

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|--|
| Claim 1 | |
| [1] A method of encoding a signal, comprising: | <p>To the extent this preamble is construed to be limiting, the Accused Products practice a method of encoding a signal.</p> <p>For example, the Accused Products implement the IEEE Standards, which provide a method of encoding a signal.</p> <p>For example, the IEEE 802.11n-2009 amendment to the IEEE 802.11-2007 standard and the IEEE 802.11-2012 version of the 802.11 standard include “low-density parity check (LDPC) encoding.”</p> <p>An HT STA has PHY features consisting of the modulation and coding scheme (MCS) set described in 20.3.5 and physical layer convergence procedure (PLCP) protocol data unit (PPDU) formats described in 20.1.4. Some PHY features that distinguish an HT STA from a non-HT STA are referred to as multiple input, multiple output (MIMO) operation; spatial multiplexing (SM); spatial mapping (including transmit beamforming); space-time block coding (STBC); low-density parity check (LDPC) encoding; and antenna selection (ASEL). The allowed PPDU formats are non-HT format, HT-mixed format, and HT-greenfield format. The PPDU may be transmitted with 20 MHz or 40 MHz bandwidth.</p> <p>IEEE 802.11n-2009 at § 5.2.9; IEEE 802.11-2012 at § 4.3.10; IEEE 802.11-2020 at § 4.3.13 (emphasis added).</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------|--|--|---------------|----------------------|-------------------------------|---------|---------|----|----|----|----|------------------------|-----------------------------|---------------|---------------|---------------------|---------------------|---------|---------|--|--|-----|-----|-----|-----|-----|-----|----------------------|-----------------------|-------------------------|----------|----------------------|-------------------------------|----------|------------|----------|------------------------|--|--|
| | <p>The HT Capabilities Info field of the HT Capabilities element is 2 octets in length, and contains capability information bits. The structure of this field is defined in Figure 7-95o18.</p> <table><tr><td>B0</td><td>B1</td><td>B2</td><td>B3</td><td>B4</td><td>B5</td><td>B6</td><td>B7</td><td>B8</td><td>B9</td></tr><tr><td>LDPC Coding Capability</td><td>Supported Channel Width Set</td><td>SM Power Save</td><td>HT-Greenfield</td><td>Short GI for 20 MHz</td><td>Short GI for 40 MHz</td><td>Tx STBC</td><td>Rx STBC</td><td></td><td></td></tr></table> <table><tr><td>B10</td><td>B11</td><td>B12</td><td>B13</td><td>B14</td><td>B15</td></tr><tr><td>HT-Delayed Block Ack</td><td>Maximum A-MSDU Length</td><td>DSSS/CCK Mode in 40 MHz</td><td>Reserved</td><td>Forty MHz Intolerant</td><td>L-SIG TXOP Protection Support</td></tr></table> <p style="text-align: center;">Figure 7-95o18—HT Capabilities Info field</p> <p>IEEE 802.11n-2009 at § 7.3.2.56.2; IEEE 802.11-2012 at § 8.4.2.58.2; IEEE 802.11-2020 at § 9.4.2.55 (emphasis added).</p> <p style="text-align: center;">Table 7-43j—Subfields of the HT Capabilities Info field</p> <table><tr><th>Subfield</th><th>Definition</th><th>Encoding</th></tr><tr><td>LDPC Coding Capability</td><td>Indicates support for receiving LDPC coded packets</td><td>Set to 0 if not supported Set to 1 if supported</td></tr></table> <p>IEEE 802.11n-2009 at Table 7-43j; <i>see also</i> IEEE 802.11-2012 at Table 8-124; IEEE 802.11-2020 at Table 9-184.</p> <p>A STA shall not transmit a control response frame with TXVECTOR parameter FEC_CODING set to LDPC_CODING unless it is in response to a reception of a frame with the RXVECTOR parameter FEC_CODING set to LDPC_CODING.</p> | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | LDPC Coding Capability | Supported Channel Width Set | SM Power Save | HT-Greenfield | Short GI for 20 MHz | Short GI for 40 MHz | Tx STBC | Rx STBC | | | B10 | B11 | B12 | B13 | B14 | B15 | HT-Delayed Block Ack | Maximum A-MSDU Length | DSSS/CCK Mode in 40 MHz | Reserved | Forty MHz Intolerant | L-SIG TXOP Protection Support | Subfield | Definition | Encoding | LDPC Coding Capability | Indicates support for receiving LDPC coded packets | Set to 0 if not supported Set to 1 if supported |
| B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDPC Coding Capability | Supported Channel Width Set | SM Power Save | HT-Greenfield | Short GI for 20 MHz | Short GI for 40 MHz | Tx STBC | Rx STBC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| B10 | B11 | B12 | B13 | B14 | B15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HT-Delayed Block Ack | Maximum A-MSDU Length | DSSS/CCK Mode in 40 MHz | Reserved | Forty MHz Intolerant | L-SIG TXOP Protection Support | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Subfield | Definition | Encoding | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDPC Coding Capability | Indicates support for receiving LDPC coded packets | Set to 0 if not supported Set to 1 if supported | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | |
|----------------|--|------------|-----------|-------|----------|----------|------------|--|--|--|--|--|--|
| | <p>IEEE 802.11n-2009 at § 9.6.0e.5.5; IEEE 802.11-2012 at § 9.7.6.5.5; IEEE 802.11-2020 at § 10.6.6.5.7 (emphasis added).</p> <p>A STA shall not transmit a control frame that initiates a TXOP with the TXVECTOR parameter FEC_CODING set to a value of LDPC_CODING.</p> <p>IEEE 802.11n-2009 at § 9.6.0e.7; IEEE 802.11-2012 at § 9.7.6.7; IEEE 802.11-2020 at § 10.6.6.7 (emphasis added)</p> <p>LDPC coding was incorporated into the IEEE 802.11 standard via the 802.11n-2009 amendment. In general, the following sections of 802.11n discuss LDPC coding: § 9.7f, § 20.3.11.6, Annex G at sections G.2 and G.3 and Annex R.</p> <p>9.7f LDPC operation</p> <p>An HT STA shall not transmit a frame with the TXVECTOR parameter FORMAT set to HT_MF or HT_GF and the TXVECTOR parameter FEC_CODING set to LDPC_CODING unless the RA of the frame corresponds to a STA for which the LDPC Coding Capability subfield of the most recently received HT Capabilities element from that STA contained a value of 1 and the MIB variable dot11LDPCCodingOption-Enabled is set to TRUE.</p> <p>Further restrictions on TXVECTOR parameter values may apply due to rules found in 9.13 and 9.6.</p> <p>IEEE 802.11n-2009 at § 9.7f; <i>see also</i> IEEE 802.11-2012 at § 9.14; IEEE 802.11-2020 at § 10.15.</p> <p>Table 20-1—TXVECTOR and RXVECTOR parameters</p> <table><tr><th rowspan="2">Parameter</th><th rowspan="2">Condition</th><th rowspan="2">Value</th><th>TXVECTOR</th><th>RXVECTOR</th></tr><tr><th colspan="2">See NOTE 1</th></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> | Parameter | Condition | Value | TXVECTOR | RXVECTOR | See NOTE 1 | | | | | | |
| Parameter | Condition | | | | Value | TXVECTOR | RXVECTOR | | | | | | |
| | | See NOTE 1 | | | | | | | | | | | |
| | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | |
|----------------|--|-------------|--|--|---|---|-----------|-------------|---|---|
| | <div>***</div> <table><tr><td rowspan="2">FEC_CODING</td><td>FORMAT is HT_MF or HT_GF</td><td>Indicates which FEC encoding is used. Enumerated type: BCC_CODING indicates binary convolutional code. LDPC_CODING indicates low-density parity check code.</td><td>Y</td><td>Y</td></tr><tr><td>Otherwise</td><td>Not present</td><td>N</td><td>N</td></tr></table> <p>IEEE 802.11n-2009 at Table 20-1; IEEE 802.11-2012 at Table 20-1; IEEE 802.11-2020 at Table 19-1 (emphasis added).</p> <p>HT-mixed format and HT-greenfield format transmissions can be generated using a transmitter consisting of the following blocks:</p> <ul style="list-style-type: none">a) <i>Scrambler</i> scrambles the data to reduce the probability of long sequences of 0s or 1s; see 20.3.11.3.b) <i>Encoder parser</i>, if BCC encoding is to be used, demultiplexes the scrambled bits among N_{ES} (number of BCC encoders for the Data field) BCC encoders, in a round robin manner.c) <i>FEC encoders</i> encode the data to enable error correction. An FEC encoder may include a binary convolutional encoder followed by a puncturing device, or it may include an LDPC encoder. <p>IEEE 802.11n-2009 at § 20.3.3 (emphasis added); see also IEEE 802.11-2012 at § 20.3.3; IEEE 802.11-2020 at § 19.3.3</p> | FEC_CODING | FORMAT is HT_MF or HT_GF | Indicates which FEC encoding is used. Enumerated type: BCC_CODING indicates binary convolutional code. LDPC_CODING indicates low-density parity check code. | Y | Y | Otherwise | Not present | N | N |
| FEC_CODING | FORMAT is HT_MF or HT_GF | | Indicates which FEC encoding is used. Enumerated type: BCC_CODING indicates binary convolutional code. LDPC_CODING indicates low-density parity check code. | Y | Y | | | | | |
