)" and "Leftover bits processing based on (34-x3)". This text is out of date. | Correct text in Figure 36-57 | REVISEDReplace “Equation (34-x2)” with “Equation (36-70)”.Replace “Equation (34-x3)” with “Equation (36-71)”. |



## CID 7010, 7301

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7010 | 487.26 | Replace 'For the MRU with the number...is not equal to 0' with 'For an MRU with number...not equal to'. | Replace 'For the MRU with the number...is not equal to 0' with 'For an MRU with number...not equal to'. | REVISEDAgree in principle with the change. Change has been integrated with further changes in full text proposal.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 7301 | 487.27 | Change "continue process" to "continue to process" | See comment | ACCEPTED |

## CID 7288

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7288 | 485.31 | The segment parsing depends on the size of the RU/MRU, not the bandwidth. Change wording of first paragraph to be similar to first paragraph of 36.3.13.9 (Segment deparser). | Reword first paragraph as "Segment parsing shall be performed for RU or MRU of size 2×996-, 996+484-, 996+484+242-, 2x996+484-, 3x996-, 3x996+484-, or 4x996-tone. For a 26-, 52-, 52+26-, 106-, 106+26-, 242-, 484-, 484+242-,and 996-tone RU or MRU, segment parsing is bypassed." | REVISEDAgree in principle.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |

Original text:



Changed to:



## CID 7290, 7295, 7297, 7298

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7290 | 485.52 | cbpss,l,u should use uppercase to be consistent with other notations | Change N\_cbpss,l,u to N\_CBPSS\_l,u | REVISEDReplace N\_cbpss,l,u with N\_CBPSS\_l,u in all appropriate places.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 7295 | 486.53 | bpscs,u should be uppercase | Change N\_bpscs,u to N\_BPSCS\_u | REVISEDReplace N\_bpscs,u with N\_BPSCS\_u in all appropriate places.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 7297 | 487.02 | cbpss,l,u should use uppercase to be consistent with other notations | Change N\_cbpss,l,u to N\_CBPSS\_l,u (several instances) | REVISEDReplace N\_cbpss,l,u with N\_CBPSS\_l,u in all appropriate places.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 7298 | 487.02 | bpscs,l,u should use uppercase to be consistent with other notations | Change N\_bpscs,l,u to N\_BPSCS\_l,u | REVISEDNote that N\_bpscs,l,u does not depend on the frequency subblock index. Sub-index “l” should be removed as well.Change N\_bpscs,l,u to N\_BPSCS,u in all appropriate places.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |

## CID 7291

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7291 | 485.45 | Definition of y\_k,l,u: add "for user u". | Change "is bit k of the frequency subblock l." to "is bit k of the frequency subblock l for user u." | REVISEDAgreed in principle, change also needs to be made on page 487.20.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |

Current text:



## CID 7293

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7293 | 486.09 | Where do we use "Nsd,total" (fourth column of Table 36-48)? | Clarify. Remove if not needed. | REVISEDNsd,total is not used or needed in this section. Remove this column from the Table.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |

Current text:



## CID 7289, 7294

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7289 | 485.52 | Where is "L" defined? Should be dome before its first use. | Define "L" as the number of segments | REVISEDL values are added to Table 36-48.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 7294 | 486.06 | Add "L" and "N\_CBPSS,l,u to Table 36-48. | See comment | REVISEDN\_CBPSS,l,u need to be provided for the various RU/MRU cases for which segment parsing is needed. A separate table with these values is provided.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1Add row to Table 36-23 defining N\_CBPSS,l,u, as shown in IEEE 802.11-21/1614r1 under CID 7294. |

### Discussion

Although operation of the segment parsing immediately assumes knowledge of , no definition is ever given. represents the number of coded bits per symbol and per spatial stream for frequency subblock *l*. It can be calculated from the number of data subcarriers in subblock *l* and the number of bits per subcarrier. For the RU and MRU for which segment parsing is needed, the values of are shown in the table below.

For completeness, we also propose to add a definition of to Table 36-23.