| | Otherwise | Not present | N | N | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | <p>NOTES</p> <ul style="list-style-type: none">—There may be 1 or 2 FEC encoders when BCC encoding is used.—The stream parser may have 1, 2, 3 or 4 outputs.—When LDPC encoding is used, the interleavers are not used—When STBC is used, the STBC block has more outputs than inputs.—When spatial mapping is used, there may be more transmit chains than space time streams.—The number of inputs to the spatial mapper may be 1, 2, 3, or 4. <p>IEEE 802.11n-2009 at Fig. 20-3; IEEE 802.11-2012 at Fig. 20-3; IEEE 802.11-2020 at Fig. 19-3 (emphasis added)</p> <p>20.3.4 Overview of the PPDU encoding process</p> <p>The encoding process is composed of the steps described below. The following overview is intended to facilitate an understanding of the details of the convergence procedure:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p style="text-align: center;">* * *</p> <p>e) Append the PSDU to the SERVICE field (see 20.3.11.1). If BCC encoding is to be used, as indicated by the FEC_CODING parameter of the TXVECTOR, tail bits are appended to the PSDU. If a single BCC encoder is used (i.e., when the value of N_{ES} is 1), the bit string is extended by 6 zero bits. If two BCC encoders are used (i.e., when the value of N_{ES} is 2), the bit string is extended by 12 zero bits. The number of symbols, N_{SYM}, is calculated according to Equation (20-32), and if necessary, the bit string is further extended with zero bits so that the resulting length is a multiple of $N_{SYM} \times N_{DBPS}$, as described in 20.3.11. If LDPC encoding is to be used, as indicated by the FEC_CODING parameter of the TXVECTOR, the resulting bit string is padded, if needed, by repeating coded bits rather than using zero bits, as given in the encoding procedure of 20.3.11.6.5. The number of resulting symbols is given by Equation (20-41), and the number of repeated coded bits used for padding is given by Equation (20-42). The resulting bit string constitutes the DATA part of the packet.</p> <p style="text-align: center;">* * *</p> <p>i) If BCC encoding is to be used, encode the extended, scrambled data string with a rate 1/2 convolutional encoder (see 17.3.5.5). Omit (puncture) some of the encoder output string (chosen according to puncturing pattern) to reach the desired coding rate, R. Refer to 20.3.11.5 for details. If LDPC encoding is to be used, encode the scrambled data stream according to 20.3.11.6.5.</p> <p>j) Parse the coded bit stream that results from the BCC encoding or LDPC encoding into N_{SS} spatial streams, where the value of N_{SS} is determined from the MCS parameter of the TXVECTOR. See 20.3.11.7.2 for details.</p> <p>IEEE 802.11n-2009 at § 20.3.4 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.4; IEEE 802.11-2020 at § 19.3.4.</p> <p>20.3.9.4.3 HT-SIG definition</p> <p>The HT-SIG is used to carry information required to interpret the HT packet formats. The fields of the HT-SIG are described in Table 20-10.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | |
|----------------|---|---|----------------|------------------------|-------|--|--|------------|---|---|
| | <p align="center">Table 20-10—HT-SIG fields</p> <table border="1"><tr><th>Field</th><th>Number of bits</th><th>Explanation and coding</th></tr><tr><td align="center" colspan="3">* * *</td></tr><tr><td>FEC coding</td><td align="center">1</td><td>Set to 1 for LDPC. Set to 0 for BCC.</td></tr></table> <p>IEEE 802.11n-2009 at § 20.3.9.4.3, Table 20-10; <i>see also</i> IEEE 802.11-2012 at Table 20-11; IEEE 802.11-2020 at Table 19-11.</p> <p>20.3.11 Data field</p> <p>When BCC encoding is used, the Data field consists of the 16-bit SERVICE field, the PSDU, either six or twelve tail bits, depending on whether one or two encoding streams are represented, and pad bits. When LDPC encoding is used, the Data field consists of the 16-bit SERVICE field and the PSDU, processed by the procedure in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.1; IEEE 802.11-2020 at § 19.3.11.1.</p> <p>20.3.11.3 Coding</p> <p>The Data field shall be encoded using either the BCC defined in 17.3.5.5 or the LDPC code defined in 20.3.11.6. The encoder is selected by the FEC coding field in the HT-SIG, as described in 20.3.9.4.3. A single FEC encoder is always used when LDPC coding is used. When the BCC FEC encoder is used, a single encoder is used, except that two encoders shall be used when the selected MCS has a PHY rate greater than 300 Mb/s. To determine whether to use one or two BCC FEC encoders, the rate is calculated based on the use of an 800 ns GI. The operation of the BCC FEC is described in 20.3.11.5. The operation of the LDPC coder is described in 20.3.11.6.</p> <p>IEEE 802.11n-2009 at § 20.3.11.3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.4; IEEE 802.11-2020 at § 19.3.11.4.</p> | Field | Number of bits | Explanation and coding | * * * | | | FEC coding | 1 | Set to 1 for LDPC. Set to 0 for BCC. |
| Field | Number of bits | Explanation and coding | | | | | | | | |
| * * * | | | | | | | | | | |
| FEC coding | 1 | Set to 1 for LDPC. Set to 0 for BCC. | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------------------|--|-----------------------|---------------------|-----------------------|-------|--|--|-----------------|--|--|-------|--|--|----------------------------------|---------------|--------|--------------------------------|---------------|---------|-----------|---------------------|-------|-------|--|--|
| | <p>20.3.11.6.1 Introduction</p> <p>HT LDPC codes are described in 20.3.11.6.2 through 20.3.11.6.6. These codes are optionally used in the HT system as a high-performance error correcting code instead of the convolutional code (20.3.11.5). The LDPC encoder shall use the rate-dependent parameters in Table 20-29 through Table 20-43, with the exception of the N_{ES} parameter.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.1 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7; IEEE 802.11-2020 at § 19.3.11.7.</p> <p>Table 20-23—HT PHY MIB attributes (continued)</p> <table><tr><th>Managed object</th><th>Default value/range</th><th>Operational semantics</th></tr><tr><td colspan="3">* * *</td></tr><tr><td colspan="3">dot11PHYHTTable</td></tr><tr><td colspan="3">* * *</td></tr><tr><td>dot11LDPCCodingOptionImplemented</td><td>False/Boolean</td><td>Static</td></tr><tr><td>dot11LDPCCodingOptionActivated</td><td>False/Boolean</td><td>Dynamic</td></tr></table> <p>IEEE 802.11n-2009 at Table 20-23; <i>see also</i> IEEE 802.11-2012 at Table 20-24; IEEE 802.11-2020 at Table 19-24.</p> <p>Table 20-27—List of parameters for PMD primitives</p> <table><tr><th>Parameter</th><th>Associate primitive</th><th>Value</th></tr><tr><td colspan="3">* * *</td></tr></table> | Managed object | Default value/range | Operational semantics | * * * | | | dot11PHYHTTable | | | * * * | | | dot11LDPCCodingOptionImplemented | False/Boolean | Static | dot11LDPCCodingOptionActivated | False/Boolean | Dynamic | Parameter | Associate primitive | Value | * * * | | |
| Managed object | Default value/range | Operational semantics | | | | | | | | | | | | | | | | | | | | | | | |
| * * * | | | | | | | | | | | | | | | | | | | | | | | | | |
| dot11PHYHTTable | | | | | | | | | | | | | | | | | | | | | | | | | |
| * * * | | | | | | | | | | | | | | | | | | | | | | | | | |
| dot11LDPCCodingOptionImplemented | False/Boolean | Static | | | | | | | | | | | | | | | | | | | | | | | |
| dot11LDPCCodingOptionActivated | False/Boolean | Dynamic | | | | | | | | | | | | | | | | | | | | | | | |
| Parameter | Associate primitive | Value | | | | | | | | | | | | | | | | | | | | | | | |
| * * * | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | |
|----------------|--|--|--------|---|---------------------------|--|------|---------------------|------------|--------|---------|-------|--|--|--|--|----------|-------------------|-----------|--------|---|
| | <table><tr><td>FEC_CODING</td><td>PMD_TX_PARAMETERS.request</td><td>Indicates which FEC encoding is used. Enumerated type: BCC_CODING indicates binary convolutional code. LDPC_CODING indicates low-density parity check code.</td></tr></table> <p>IEEE 802.11n-2009 at Table 20-27 (emphasis added); <i>see also</i> IEEE 802.11-2012 at Table 20-28.</p> <p>A.4.19.2 HT PHY features</p> <table><tr><th>Item</th><th>Protocol capability</th><th>References</th><th>Status</th><th>Support</th></tr><tr><td colspan="5">* * *</td></tr><tr><td>HTP2.6.1</td><td>Use of LDPC codes</td><td>20.3.11.6</td><td>CF16:O</td><td>Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/></td></tr></table> <p>IEEE 802.11n-2009 at Annex A, § A.4.19.2; <i>see also</i> IEEE 802.11-2012 at Annex B, § B.4.19.2; IEEE 802.11-2020 at Annex B, § B.4.17.2.</p> | | | FEC_CODING | PMD_TX_PARAMETERS.request | Indicates which FEC encoding is used. Enumerated type: BCC_CODING indicates binary convolutional code. LDPC_CODING indicates low-density parity check code. | Item | Protocol capability | References | Status | Support | * * * | | | | | HTP2.6.1 | Use of LDPC codes | 20.3.11.6 | CF16:O | Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/> |
| FEC_CODING | PMD_TX_PARAMETERS.request | Indicates which FEC encoding is used. Enumerated type: BCC_CODING indicates binary convolutional code. LDPC_CODING indicates low-density parity check code. | | | | | | | | | | | | | | | | | | | |
| Item | Protocol capability | References | Status | Support | | | | | | | | | | | | | | | | | |
| * * * | | | | | | | | | | | | | | | | | | | | | |
| HTP2.6.1 | Use of LDPC codes | 20.3.11.6 | CF16:O | Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/> | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | |
|---|----------|---------------------------------------|-------------|--|---|
| | HEP8.2 | LDPC with more than 4 spatial streams | 27.3.12.5.2 | CFHE80:M CFHE20:M | Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/> |
| | HEP8.3 | LDPC with 4 or fewer spatial streams | 27.3.12.5.2 | (HEP6.5 OR HEP6.6 OR HEP6.7):M ((HEP3.2 OR HEP3.3 OR HEP3.4 OR HEP3.5) AND HEP2.1):M (HEP6.1 OR HEP6.2 OR HEP6.3 OR HEP6.4):O (HEP3.1 AND HEP2.1):O CFHE20:O | Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/> |
| <p>IEEE 802.11ax-2021 at Annex B, § B.4.33.2.</p> <p><i>See also</i> IEEE 802.11n-2009 at § 20.3.11.6.3, Annex G, Annex R; IEEE 802.11-2012 at § 20.3.11.7.3, Annex L, Annex F; IEEE 802.11-2020 at § 19.3.11.7.3, Annex I, Annex F.</p> <p>The Very High Throughput (“VHT”) PHY specification of the IEEE 802.11ac-2013 amendment to the IEEE 802.11-2012 standard, which uses LDPC coding, is based on and incorporates the LDPC coding used in the High Throughput (“HT”) PHY, as defined in Clause 20 of the IEEE 802.11-2012 standard.</p> | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>22. Very High Throughput (VHT) PHY specification</p> <p>22.1 Introduction</p> <p>22.1.1 Introduction to the VHT PHY</p> <p>Clause 22 specifies the PHY entity for a very high throughput (VHT) orthogonal frequency division multiplexing (OFDM) system.</p> <p>In addition to the requirements in Clause 22, a VHT STA shall be capable of transmitting and receiving PPDU's that are compliant with the mandatory PHY specifications defined in Clause 20.</p> <p>The VHT PHY is based on the HT PHY defined in Clause 20, which in turn is based on the OFDM PHY defined in Clause 18. The VHT PHY extends the maximum number of space-time streams supported to eight and provides support for downlink multi-user (MU) transmissions. A downlink MU transmission supports up to four users with up to four space-time streams per user with the total number of space-time streams not exceeding eight.</p> <p>IEEE 802.11ac-2013 at § 22.1.1 (emphasis added).</p> <p>22.3.10.5.4 LDPC coding</p> <p>For a VHT SU PPDU using LDPC coding to encode the Data field, the LDPC code and encoding process described in 20.3.11.7 (LDPC codes) shall be used with the following modifications. First, all bits in the Data field including the scrambled SERVICE, PSDU, and pad bits are encoded. Thus, N_{pld} for VHT PPDU's shall be computed using Equation (22-61) instead of Equation (20-35).</p> $N_{pld} = N_{SYM,init} N_{DBPS} \quad (22-61)$ <p>where</p> <p>$N_{SYM,init}$ is given by Equation (22-62)</p> $N_{SYM,init} = m_{STBC} \times \left\lceil \frac{8 \cdot APEP_LENGTH + N_{service}}{m_{STBC} \cdot N_{DBPS}} \right\rceil \quad (22-62)$ |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | <p>where</p> <p>m_{STBC} is equal to 2 when STBC is used, and 1 otherwise</p> <p>APEP_LENGTH is the TXVECTOR parameter APEP_LENGTH</p> <p>Following the calculation of N_{pld}, N_{avbits} shall be computed using Equation (22-63) instead of Equation (20-36).</p> $N_{avbits} = N_{SYM,init} N_{CBPS} \quad (22-63)$ <p>In addition, if N_{SYM} computed in Equation (20-41) in step (d) of 20.3.11.7.5 (LDPC PPDU encoding process) is greater than $N_{SYM,init}$, then the LDPC Extra OFDM Symbol field of VHT-SIG-A shall be set to 1. Otherwise, the LDPC Extra OFDM Symbol field of VHT-SIG-A shall be set to 0.</p> <p>LDPC codes used in VHT MU PPDU shall also follow the definitions in 20.3.11.7 (LDPC codes). Refer to 22.3.10.5.5 for a description of the LDPC encoding process for VHT MU PPDU.</p> <p>IEEE 802.11ac-2013 at § 22.3.10.5.4 (emphasis added).</p> <p>The High Efficiency (“HE”) PHY specification of the IEEE 802.11ax-2021 amendment to the IEEE 802.11-2020 standard, which uses LDPC coding, is based on and incorporates the LDPC coding used in the High Throughput (“HT”) PHY, as defined in Clause 19 of the IEEE 802.11-2020 standard.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>27. High-efficiency (HE) PHY specification</p> <p>27.1 Introduction</p> <p>27.1.1 Introduction to the HE PHY</p> <p>Clause 27 specifies the PHY entity for a high-efficiency (HE) orthogonal frequency division multiplexing (OFDM) system. In addition to the requirements in Clause 27, an HE STA shall be capable of transmitting and receiving PPDU's that are compliant with the mandatory requirements of the following PHY specifications:</p> <ul style="list-style-type: none">— Clause 19 and Clause 21 if the HE STA supports an operating channel width greater than or equal to 80 MHz and is operating in the 5 GHz band.— Clause 19 and Clause 21 transmission and reception on 20 MHz channel width (see 26.17.1) if the HE STA is a 20 MHz-only non-AP HE STA and is operating in the 5 GHz band.— Clause 19 if the HE STA is operating in the 2.4 GHz band.— Clause 17 if the HE STA is operating in the 6 GHz band. <p>For 2.4 GHz band operation, the HE PHY is based on HT PHY defined in Clause 19, which in turn is based on the OFDM PHY defined in Clause 17.</p> <p>For 5 GHz band operation, the HE PHY is based on the VHT PHY defined in Clause 21, which in turn is based on the HT PHY defined in Clause 19, which in turn is further based on the OFDM PHY defined in Clause 17.</p> <p>IEEE 802.11ax-2021 at § 27.1.1 (emphasis added).</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>27.3.12.5.2 LDPC coding</p> <p>For an HE SU PPDU or HE ER SU PPDU using LDPC coding on the Data field, the LDPC coding process described in 19.3.11.7 shall be used with the following modifications:</p> <p>First, all bits in the Data field including the scrambled SERVICE, PSDU, and pre-FEC pad bits are encoded. Thus, N_{pld} for HE PPDUs shall be computed using Equation (27-68) instead of Equation (19-35).</p> $N_{pld} = (N_{SYM,init} - m_{STBC})N_{DBPS} + m_{STBC}N_{DBPS,last,init} \quad (27-68)$ <p>where $N_{SYM,init}$ is defined in Equation (27-64).</p> <p>N_{avbits} shall be computed using Equation (27-69) instead of Equation (19-36).</p> $N_{avbits} = (N_{SYM,init} - m_{STBC}) \cdot N_{CBPS} + m_{STBC} \cdot N_{CBPS,last,init} \quad (27-69)$ <p>Using the pre-FEC padding factor a, compute N_{CBPS} of the last symbol using Equation (27-73).</p> $N_{CBPS,last} = \begin{cases} a \cdot N_{CBPS,short} & \text{if } a < 4 \\ N_{CBPS} & \text{if } a = 4 \end{cases} \quad (27-73)$ <p>The number of data bits of the last symbol is calculated as $N_{DBPS,last} = N_{DBPS,last,init}$.</p> <p>LDPC coding used in HE MU PPDUs shall also follow the definitions in 19.3.11.7. Refer to 27.3.12.5.4 for a description of the LDPC encoding process for HE MU PPDUs.</p> <p>IEEE 802.11ax-2021 at § 27.3.12.5.2 (emphasis added).</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|---|
| <p>[1a] receiving a collection of message bits having a first sequence in a source data stream;</p> | <p>The Accused Products receive a collection of message bits having a first sequence in a source data stream.</p> <p>For example, the Accused Products implement the IEEE Standards, which receive a collection of message bits having a first sequence in a source data stream.</p> <p>For example, the process of LDPC-encoding a signal per IEEE 802.11 includes receiving a collection of message bits having a first sequence in a source data stream. The IEEE 802.11 LDPC codes are block codes. Each source data stream to be encoded, <i>e.g.</i> an MPDU from the MAC layer, is divided into one or more collections of message bits as shown in Table 20-14.</p> <p>20.3.11.6.2 LDPC coding rates and codeword block lengths</p> <p>The supported coding rates, information block lengths, and codeword block lengths are described in Table 20-14.</p> <p style="text-align: right;">\</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|---|-----------------------------------|--------------------------------------|-----------------------------------|-----|-----|------|-----|-----|------|-----|-----|-----|-----|------|------|-----|-----|------|-----|-----|-----|-----|------|------|-----|-----|------|-----|-----|-----|-----|------|------|-----|------|------|-----|-----|-----|
| | <p style="text-align: center;">Table 20-14—LDPC parameters</p> <table><tr><th>Coding rate (R)</th><th>LDPC information block length (bits)</th><th>LDPC codeword block length (bits)</th></tr><tr><td>1/2</td><td>972</td><td>1944</td></tr><tr><td>1/2</td><td>648</td><td>1296</td></tr><tr><td>1/2</td><td>324</td><td>648</td></tr><tr><td>2/3</td><td>1296</td><td>1944</td></tr><tr><td>2/3</td><td>864</td><td>1296</td></tr><tr><td>2/3</td><td>432</td><td>648</td></tr><tr><td>3/4</td><td>1458</td><td>1944</td></tr><tr><td>3/4</td><td>972</td><td>1296</td></tr><tr><td>3/4</td><td>486</td><td>648</td></tr><tr><td>5/6</td><td>1620</td><td>1944</td></tr><tr><td>5/6</td><td>1080</td><td>1296</td></tr><tr><td>5/6</td><td>540</td><td>648</td></tr></table> <p>IEEE 802.11n-2009 at § 20.3.11.6.2 (emphasis added); <i>see also</i> 802.11-2012 at § 20.3.11.7.2; 802.11-2020 at § 19.3.11.7.2.</p> <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, \mathbf{c}, of size n, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = \mathbf{0}$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> | Coding rate (R) | LDPC information block length (bits) | LDPC codeword block length (bits) | 1/2 | 972 | 1944 | 1/2 | 648 | 1296 | 1/2 | 324 | 648 | 2/3 | 1296 | 1944 | 2/3 | 864 | 1296 | 2/3 | 432 | 648 | 3/4 | 1458 | 1944 | 3/4 | 972 | 1296 | 3/4 | 486 | 648 | 5/6 | 1620 | 1944 | 5/6 | 1080 | 1296 | 5/6 | 540 | 648 |
| Coding rate (R) | LDPC information block length (bits) | LDPC codeword block length (bits) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1/2 | 972 | 1944 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1/2 | 648 | 1296 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1/2 | 324 | 648 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2/3 | 1296 | 1944 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2/3 | 864 | 1296 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2/3 | 432 | 648 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3/4 | 1458 | 1944 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3/4 | 972 | 1296 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3/4 | 486 | 648 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5/6 | 1620 | 1944 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5/6 | 1080 | 1296 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5/6 | 540 | 648 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

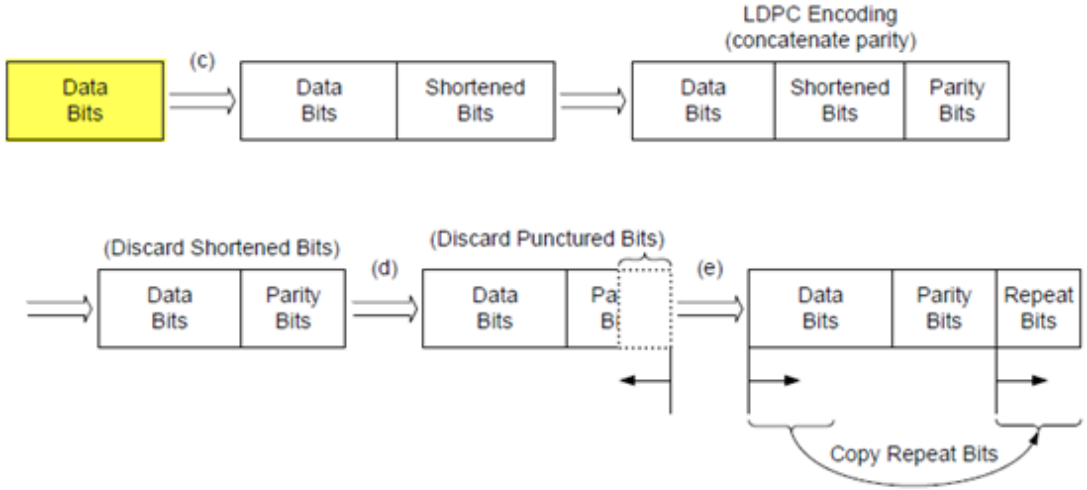
| Claim Language | Analysis |
|----------------|--|
| |  <p align="center">Figure 20-13—LDPC PDU encoding padding and puncturing of a single codeword</p> <p>IEEE 802.11n-2009 at Figure 20-13; <i>see also</i> IEEE 802.11-2012 at Figure 20-13; IEEE 802.11-2020 at Figure 19-13.</p> <p>Annex G of IEEE 802.11n-2009 and Annex L of IEEE 802.11-2012 and Annex I of IEEE 802.11-2020 provide examples of how a collection of message bits in the source data stream is received.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | |
|----------------|--|-----------------------------|----------------------------|-----------------------------|----------------------------|--------------|--------------|--------------|-----------|----------|----------|----------|------|------|------|-----------|----------|----------|----------|------|------|------|
| | <p>G.2.1 The message for LDPC example 1</p> <p>The message being encoded consists of the first 72 characters (shown in bold and including line breaks) of the well-known “Ode to Joy” by F. Schiller:</p> <p>Joy, bright spark of divinity, Daughter of Elysium, Fire-insired we tread Thy sanctuary. Thy magic power re-unites All that custom has divided, All men become brothers Under the sway of thy gentle wings.</p> <p>The message is converted to ASCII; then it is prepended with an appropriate MAC header, and a CRC32 is added. The resulting 100 octets PSDU is shown in Table G.33.</p> <p>IEEE 802.11n-2009 at Annex G, § G.2.1; <i>see also</i> IEEE 802.11-2012 at Annex L, § L.2.1; IEEE 802.11-2020 at Annex I, § I.2.2.</p> <p>G.2.5 Encoding data for LDPC example 1</p> <p>The DATA with shortening bits are LDPC encoded as a single (N_{CW}=1) codeword (L_{LDPC}=944; R=3/4) as prescribed by paragraph (c) of 20.3.11.6.5. The results are given in Table G.37.</p> <p>NOTE—The LDPC encoder appends 486 bits (i.e., bits 1458–1943) after the shortening bits.</p> <p>Table G.37—DATA bits for LDPC example 1 after LDPC encoding</p> <table><tr><th>Bit ##</th><th>Binary value b0 b7</th><th>Binary value b8 b15</th><th>Binary value b16 b23</th><th>Hex value</th><th>Hex value</th><th>Hex value</th></tr><tr><td>0000–0023</td><td>01101100</td><td>00011001</td><td>10001001</td><td>0x6C</td><td>0x19</td><td>0x89</td></tr><tr><td>0024–0047</td><td>10001111</td><td>01101000</td><td>00100001</td><td>0x8F</td><td>0x68</td><td>0x21</td></tr></table> | Bit ## | Binary value b0 b7 | Binary value b8 b15 | Binary value b16 b23 | Hex value | Hex value | Hex value | 0000–0023 | 01101100 | 00011001 | 10001001 | 0x6C | 0x19 | 0x89 | 0024–0047 | 10001111 | 01101000 | 00100001 | 0x8F | 0x68 | 0x21 |
| Bit ## | Binary value b0 b7 | Binary value b8 b15 | Binary value b16 b23 | Hex value | Hex value | Hex value | | | | | | | | | | | | | | | | |
| 0000–0023 | 01101100 | 00011001 | 10001001 | 0x6C | 0x19 | 0x89 | | | | | | | | | | | | | | | | |
| 0024–0047 | 10001111 | 01101000 | 00100001 | 0x8F | 0x68 | 0x21 | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | |
|----------------|---|----------|----------|------|------|------|-----------|----------|----------|----------|------|------|------|-----------|----------|----------|----------|------|------|------|
| | * * * | | | | | | | | | | | | | | | | | | | |
| | <div style="display: flex; justify-content: space-between; align-items: center;"> <table border="1" style="margin-bottom: 10px;"> <tr> <td>1896–1919</td><td>00100011</td><td>10011011</td><td>00110110</td><td>0x23</td><td>0x9B</td><td>0x36</td></tr> <tr> <td>1920–1943</td><td>00111110</td><td>00010101</td><td>00010001</td><td>0x3E</td><td>0x15</td><td>0x11</td></tr> </table> <p>IEEE 802.11n-2009 at Annex G, § G.2.5; <i>see also</i> IEEE 802.11-2012 at Annex L, § L.2.5; IEEE 802.11-2020 at Annex I, § I.2.6.</p> <p><i>See also</i> IEEE 802.11n-2009 at Annex G, §§ G.3.1, G.3.5; IEEE 802.11-2012 at Annex L, §§ L.3.1, L.3.5; IEEE 802.11-2020 at Annex I, §§ I.3.2, I.3.6.</p> </div> | | | | | | 1896–1919 | 00100011 | 10011011 | 00110110 | 0x23 | 0x9B | 0x36 | 1920–1943 | 00111110 | 00010101 | 00010001 | 0x3E | 0x15 | 0x11 |
| 1896–1919 | 00100011 | 10011011 | 00110110 | 0x23 | 0x9B | 0x36 | | | | | | | | | | | | | | |
| 1920–1943 | 00111110 | 00010101 | 00010001 | 0x3E | 0x15 | 0x11 | | | | | | | | | | | | | | |
| | <p>NOTE—This procedure does not describe the operation of optional features, such as LDPC or STBC</p> <p style="text-align: center;">Figure 20-20—PLCP transmit procedure (HT-mixed format PDU)</p> <p>IEEE 802.11n-2009 at Figure 20-20 (emphasis added); <i>see also</i> IEEE 802.11-2012 at Figure 20-22; IEEE 802.11-2020 at Figure 19-22.</p> | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

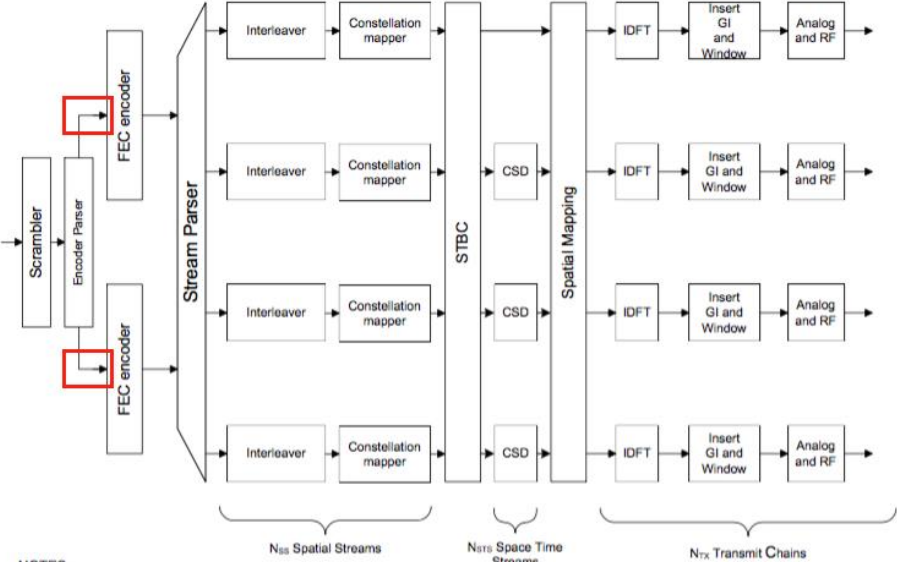
| Claim Language | Analysis |
|----------------|--|
| |  <p data-bbox="892 880 949 896">NOTES</p> <ul data-bbox="892 896 1491 998" style="list-style-type: none"> —There may be 1 or 2 FEC encoders when BCC encoding is used. —The stream parser may have 1, 2, 3 or 4 outputs. —When LDPC encoding is used, the interleavers are not used. —When STBC is used, the STBC block has more outputs than inputs. —When spatial mapping is used, there may be more transmit chains than space time streams. —The number of inputs to the spatial mapper may be 1, 2, 3, or 4. <p data-bbox="1113 1015 1528 1039" style="text-align: center;">Figure 20-3—Transmitter block diagram 2</p> <p data-bbox="751 1071 1869 1144">IEEE 802.11n-2009 at Fig. 20-3; IEEE 802.11-2012 at Fig. 20-3; IEEE 802.11-2020 at Fig. 19-3 (emphasis added)</p> <p data-bbox="751 1161 1921 1339">Further, the message bits have a first sequence. More specifically, the sequence of the message bits is defined as having a first sequence as part of the specification of the systematic code. Each LDPC codeword is of the form $\mathbf{c}=(i_0,i_1,\dots,i_{(k-1)},p_0,p_1,\dots,p_{(n-k-1)})$, where the message bits have a first sequence a listed from i_0 to $i_{(k-1)}$. The subscripts 0 to k-1 specify the individual message bits and index their order.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $c=(i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, c, of size n, $c=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $H \times c^T = 0$, where H is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); see also IEEE 802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> <p>As discussed above, the message bits are taken from (and thus “in”) a “source data stream.” The data stream is passed through to the MAC layer and then subsequently to the PHY layer. The PHY layer contains an LDPC encoder that receives the source data stream and then parses the data stream into blocks of message bits that are processed to generate a sequence of LDPC codewords. The 802.11 specification specifically describes how to take the message bits from the source data stream and parse the message bits into groups of message bits with specific first sequences. “A succession of LDPC codewords” results from LDPC encoding of the “message bits” in the “source data stream.” <i>See, e.g.</i>, 802.11n-2009 at § 20.3.11.6.2-6; 802.11-2012 at § 20.3.11.7.2-6; 802.11-2020 at § 19.3.11.7.2-6.</p> <p>20.3.11.6.6 LDPC parser</p> <p>The succession of LDPC codewords that result from the encoding process of 20.3.11.6.5 shall be converted into a bitstream in sequential fashion. Within each codeword, bit i_0 is ordered first. The parsing of this encoded data stream into spatial streams shall follow exactly the parsing rules defined for the BCC encoder, as defined in 20.3.11.7.1. However, the frequency interleaver of 20.3.11.7.3 is bypassed.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.6 (emphasis added); <i>see also</i> IEEE802.11-2012 at § 20.3.11.7.6; IEEE 802.11-2020 at § 19.3.11.7.6.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|---|
| | <p>infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> <p>For example, to the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe because they do not receive a “stream” of bits, <i>e.g.</i>, because they receive a “block” of bits, the Accused Products would infringe under the doctrine of equivalents even if the Court were to adopt an unduly narrow interpretation of “stream” (when in reality “stream” and “block” are not mutually exclusive) because the Accused Products receive a sequence of information bits to be encoded. Whether the bits are received as a stream or as a block under such an interpretation is an insubstantial difference, particularly because even if bits are received as a stream, they must be formed into a block for LDPC encoding. Thus, both scenarios accomplish substantially the same function of obtaining data to be encoded, in substantially the same way, <i>i.e.</i>, receiving bits of data to form a block to be encoded, and accomplish substantially the same result of preparing data for encoding by the LDPC encoder.</p> |
| <p>[1b] generating a sequence of parity bits, wherein each parity bit “x_j” in the sequence is in accordance with the formula</p> $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$ <p>where “x_{j-1}” is the value of a parity bit “$j-1$,” and</p> $\sum_{i=1}^a v_{(j-1)a+i}$ | <p>The Accused Products generate a sequence of parity bits, wherein each parity bit “x_j” in the sequence is in accordance with the formula $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$, where “$x_{j-1}$” is the value of a parity bit “$j-1$,” and $\sum_{i=1}^a v_{(j-1)a+i}$ is the value of a sum of “a” randomly chosen irregular repeats of the message bits.</p> <p>For example, the Accused Products implement the IEEE Standards, which generate a sequence of parity bits, wherein each parity bit “x_j” in the sequence is in accordance with the formula $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$, where “$x_{j-1}$” is the value of a parity bit “$j-1$,” and $\sum_{i=1}^a v_{(j-1)a+i}$ is the value of a sum of “a” randomly chosen irregular repeats of the message bits.</p> <p>The use of LDPC coding requires the generation of parity bits. The LDPC codes are systematic codes wherein each codeword is formed by combining the message bits with parity bits.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|--|