Editor’s instruction: insert row in Table 36-23 after existing row :

|  |  |
| --- | --- |
|  |  |
|  | Number of coded bits per OFDM symbol per spatial stream for frequency subblock *l* and user *u*,, with the number of frequency subblocks |
|  |  |

## *Table 36-xxxx Values of*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RU/MRU** | **RU order (low to high frequency)** | **L** | **Is DCM****used?** | ***NCBPSS,0,u*** | ***NCBPSS,1,u*** | ***NCBPSS,2,u*** | ***NCBPSS,3,u*** |
| 996+484 | 484+996 | 2 | No | 468 ×  | 980 ×  |  |
| Yes | 234  | 490  |
| 996+484 | No | 980 ×  | 468 ×  |
| Yes | 490  | 234  |
| 996+484+242 | (242+484)+996 | No | 702 ×  | 980 ×  |
| Yes | 351 | 490 |
| 996+(242+484) | No | 980 ×  | 702 ×  |
| Yes | 490 | 351 |
| 2996+484 | 484+996+996 | 3 | No | 468 ×  | 980 ×  |  980 ×  |  |
| 996+484+996 | No | 980 ×  | 468 ×  | 980 ×  |
| 996+996+484 | No | 980 ×  | 980 ×  | 468 ×  |
| 3996+484 | 484+996+996+996 | 4 | No | 468 ×  | 980 ×  | 980 ×  | 980 ×  |
| 996+484+996+996 | No | 980 ×  | 468 ×  | 980 ×  | 980 ×  |
| 996+996+484+996 | No | 980 ×  | 980 ×  | 468 ×  | 980 ×  |
| 996+996+996+484 | No | 980 ×  | 980 ×  | 980 ×  | 468 ×  |
| 2996 | 996+996 | 2 | No | 980 ×  | 980 ×  |  |
| Yes | 490 | 490 |
| 3996 | 996+996+996 | 3 | No | 980 ×  | 980 ×  | 980 ×  |  |
| Yes | 490 | 490 | 490 |
| 4996 | 996+996+996+996 | 4 | No | 980 ×  | 980 ×  | 980 ×  | 980 ×  |
| Yes | 490 | 490 | 490 | 490 |

## CID 7299

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7299 | 487.19 | How are the values of n\_l determined? The current definition is rather vague. Should we add them to Table 36-48? | See comment | REJECTEDn\_l is defined p 487.19 of D1.0 |

## CID 7300, 7302

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7300 | 487.26 | The concept of leftover bits is not clearly explained. Provide a better explanation/definition. | See comment | REVISEDAdd some background on the concept of leftover bits.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |
| 7302 | 487.44 | "the subblock without leftover bits". This implies there is only one. Add this to the explanation of leftover bits. | See comment | REVISEDAdd some background on the concept of leftover bits.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |

Existing text:



Proposed change:



## CID 7304, 7305

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7304 | 487.06 | The output arrows of Figure 36-58 should connect to the segment deparser | Add segment deparser block in Figure 36-58 | REVISEDIt’s correct that LDPC tone mapper output should connect to segment deparser. Since we already include Stream parser, segment parser, constellation mapper and LDPC tone mapper in Figure 36-58, adding segment mapper would complete the flow and clearly show where per-subblock processing begins and ends.Update figure with additional block on the right representing segment deparser. LDPC Tone mapper arrows point towards this block. |
| 7305 | 487.31 | The output arrows of Figure 36-59 should connect to the segment deparser | Add segment deparser block in Figure 36-59 | REVISEDIt’s correct that LDPC tone mapper output should connect to segment deparser. Since we already include Stream parser, segment parser, constellation mapper and LDPC tone mapper in Figure 36-58, adding segment mapper would complete the flow and clearly show where per-subblock processing begins and ends.Update figure with additional block on the right representing segment deparser. LDPC Tone mapper arrows point towards this block. |

Current Figure:



## CID 7306

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7306 | 487.31 | Why is RU242+484 split into two RUs RU242 and RU484 after LDPC Tone mapping in Figure 36-59? | See comment | REVISEDTHE LDPC Tone mapper works on the full MRU and output a 242+484 MRU. The input to the segment deparser also is the 242+484 MRU. There appears to be no need to split the output of the LDPC tone mapper into a 242 tone RU and a 484 tone RU.Correct Figure 36-59: have single arrow leave “ LDPC tone mapper”. Remove split afterwards (for the 242+484 MRU) |

Current Figure:



## CID 7657

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7657 | 486.06 | What will be RU/MRU size for MCS14 for segment parser? It should be 484 for 80 MHz, 996 for 160 MHz and 2x996 for 320 MHz. Thus, only in case of 320 MHz MCS14 will apply segment parser and segment deparser. Add clarification in segment parser and segment deparser sections. | See comment. | REVISEDAdd NOTE to clarify that RU size refers to RU size prior to duplication when MCS 14 is used.Make changes as shown in section “Text Proposal” of IEEE 802.11-21/1614r1 |

Proposed text:



# Text proposal

**36.3.13.5 Segment parser**

Segment parsing shall be performed for RU or MRU of size 2×996-, 996+484-, 996+484+242-, 2x996+484-, 3x996, 3x996+484-, or 4x996-tone. For a 26-, 52-, 52+26-, 106-, 106+26-, 242-, 484-, 484+242- and 996-tone RU or MRU, the segment parser is bypassed and the output bits are as specified in Equation (36-69).

 (36-69)

where

*xk,u* is bit *k* of a block of bits, to

*l* is the frequency subblock index

*yk,l,u* is bit *k* of the frequency subblock *l* and user *u.*

*u* is the user index, *u*= 0, 1, …,

NOTE: for MCS 14, the RU size refers to the RU size before duplication. Specifically, this means that segment parsing with MCS 14 is only required when using 320 MHz.

For a 160 MHz and 320 MHz transmission with a 2×996-, 996+484-, 996+484+242-, 2×996+484-, 3×996-, 3×996+484-, or 4×996-tone RU/MRU, the output bits of each stream parser are ~~first divided into~~ provided in blocks of bits. The segment parser further divides each block into *L* blocks of bits respectively, for , such that . *L* is the number of frequency subblocks in the RU or MRU. The values for are given in Table 36-xxxx for the various RU and MRU cases.

## Table 36-xxxx Values of

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RU/MRU** | **RU order (low to high frequency)** | **L** | **Is DCM****used?** | ***NCBPSS,0,u*** | ***NCBPSS,1,u*** | ***NCBPSS,2,u*** | ***NCBPSS,3,u*** |
| 996+484 | 484+996 | 2 | No | 468 ×  | 980 ×  |  |
| Yes | 234  | 490  |
| 996+484 | No | 980 ×  | 468 ×  |
| Yes | 490  | 234  |
| 996+484+242 | (242+484)+996 | No | 702 ×  | 980 ×  |
| Yes | 351 | 490 |
| 996+(242+484) | No | 980 ×  | 702 ×  |
| Yes | 490 | 351 |
| 2996+484 | 484+996+996 | 3 | No | 468 ×  | 980 ×  |  980 ×  |  |
| 996+484+996 | No | 980 ×  | 468 ×  | 980 ×  |
| 996+996+484 | No | 980 ×  | 980 ×  | 468 ×  |
| 3996+484 | 484+996+996+996 | 4 | No | 468 ×  | 980 ×  | 980 ×  | 980 ×  |
| 996+484+996+996 | No | 980 ×  | 468 ×  | 980 ×  | 980 ×  |
| 996+996+484+996 | No | 980 ×  | 980 ×  | 468 ×  | 980 ×  |
| 996+996+996+484 | No | 980 ×  | 980 ×  | 980 ×  | 468 ×  |
| 2996 | 996+996 | 2 | No | 980 ×  | 980 ×  |  |
| Yes | 490 | 490 |
| 3996 | 996+996+996 | 3 | No | 980 ×  | 980 ×  | 980 ×  |  |
| Yes | 490 | 490 | 490 |
| 4996 | 996+996+996+996 | 4 | No | 980 ×  | 980 ×  | 980 ×  | 980 ×  |
| Yes | 490 | 490 | 490 | 490 |

The segment parser bit distribution sequence starts from the lowest frequency location to the highest frequency.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |
|  |  |  |  |  |  |
|  |  |
|  |  |  |  |  |  |
|  |  |

The bits in each block of bits are determined by the segment parser as shown in Equation (36-70)(#2672).

 (36-70)

where

*k* = 0, 1, …, when DCM is not used and

 *k* = 0, 1, …, when DCM is used

 is bit *m* of a block of bits and *m* = 0,1, …,

, are the number of bits assigned to a block of output bits for each round of the round-robin parser. Values are given in in Table 36-48. The values have proportional ratios to the number of occupied data subcarriers in each 80 MHz frequency subblock.