| <p>is the value of a sum of “a” randomly chosen irregular repeats of the message bits; and</p> | <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $c=(i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, c, of size n, $c=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $H \times c^T = 0$, where H is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> <p>For example, the LDPC encoder outputs a codeword “c” of size “n” that includes “k” information bits and “n-k” parity bits. The bits p_0 through $p_{(n-k-1)}$ are the a “sequence of parity bits.”</p> <p>20.3.11.6.6 LDPC parser</p> <p>The succession of LDPC codewords that result from the encoding process of 20.3.11.6.5 shall be converted into a bitstream in sequential fashion. Within each codeword, bit i_0 is ordered first. The parsing of this encoded data stream into spatial streams shall follow exactly the parsing rules defined for the BCC encoder, as defined in 20.3.11.7.1. However, the frequency interleaver of 20.3.11.7.3 is bypassed.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.6 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.6; IEEE 802.11-2020 at § 19.3.11.7.6.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | <p>Figure 20-13—LDPC PDU encoding padding and puncturing of a single codeword</p> <p>IEEE 802.11n-2009 at Figure 20-13; IEEE 802.11-2012 at Figure 20-13; IEEE 802.11-2020 at Figure 19-13.</p> <p>The formula $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$ is a recursive formula. The parity bit x_j is formed from summing a previously calculated parity bit x_{j-1} (“predecessor parity bit”) and adding it to another value, $\sum_{i=1}^a v_{(j-1)a+i}$ (value of a sum of “a” randomly chosen irregular repeats of the message bits). Each parity bit for each LDPC code described in the 802.11 standard meets this limitation.</p> <p>For example, in the (648, 432) (Rate=2/3) codeword, the parity bits have the following relationship:</p> <p>$p_1 = c_1$</p> <p>$p_{54} = p_1 + c_2$</p> <p>$p_{81} = p_{54} + c_3$</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p> $p_{108} = p_{81} + c_4$ $p_{135} = p_{108} + c_5$ $p_{162} = p_{135} + c_6$ $p_{189} = p_{162} + c_7$ $p_{216} = p_{189} + c_8$ </p> <p>The 802.11 standard has specified LDPC codes where the recursive relationship between the parity bits does not use strictly ascending parity bit indices. Nonetheless, the parity bits are still generated in accordance with formulate $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$.</p> <p>For example, let $x_1 = p_1$, $x_2 = p_{54}$, $x_3 = p_{81}$, $x_4 = p_{108}$, $x_5 = p_{135}$, $x_6 = p_{162}$, $x_7 = p_{189}$ and $x_8 = p_{216}$. It can be seen that x_i is still a sequence of parity bits. It is simply in a different order than the parity bits p_i. Now, writing the relationships above again provides:</p> <p> $x_1 = c_1$ $x_2 = x_1 + c_2$ $x_3 = x_2 + c_3$ $x_4 = x_3 + c_4$ $x_5 = x_4 + c_5$ $x_6 = x_5 + c_6$ $x_7 = x_6 + c_7$ $x_8 = x_7 + c_8$ </p> <p>This can be written as:</p> <p align="right">$x_j = x_{j-1} + c_j$</p> <p>Thus each parity bit x_j is in accordance with</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$ <p>as long as</p> $c_j = \sum_{i=1}^a v_{(j-1)a+i}$ <p>which is shown below.</p> <p>Alternatively, these parity bits may be written as follows, which also meets the limitation of this section:</p> $p_1 = c_1$ $p_{54} = p_1 + c_2$ $p_{81} = p_{54} + c_3$ $p_{108} = p_{81} + c_4$ $p_{135} = p_{108} + c_5$ $p_{216} = p_1 + c'_6$ $p_{189} = p_{216} + c'_7$ $p_{162} = p_{189} + c'_8$ <p>For avoidance of doubt, the (648, 432) code used here as an example has 216 parity bits. Only eight of the parity bits have been shown here for brevity. However, the remaining parity bits are also generated in accordance with the same equation, i.e. $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$.</p> <p>Further, $\sum_{i=1}^a v_{(j-1)a+i}$ is the value of a sum of “a” randomly chosen irregular repeats of the message bits.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | <p>Each of the $c_j = \sum_{i=1}^a v_{(j-1)a+i}$ values is the sum of a message bits. For example, the (648, 432) code has c_j values as follows:</p> $c_1 = i_6 + i_{14} + i_{16} + i_{17} + \dots + i_{423} + i_{427}$ $c_2 = i_{24} + i_{52} + i_{67} + i_{127} + i_{163} + i_{219} + i_{304} + i_{366} + i_{422}$ $c_3 = i_9 + i_{35} + i_{68} + i_{91} + i_{161} + i_{189} + i_{287} + i_{331} + i_{387}$ $c_4 = i_{15} + i_{28} + i_{73} + i_{106} + i_{128} + i_{167} + i_{216} + i_{268} + i_{303}$ $c_5 = i_9 + i_{39} + i_{58} + i_{107} + i_{137} + i_{195} + i_{295} + i_{363} + i_{421}$ $c_6 = i_5 + i_{13} + i_{15} + i_{16} + \dots + i_{422} + i_{426}$ $c_7 = i_5 + i_{48} + i_{62} + i_{100} + i_{159} + i_{205} + i_{250} + i_{310} + i_{368}$ $c_8 = i_{13} + i_{49} + i_{74} + i_{91} + i_{127} + i_{185} + i_{233} + i_{288} + i_{426}$ <p>The number of bits summed, “a,” in each of these equations is 63, 9, 9, 9, 9, 55, 9, 9.</p> <p>The alternative representation is reproduced here:</p> $p_1 = c_1$ $p_{54} = p_1 + c_2$ $p_{81} = p_{54} + c_3$ $p_{108} = p_{81} + c_4$ $p_{135} = p_{108} + c_5$ $p_{216} = p_1 + c'_6$ $p_{189} = p_{216} + c'_7$ $p_{162} = p_{189} + c'_8$ |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>In this representation, p_1, p_{54}, p_{81}, p_{108} and p_{135} are determined exactly the same way. Thus, only analysis of p_{216}, p_{189}, p_{162} is needed. Starting with c'_6, it is also the sum “a” of message bits as shown below:</p> $c'_6 = i_{16} + i_{37} + i_{64} + i_{100} + i_{155} + i_{214} + i_{245} + i_{341} + i_{403}$ <p>With respect to c'_7, it relates p_{189} to p_{216}. C_8 also does this. Thus, they are equal:</p> $c'_7 = c_8 = i_{13} + i_{49} + i_{74} + i_{91} + i_{127} + i_{185} + i_{233} + i_{288} + i_{426}$ <p>Likewise, c'_8 and c_7 relate p_{162} to p_{189} and are equal to each other:</p> $c'_8 = c_7 = i_5 + i_{48} + i_{62} + i_{100} + i_{159} + i_{205} + i_{250} + i_{310} + i_{368}$ <p>In this alternative representation, the number of information bits summed for summed for c_1, c_2, c_3, c_4, c_5, c'_6, c'_7 and c'_8 is 63, 9, 9, 9, 9, 9, 9 and 9, respectively.</p> <p>Thus the sum terms $\sum_{i=1}^a v_{(j-1)a+i}$ are sums of message bits with an associated “a” values.</p> <p>Further, 802.11 LDPC codes use an irregular repeat of message bits. For example, an analysis of an exemplary Tanner Graph for the (648, 432) code (<i>See Appendix B</i>) yields the following frequencies of irregularly repeated message bits:</p> $f_5 = 18.75\%$ $f_6 = 31.25\%$ $f_7 = 12.5\%$ $f_8 = 18.75\%$ $f_{13} = 6.25\%$ $f_{15} = 12.5\%$ <p>In this example, 81 info bits are repeated 5 times, 135 are repeated 6 times, 54 are repeated 7 times, 81 are repeated 8 times, 27 are repeated 13 times, and 54 are repeated 15 times. Thus, the message bits are repeated irregularly. These frequencies are illustrative only; to</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | <p>the extent the Accused Products use alternative frequencies, they literally infringe so long as at least two message bits appear with different frequency.</p> <p>Finally, the sum $\sum_{i=1}^a v_{(j-1)a+i}$ is of “a” “randomly chosen” irregular repeats of the message bits. In the field of error control coding, randomly chosen includes the use of fixed randomization functions. Random selections may be made with permutations or interleavers.</p> <p>The 802.11 standard defines a codeword $\mathbf{c} = (i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, with information bits i and parity bits p. The parity bits p may be determined from a parity check matrix \mathbf{H} by computing $\mathbf{H} \times \mathbf{c}^T = 0$.</p> <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, \mathbf{c}, of size n, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = 0$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> <p>The parity check matrix \mathbf{H} for each block size and code rate is defined in Tables R.1 to R.3 of the 802.11n-2009 amendment, and in Tables F-1 to F-3 of the 802.11-2012 standard and the 802.11-2020 standard.</p> <p>Table R.1 defines the matrix prototypes of the parity-check matrices for a codeword block length $n=648$ bits, with a subblock size $Z=27$ bits.</p> <p align="center">Table R.1—Matrix prototypes for codeword block length $n=648$ bits, subblock size is $Z = 27$ bits</p> <p align="center">* * *</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|----|----|----|---|---|----|---|----|---|----|----|---|----|---|---|---|---|----|---|---|---|---|---|---|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|----|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|----|---|----|---|---|----|---|---|---|---|---|---|---|----|---|----|---|---|---|---|----|----|----|---|---|---|----|---|----|---|----|---|---|---|---|---|---|---|---|----|----|----|---|----|----|---|----|---|---|---|----|---|---|---|---|---|---|---|---|---|---|
| | <div>(c) Coding rate $R = 3/4$.</div> <table><tr><td>16</td><td>17</td><td>22</td><td>24</td><td>9</td><td>3</td><td>14</td><td>-</td><td>4</td><td>2</td><td>7</td><td>-</td><td>26</td><td>-</td><td>2</td><td>-</td><td>21</td><td>-</td><td>1</td><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>12</td><td>12</td><td>3</td><td>3</td><td>26</td><td>6</td><td>21</td><td>-</td><td>15</td><td>22</td><td>-</td><td>15</td><td>-</td><td>4</td><td>-</td><td>-</td><td>16</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>18</td><td>26</td><td>16</td><td>22</td><td>23</td><td>9</td><td>-</td><td>0</td><td>-</td><td>4</td><td>-</td><td>4</td><td>-</td><td>8</td><td>23</td><td>11</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td></tr><tr><td>9</td><td>7</td><td>0</td><td>1</td><td>17</td><td>-</td><td>-</td><td>7</td><td>3</td><td>-</td><td>3</td><td>23</td><td>-</td><td>16</td><td>-</td><td>-</td><td>21</td><td>-</td><td>0</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td></tr><tr><td>24</td><td>5</td><td>26</td><td>7</td><td>1</td><td>-</td><td>-</td><td>15</td><td>24</td><td>15</td><td>-</td><td>8</td><td>-</td><td>13</td><td>-</td><td>13</td><td>-</td><td>11</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td></tr><tr><td>2</td><td>2</td><td>19</td><td>14</td><td>24</td><td>1</td><td>15</td><td>19</td><td>-</td><td>21</td><td>-</td><td>2</td><td>-</td><td>24</td><td>-</td><td>3</td><td>-</td><td>2</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td></tr></table> <p>IEEE 802.11n-2009 at Annex R, Table R.1; <i>see also</i> IEEE 802.11-2012 at Annex F, Table F-1; IEEE 802.11-2020 at Annex F, Table F-1.</p> <p>Each of the parity-check matrices can be partitioned into square subblocks (submatrices) of size $Z \times Z$. These submatrices are either cyclic-permutations of the identity matrix or null submatrices.</p> <p>The cyclic-permutation matrix P_i is obtained from the $Z \times Z$ identity matrix by cyclically shifting the columns to the right by i elements. The matrix P_0 is the $Z \times Z$ identity matrix. Figure 20-12 illustrates examples (for a subblock size of 8×8) of cyclic-permutation matrices P_i.</p> <div>$P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, P_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, P_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$</div> <p>Figure 20-12—Examples of cyclic-permutation matrices with Z=8</p> <p>Table R.1 displays the “matrix prototypes” of parity-check matrices for all four coding rates at block length $n=648$ bits. The integer i denotes the cyclic-permutation matrix P_i, as illustrated in Figure 20-12. Vacant entries of the table denote null (zero) submatrices.</p> | 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | - | 0 | 0 | - | 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | - | 0 | 0 | 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 |
| 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | - | 0 | 0 | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | - | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>IEEE 802.11n-2009 at § 20.3.11.6.4; <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.4; IEEE 802.11-2020 at § 19.3.11.7.4.</p> <p>The parity check matrices may also be represented as a combination of two matrices, \mathbf{H}_1 and \mathbf{H}_2, where $\mathbf{H} = [\mathbf{H}_1 \mathbf{H}_2]$, \mathbf{H}_1 is an $(n-k) \times k$ matrix, and \mathbf{H}_2 is an $(n-k) \times (n-k)$ matrix, such that $\mathbf{H}_1 \times i^T + \mathbf{H}_2 \times p^T = 0$, $i = (i_0, i_1, \dots, i_{(k-1)})$ and $p = (p_0, p_1, \dots, p_{(n-k-1)})$.</p> <p>The irregular repeats are chosen via a randomizing function as specified by the parity check matrices, H, in the 802.11 standard.</p> <p>Annex G of IEEE 802.11n-2009 and Annex L of IEEE 802.11-2012 and Annex I of IEEE 802.11-2020 provide examples of calculating the sequence of parity bits, and further illustrate randomly chosen irregular repeats of the message bits.</p> <p>G.2.1 The message for LDPC example 1</p> <p>The message being encoded consists of the first 72 characters (shown in bold and including line breaks) of the well-known “Ode to Joy” by F. Schiller:</p> <p>Joy, bright spark of divinity, Daughter of Elysium, Fire-insired we tread Thy sanctuary. Thy magic power re-unites All that custom has divided, All men become brothers Under the sway of thy gentle wings.</p> <p>The message is converted to ASCII; then it is prepended with an appropriate MAC header, and a CRC32 is added. The resulting 100 octets PSDU is shown in Table G.33.</p> <p>IEEE 802.11n-2009 at Annex G, § G.2.1; <i>see also</i> IEEE 802.11-2012 at Annex L, § L.2.1; IEEE 802.11-2020 at Annex I, § I.2.2.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|---|---------------------------|---------------------------|---------------------------|---------------------------|--------------|--------------|--------------|-----------|----------|----------|----------|------|------|------|-----------|----------|----------|----------|------|------|------|-------|--|--|--|--|--|--|-----------|----------|----------|----------|------|------|------|-----------|----------|----------|----------|------|------|------|
| | <p>G.2.5 Encoding data for LDPC example 1</p> <p>The DATA with shortening bits are LDPC encoded as a single ($N_{CW}=1$) codeword ($L_{LDPC}=944$; $R=3/4$) as prescribed by paragraph (c) of 20.3.11.6.5. The results are given in Table G.37.</p> <p>NOTE—The LDPC encoder appends 486 bits (i.e., bits 1458–1943) after the shortening bits.</p> <p>Table G.37—DATA bits for LDPC example 1 after LDPC encoding</p> <table><tr><th>Bit ##</th><th>Binary value b0 b7</th><th>Binary value b8 b15</th><th>Binary value b16 b23</th><th>Hex value</th><th>Hex value</th><th>Hex value</th></tr><tr><td>0000–0023</td><td>01101100</td><td>00011001</td><td>10001001</td><td>0x6C</td><td>0x19</td><td>0x89</td></tr><tr><td>0024–0047</td><td>10001111</td><td>01101000</td><td>00100001</td><td>0x8F</td><td>0x68</td><td>0x21</td></tr><tr><td colspan="7">* * *</td></tr><tr><td>1896–1919</td><td>00100011</td><td>10011011</td><td>00110110</td><td>0x23</td><td>0x9B</td><td>0x36</td></tr><tr><td>1920–1943</td><td>00111110</td><td>00010101</td><td>00010001</td><td>0x3E</td><td>0x15</td><td>0x11</td></tr></table> <p>IEEE 802.11n-2009 at Annex G, § G.2.5; <i>see also</i> IEEE 802.11-2012 at Annex L, § L.2.5; IEEE 802.11-2020 at Annex I, § I.2.6.</p> <p><i>See also</i> IEEE 802.11n-2009 at Annex G, § § G.3.1, G.3.5; IEEE 802.11-2012 at Annex L, § L.3.1, L.3.5; IEEE 802.11-2020 at Annex I, § I.3.2, I.3.6..</p> <p><i>See also</i> Appendix B (illustrating randomly chosen irregular repeats of the message bits in exemplary Tanner Graphs).</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> | Bit ## | Binary value b0 b7 | Binary value b8 b15 | Binary value b16 b23 | Hex value | Hex value | Hex value | 0000–0023 | 01101100 | 00011001 | 10001001 | 0x6C | 0x19 | 0x89 | 0024–0047 | 10001111 | 01101000 | 00100001 | 0x8F | 0x68 | 0x21 | * * * | | | | | | | 1896–1919 | 00100011 | 10011011 | 00110110 | 0x23 | 0x9B | 0x36 | 1920–1943 | 00111110 | 00010101 | 00010001 | 0x3E | 0x15 | 0x11 |
| Bit ## | Binary value b0 b7 | Binary value b8 b15 | Binary value b16 b23 | Hex value | Hex value | Hex value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0000–0023 | 01101100 | 00011001 | 10001001 | 0x6C | 0x19 | 0x89 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0024–0047 | 10001111 | 01101000 | 00100001 | 0x8F | 0x68 | 0x21 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| * * * | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1896–1919 | 00100011 | 10011011 | 00110110 | 0x23 | 0x9B | 0x36 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1920–1943 | 00111110 | 00010101 | 00010001 | 0x3E | 0x15 | 0x11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|--|
| | <p>For example, to the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe because they do not “repeat” information bits, <i>e.g.</i>, by construing “repeat” to require copies of information bits to be stored in a separate memory, the Accused Products would infringe under the doctrine of equivalents because each information bit is used multiple times in the first encoding, and appears in a plurality of sums of information bits at the check nodes. Whether or not the information bits are duplicated and separately stored in memory before they are summed or combined is an insubstantial difference: both scenarios accomplish substantially the same function of summing pluralities of information bits at the check nodes, in substantially the same way, <i>i.e.</i>, by directing each information bit to a plurality of check nodes to be summed or combined, and accomplish substantially the same result of forming the first encoded data block.</p> |
| <p>[1c] making the sequence of parity bits available for transmission in a transmission data stream.</p> | <p>The Accused Products make the sequence of parity bits available for transmission in a transmission data stream.</p> <p>For example, the Accused Products implement the IEEE Standards, which make the sequence of parity bits available for transmission in a transmission data stream.</p> <p>For example, the LDPC encoder makes available for transmission a codeword “c” of size “n” that includes “k” information bits and “n-k” parity bits in a transmission data stream.</p> <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, <i>i.e.</i>, it encodes an information block, $c=(i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, c, of size n, $c=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $H \times c^T = 0$, where H is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>20.3.11.6.6 LDPC parser</p> <p>The succession of LDPC codewords that result from the encoding process of 20.3.11.6.5 shall be converted into a bitstream in sequential fashion. Within each codeword, bit i_0 is ordered first. The parsing of this encoded data stream into spatial streams shall follow exactly the parsing rules defined for the BCC encoder, as defined in 20.3.11.7.1. However, the frequency interleaver of 20.3.11.7.3 is bypassed.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.6 (emphasis added); <i>see also</i> IEEE802.11-2012 at § 20.3.11.7.6; IEEE 802.11-2020 at § 19.3.11.7.6.</p> <p>Figure 20-13—LDPC PDU encoding padding and puncturing of a single codeword</p> <p>IEEE 802.11n-2009 at Figure 20-13; <i>see also</i> IEEE 802.11-2012 at Figure 20-13; IEEE 802.11-2020 at Figure 19-13.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>NOTES</p> <ul style="list-style-type: none">—There may be 1 or 2 FEC encoders when BCC encoding is used.—The stream parser may have 1, 2, 3 or 4 outputs.—When LDPC encoding is used, the interleavers are not used—When STBC is used, the STBC block has more outputs than inputs.—When spatial mapping is used, there may be more transmit chains than space time streams.—The number of inputs to the spatial mapper may be 1, 2, 3, or 4. <p>IEEE 802.11n-2009 at Fig. 20-3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at Fig. 20-3; IEEE 802.11-2020 at Fig. 19-3.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|---|