*l* is the frequency subblock index,

*L* is the number of frequency subblocks. for 996+484-, 996+484+242-, 2×996-tone RU/MRU; for 2×996+484- and 3×996-tone MRU; for 3×996+484- and 4×996-tone RU/MRU.

 is bit *k* of frequency subblock (or RU in 80 MHz subblock(#1279)) *l* for user *u*.

 for subblock *l* with nonzero leftover bits, otherwise.

*u* is the user index, *u* = 0, 1, …,

 for frequency subblock (#2952)(#3072).

## Table 36-48 Segment parser parameters

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RU/MRU** | **RU order****(low to high frequency)** | **L** | **Is DCM****used?** | ***m*0** | ***m1*** | ***m2*** | ***m3*** | **Leftover bits per frequency subblock** |
| 996+484 | 484+996 | 2 | No | *s* | *2s* |  | 44  *NBPSCS,u* |
| Yes | 22 |
| 996+484 | No | *2s* | *s* | 44  *NBPSCS,u* |
| Yes | 22 |
| 996+484+242 | (242+484)+996 | No | *3s* | *4s* | 44  *NBPSCS,u* |
| Yes | 22 |
| 996+(242+484) | No | *4s* | *3s* | 44  *NBPSCS,u* |
| Yes | 22 |
| 2996+484 | 484+996+996 | 3 | No | *s* | *2s* | *2s* |  | 44  *NBPSCS,u* |
| 996+484+996 | No | *2s* | *s* | *2s* | 44  *NBPSCS,u* |
| 996+996+484 | No | *2s* | *2s* | *s* | 44  *NBPSCS,u* |
| 3996+484 | 484+996+996+996 | 4 | No | *s* | *2s* | *2s* | *2s* | 44  *NBPSCS,u* |
| 996+484+996+996 | No | *2s* | *s* | *2s* | *2s* | 44  *NBPSCS,u* |
| 996+996+484+996 | No | *2s* | *2s* | *s* | *2s* | 44  *NBPSCS,u* |
| 996+996+996+484 | No | *2s* | *2s* | *2s* | *s* | 44  *NBPSCS,u* |
| 2996 | 996+996 | 2 | No | *s* | *s* |  | 0 |
| Yes |
| 3996 | 996+996+996 | 3 | No | *s* | *s* | *s* |  | 0 |
| Yes |
| 4996 | 996+996+996+996 | 4 | No | *s* | *s* | *s* | *s* | 0 |
| Yes |

In Table 36-48 (Segment parser parameters), .

If the RU or MRU contains a frequency subblock that is not fully occupied (i.e., the frequency subblock consist of 484 or 242+484 occupied tones), that frequency subblock will reach its full value before the other frequency subblocks. At that point, no further bits are output by the segment parser for that subblock. For the other frequency subblocks, the number of leftover bits as defined in Table 36-48 (Proportional round robin segment parser parameters(#1411)) is not equal to 0, and proportional round robin parser will continue to process the leftover bits as shown in Equation (36-71)(#2443).

 (36-71)

where

*k* = , …, when DCM is not used and

 *k* = , …, when DCM is used.

*k’* = when DCM is not used and

 *k’* = when DCM is used.

*l0* is the subblock index with (i.e., the subblock without leftover bits).

Illustration of the proportional round robin parser with leftover bits processing is shown in Figure 36-57 (Illustration of the proportional round robin parser with leftover bits processing).



**Figure 36-57—Illustration of the proportional round robin parser with leftover bits processing**

Editor’s note:

Replace “Equation (34-x2)” in Figure 36-57 with “Equation (36-70)”.

Replace “Equation (34-x3)” in Figure 36-57 with “Equation (36-71)”.

Illustration of the segment parser for 996+484-tone MRU and 996+484+242-tone MRU are shown in Figure 36-58 (Illustration of the segment parser for 996+484-tone RU) and Figure 36-59 (Illustration of the segment parser for 996+484+242-tone RU), respectively.



**Figure 36-58—Illustration of the segment parser for 996+484-tone RU**

Editor’s note: update Figure 36-58, showing the segment deparser to the right of the LDPC Tone mapper outputs – one per stream.



**Figure 36-59—Illustration of the segment parser for 996+484+242-tone RU**

Editor’s note: update Figure 36-59, showing the segment deparser to the right of the LDPC Tone mapper outputs – one per stream. Also, remove split of LDPC output mapper into RU242 and RU484.