| Claim 2 | |
| [2] The method of claim 1, wherein the sequence of parity bits is generated is in accordance with “a” being constant. | <p>The Accused Products practice the method of claim 1.</p> <p>See <i>supra</i> [1]-[1c].</p> <p>In the Accused Products, the sequence of parity bits is generated is in accordance with “a” being constant.</p> <p>For example, the Accused Products implement the IEEE Standards, and in said standards the sequence of parity bits is generated is in accordance with “a” being constant.</p> <p>Analysis of the twelve codes defined in the standard reveals that the following values for “a” being used when generating the sequence of parity bits:</p> <p>(1944, 974) – 5, 6, 57</p> <p>(1296, 648) – 5, 6, 57</p> <p>(648, 324) – 5, 6, 53</p> <p>(1944, 1296) – 9, 65</p> <p>(1296, 864) – 9, 69</p> <p>(648, 432) – 9, 63</p> <p>(1944, 1458) – 12, 72</p> <p>(1296, 972) - 12, 13, 73</p> <p>(648, 486) - 12, 13, 67</p> <p>(1944, 1620) – 17, 18, 70</p> <p>(1296, 1080) – 19, 74</p> <p>(648, 540) – 20, 75</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|--|
| | <p>At least the (648, 432) and (1944,1458) codes infringe this limitation under the doctrine of equivalents. Each subset of message bits in these codes contains either a first number of message bits, a, or a number of message bits that is a multiple of a. For example, in the (1944, 1458) code, each subset contains either 12 or $6 * 12 = 72$ message bits. The larger subsets can be equivalently represented as a sequence of smaller subsets, wherein only the final parity bit generated by the sequence is used in the codeword. For example, a subset of 72 can be equivalently generated using six subsets of 12, in which the parity bits generated by the first five subsets are not used—which would literally infringe claim 7. Whether a single large subset or a plurality of smaller subsets are used is insubstantial: in both scenarios, substantially the same function of generating parity bits from sums of message bits is performed, in substantially the same way, <i>i.e.</i>, accumulating subsets of message bits, to obtain substantially the same result of encoding a codeword using an encoding operation with linear time complexity.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 3 | |
| <p>[3] The method of claim 1, wherein the sequence of parity bits is generated is in accordance with “a” varying for different parity bits.</p> | <p>The Accused Products practice the method of claim 1.</p> <p>See <i>supra</i> [1]-[1c].</p> <p>In the Accused Products, the sequence of parity bits is generated is in accordance with “a” varying for different parity bits.</p> <p>For example, the Accused Products implement the IEEE Standards, and in said standards the sequence of parity bits is generated is in accordance with “a” varying for different parity bits.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | <p>As shown below, the parity check matrices of the LDPC codes in the 802.11 standard use a dual diagonal pattern. This dual diagonal pattern permits efficient encoding.</p> <p>The 802.11 standard defines a codeword $\mathbf{c} = (i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, with information bits i and parity bits p. The parity bits p may be determined from a parity check matrix \mathbf{H} by computing $\mathbf{H} \times \mathbf{c}^T = 0$.</p> <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, \mathbf{c}, of size n, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = 0$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> <p>The parity check matrix \mathbf{H} for each block size and code rate is defined in Tables R.1 to R.3 of the 802.11n-2009 amendment, and in Tables F-1 to F-3 of the 802.11-2012 standard and the 802.11-2020 standard.</p> <p>Table R.1 defines the matrix prototypes of the parity-check matrices for a codeword block length $n=648$ bits, with a subblock size $Z=27$ bits.</p> <p align="center">Table R.1—Matrix prototypes for codeword block length $n=648$ bits, subblock size is $Z = 27$ bits</p> <p align="center">* * *</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|----|----|----|---|---|----|---|----|---|----|----|---|----|---|---|---|---|----|---|---|---|---|---|---|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|----|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|----|---|----|---|---|----|---|---|---|---|---|---|---|----|---|----|---|---|---|---|----|----|----|---|---|---|----|---|----|---|----|---|---|---|---|---|---|---|---|----|----|----|---|----|----|---|----|---|---|---|----|---|---|---|---|---|---|---|---|---|---|
| | <div>(c) Coding rate $R = 3/4$.</div> <table><tr><td>16</td><td>17</td><td>22</td><td>24</td><td>9</td><td>3</td><td>14</td><td>-</td><td>4</td><td>2</td><td>7</td><td>-</td><td>26</td><td>-</td><td>2</td><td>-</td><td>21</td><td>-</td><td>1</td><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>12</td><td>12</td><td>3</td><td>3</td><td>26</td><td>6</td><td>21</td><td>-</td><td>15</td><td>22</td><td>-</td><td>15</td><td>-</td><td>4</td><td>-</td><td>-</td><td>16</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>18</td><td>26</td><td>16</td><td>22</td><td>23</td><td>9</td><td>-</td><td>0</td><td>-</td><td>4</td><td>-</td><td>4</td><td>-</td><td>8</td><td>23</td><td>11</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td></tr><tr><td>9</td><td>7</td><td>0</td><td>1</td><td>17</td><td>-</td><td>-</td><td>7</td><td>3</td><td>-</td><td>3</td><td>23</td><td>-</td><td>16</td><td>-</td><td>-</td><td>21</td><td>-</td><td>0</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td></tr><tr><td>24</td><td>5</td><td>26</td><td>7</td><td>1</td><td>-</td><td>-</td><td>15</td><td>24</td><td>15</td><td>-</td><td>8</td><td>-</td><td>13</td><td>-</td><td>13</td><td>-</td><td>11</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td></tr><tr><td>2</td><td>2</td><td>19</td><td>14</td><td>24</td><td>1</td><td>15</td><td>19</td><td>-</td><td>21</td><td>-</td><td>2</td><td>-</td><td>24</td><td>-</td><td>3</td><td>-</td><td>2</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td></tr></table> <p>IEEE 802.11n-2009 at Annex R, Table R.1; <i>see also</i> IEEE 802.11-2012 at Annex F, Table F-1; IEEE 802.11-2020 at Annex F, Table F-1.</p> <p>Each of the parity-check matrices can be partitioned into square subblocks (submatrices) of size $Z \times Z$. These submatrices are either cyclic-permutations of the identity matrix or null submatrices.</p> <p>The cyclic-permutation matrix P_i is obtained from the $Z \times Z$ identity matrix by cyclically shifting the columns to the right by i elements. The matrix P_0 is the $Z \times Z$ identity matrix. Figure 20-12 illustrates examples (for a subblock size of 8×8) of cyclic-permutation matrices P_i.</p> <div>$P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, P_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, P_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$</div> <p>Figure 20-12—Examples of cyclic-permutation matrices with $Z=8$</p> <p>Table R.1 displays the “matrix prototypes” of parity-check matrices for all four coding rates at block length $n=648$ bits. The integer i denotes the cyclic-permutation matrix P_i, as illustrated in Figure 20-12. Vacant entries of the table denote null (zero) submatrices.</p> | 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | 0 | 0 | - | - | 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | 0 | 0 | - | 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 |
| 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | 0 | 0 | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | 0 | 0 | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

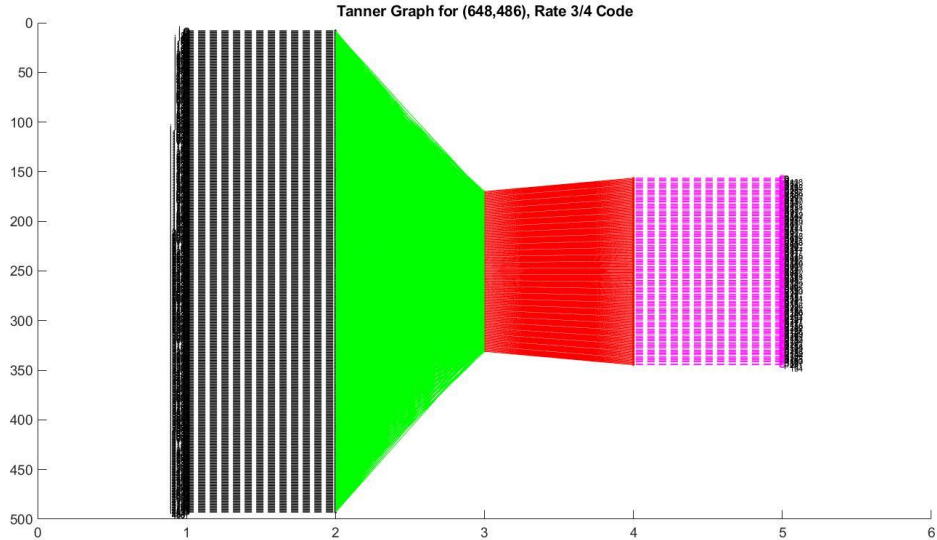
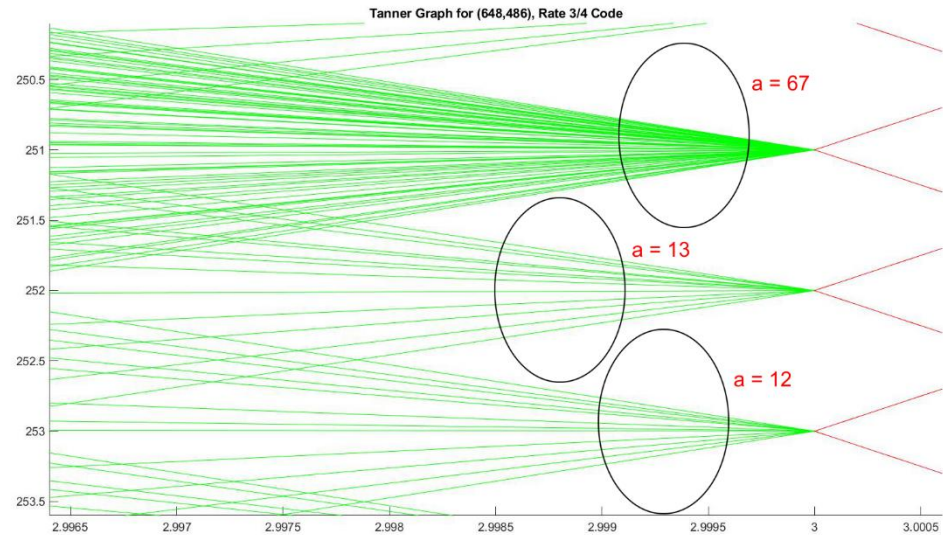
| Claim Language | Analysis |
|----------------|---|
| | <p>IEEE 802.11n-2009 at § 20.3.11.6.4; <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.4; IEEE 802.11-2020 at § 19.3.11.7.4.</p> <p>The parity check matrices may also be represented as a combination of two matrices, \mathbf{H}_1 and \mathbf{H}_2, where $\mathbf{H} = [\mathbf{H}_1 \mathbf{H}_2]$, \mathbf{H}_1 is an $(n-k) \times k$ matrix, and \mathbf{H}_2 is an $(n-k) \times (n-k)$ matrix, such that $\mathbf{H}_1 \times i^T + \mathbf{H}_2 \times p^T = 0$, $i = (i_0, i_1, \dots, i_{(k-1)})$ and $p = (p_0, p_1, \dots, p_{(n-k-1)})$. .</p> <p>The parity check matrices may be used to generate a Tanner Graph that reflects the code structure. The Tanner Graph representation of the 802.11 parity check matrices show the varying values of a used. For example, the figure below illustrates an exemplary Tanner Graph for the (648, 486) ($R = 3/4$) code.</p>  <p><i>See also</i> Appendix B (showing exemplary Tanner Graph representations for all 12 parity check matrices defined in the 802.11 standards).</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

For example, in the zoomed-in portion of the exemplary Tanner Graph shown below for the (648, 486) code ($R = 3/4$), the solid, black ovals illustrate the value of “ a ” for various check nodes.



Every code in the 802.11 Standards may be literally implemented with a varying. The following list shows the codes and the varying values of a used:

- (1944, 974) – 5, 6, 57
- (1296, 648) – 5, 6, 57
- (648, 324) – 5, 6, 53
- (1944, 1296) – 9, 65
- (1296, 864) – 9, 69
- (648, 432) – 9, 63
- (1944, 1458) – 12, 72

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|---|
| | <p>(1296, 972) - 12, 13, 73</p> <p>(648, 486) - 12, 13, 67</p> <p>(1944, 1620) – 17, 18, 70</p> <p>(1296, 1080) – 19, 74</p> <p>(648, 540) – 20, 75</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 4 | |
| <p>[4] The method of claim 1, wherein generating the sequence of parity bits comprises performing recursive modulo two addition operations on the random sequence of bits.</p> | <p>The Accused Products practice the method of claim 1.</p> <p>See <i>supra</i> [1]-[1c].</p> <p>In the Accused Products, generating the sequence of parity bits comprises performing recursive modulo two addition operations on the random sequence of bits.</p> <p>For example, the Accused Products implement the IEEE Standards, and in said standards generating the sequence of parity bits comprises performing recursive modulo two addition operations on the random sequence of bits.</p> <p>The calculation of the parity bits per claim 1 is already recursive, as required by:</p> $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$ <p>All arithmetic in the 802.11 LDPC codes is performed using modulo two arithmetic including modulo two addition.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|---|
| | <p>For example, Annex T of the 802.11n-2009 amendment discloses a “waveform generator tool [that was] written to model the PHY transmission process described in Clause 17, Clause 19, and Clause 20.” <i>See</i> 802.11n-2009 at Annex T, § T.1. The waveform generator tool is written in Matlab format and is available in IEEE 802.11 document 11-06/1715. <i>See id.</i>; <i>see also</i> IEEE 802.11 11-06/1715. The waveform generator tool includes a function named <code>ldpc_encode(inp,ldpc_params)</code> that performs LDPC encoding. <i>See</i> IEEE doc. 11-06/1715 pp. 14-16. This function contains the lines:</p> <pre>% generate parity parity = bitand((generator_matrix* ldpc_bits) , 1);</pre> <p>This line shows that the parity bits of the 802.11 LDPC codes are generated with “modulo two addition operations .” Specifically, the <code>bitand()</code> function in Matlab performs a bitwise AND operation. In the code above, a generator matrix is multiplied by the <code>ldpc_bits</code>, which are the information bits as indicated in the preceding lines of the Matlab code. The result of this multiplication is then bitwise AND’ed with 1, which means the least significant bit is maintained. Thus, the parity bit generation is performed using mod-2 summing.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 5 | |
| [5] The method of claim 1, wherein generating the sequence of parity bits comprises: | <p>The Accused Products practice the method of claim 1.</p> <p><i>See supra</i> [1]-[1c].</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|---|
| <p>[5a] generating a random sequence of bits that repeats each of the message bits one or more times with the repeats of the message bits being distributed in a random sequence, wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular; and</p> | <p>The Accused Products generate a random sequence of bits that repeats each of the message bits one or more times with the repeats of the message bits being distributed in a random sequence, wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular.</p> <p>For example, the Accused Products implement the IEEE Standards, which generate a random sequence of bits that repeats each of the message bits one or more times with the repeats of the message bits being distributed in a random sequence, wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular.</p> <p>As shown below, the parity check matrices of the LDPC codes in the 802.11 standard use a dual diagonal pattern. This dual diagonal pattern permits efficient encoding.</p> <p>The 802.11 standard defines a codeword $\mathbf{c} = (i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, with information bits i and parity bits p. The parity bits p may be determined from a parity check matrix \mathbf{H} by computing $\mathbf{H} \times \mathbf{c}^T = 0$.</p> <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c} = (i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, \mathbf{c}, of size n, $\mathbf{c} = (i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = 0$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> <p>The parity check matrix \mathbf{H} for each block size and code rate is defined in Tables R.1 to R.3 of the 802.11n-2009 amendment, and in Tables F-1 to F-3 of the 802.11-2012 standard and the 802.11-2020 standard.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|----|----|----|---|---|----|---|----|---|----|----|---|----|---|---|---|---|----|---|---|---|---|---|---|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|----|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|----|---|----|---|---|----|---|---|---|---|---|---|---|----|---|----|---|---|---|---|----|----|----|---|---|---|----|---|----|---|----|---|---|---|---|---|---|---|---|----|----|----|---|----|----|---|----|---|---|---|----|---|---|---|---|---|---|---|---|---|---|
| | <p>Table R.1 defines the matrix prototypes of the parity-check matrices for a codeword block length $n=648$ bits, with a subblock size $Z=27$ bits.</p> <p style="text-align: center;">Table R.1—Matrix prototypes for codeword block length $n=648$ bits, subblock size is $Z = 27$ bits</p> <p style="text-align: center;">* * *</p> <div style="border: 1px solid black; padding: 10px;"><p>(c) Coding rate $R = 3/4$.</p><table><tr><td>16</td><td>17</td><td>22</td><td>24</td><td>9</td><td>3</td><td>14</td><td>-</td><td>4</td><td>2</td><td>7</td><td>-</td><td>26</td><td>-</td><td>2</td><td>-</td><td>21</td><td>-</td><td>1</td><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>12</td><td>12</td><td>3</td><td>3</td><td>26</td><td>6</td><td>21</td><td>-</td><td>15</td><td>22</td><td>-</td><td>15</td><td>-</td><td>4</td><td>-</td><td>-</td><td>16</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>18</td><td>26</td><td>16</td><td>22</td><td>23</td><td>9</td><td>-</td><td>0</td><td>-</td><td>4</td><td>-</td><td>4</td><td>-</td><td>8</td><td>23</td><td>11</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td></tr><tr><td>9</td><td>7</td><td>0</td><td>1</td><td>17</td><td>-</td><td>-</td><td>7</td><td>3</td><td>-</td><td>3</td><td>23</td><td>-</td><td>16</td><td>-</td><td>-</td><td>21</td><td>-</td><td>0</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td></tr><tr><td>24</td><td>5</td><td>26</td><td>7</td><td>1</td><td>-</td><td>-</td><td>15</td><td>24</td><td>15</td><td>-</td><td>8</td><td>-</td><td>13</td><td>-</td><td>13</td><td>-</td><td>11</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td></tr><tr><td>2</td><td>2</td><td>19</td><td>14</td><td>24</td><td>1</td><td>15</td><td>19</td><td>-</td><td>21</td><td>-</td><td>2</td><td>-</td><td>24</td><td>-</td><td>3</td><td>-</td><td>2</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td></tr></table></div> <p>IEEE 802.11n-2009 at Annex R, Table R.1; <i>see also</i> IEEE 802.11-2012 at Annex F, Table F-1; IEEE 802.11-2020 at Annex F, Table F-1.</p> <p>Each of the parity-check matrices can be partitioned into square subblocks (submatrices) of size $Z \times Z$. These submatrices are either cyclic-permutations of the identity matrix or null submatrices.</p> <p>The cyclic-permutation matrix P_i is obtained from the $Z \times Z$ identity matrix by cyclically shifting the columns to the right by i elements. The matrix P_0 is the $Z \times Z$ identity matrix. Figure 20-12 illustrates examples (for a subblock size of 8×8) of cyclic-permutation matrices P_i.</p> | 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | - | 0 | 0 | - | 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | - | 0 | 0 | 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 |
| 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | - | 0 | 0 | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | - | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | $P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, P_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, P_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$ <p align="center">Figure 20-12—Examples of cyclic-permutation matrices with Z=8</p> <p>Table R.1 displays the “matrix prototypes” of parity-check matrices for all four coding rates at block length $n=648$ bits. The integer i denotes the cyclic-permutation matrix P_i, as illustrated in Figure 20-12. Vacant entries of the table denote null (zero) submatrices.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.4; <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.4; IEEE 802.11-2020 at § 19.3.11.7.4.</p> <p>The parity check matrices may also be represented as a combination of two matrices, \mathbf{H}_1 and \mathbf{H}_2, where $\mathbf{H} = [\mathbf{H}_1 \mathbf{H}_2]$, \mathbf{H}_1 is an $(n-k) \times k$ matrix, and \mathbf{H}_2 is an $(n-k) \times (n-k)$ matrix, such that $\mathbf{H}_1 \times i^T + \mathbf{H}_2 \times p^T = 0$, $i = (i_0, i_1, \dots, i_{(k-1)})$ and $p = (p_0, p_1, \dots, p_{(n-k-1)})$.</p> <p>The parity check matrices of the 802.11 standards may be used to generate Tanner Graphs that reflect the codes’ structure. The Tanner Graph representation of the 802.11 parity check matrices illustrates encoding of the message bits, in which each bit in the collection (<i>i.e.</i>, each information bit) is repeated.</p> <p>For example, in the exemplary Tanner Graph shown below for the (648, 486) code ($R = 3/4$), the dashed green rectangle illustrates the repetition of each of the message bits one or more times with the repeats of the message bits being distributed in a random sequence, wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular. The encoding takes 486</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

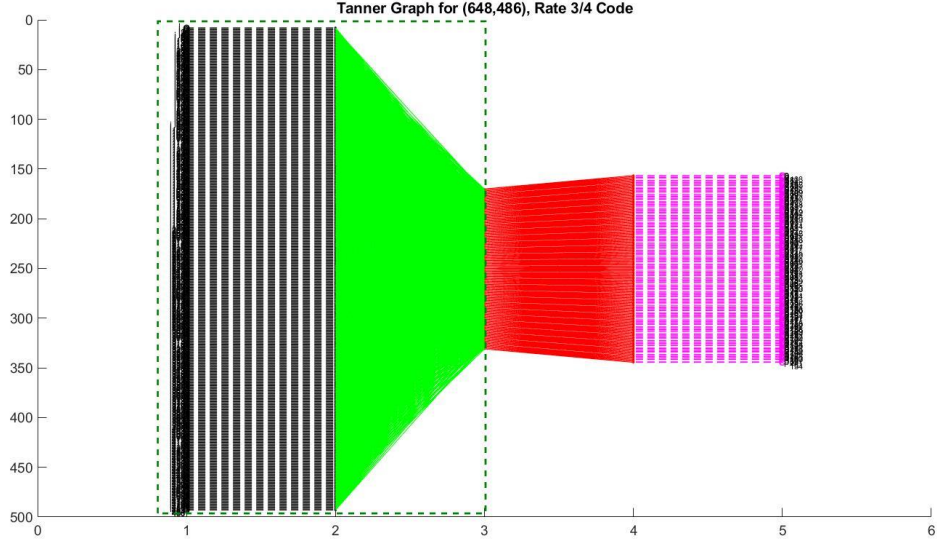
| Claim Language | Analysis |
|----------------|--|
| | <p>message bits (or “information bits”), repeats them irregularly, and sums them at check nodes to form 162 sums of irregularly repeated information bits.</p> <p>The figure is a Tanner Graph for a (648,486) code with a rate of 3/4. The vertical axis represents the number of nodes, ranging from 0 to 500. The horizontal axis represents the number of edges, ranging from 0 to 6. The graph is divided into several regions: a large green region on the left (edges 1-2), a red region in the middle (edges 3-4), and a pink region on the right (edges 5-6). The green region contains a dense grid of black lines representing the connections between nodes and edges. The red and pink regions show a more sparse, irregular pattern of connections. The graph is enclosed in a dashed green box.</p> <p><i>See also</i> Appendix B (showing exemplary Tanner Graph representation for all 12 parity check matrices defined in the 802.11 standards).</p> <p>Zooming in on the left-hand portion of the exemplary Tanner Graph shows that the message (or information) bits are inputs to the encoding:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

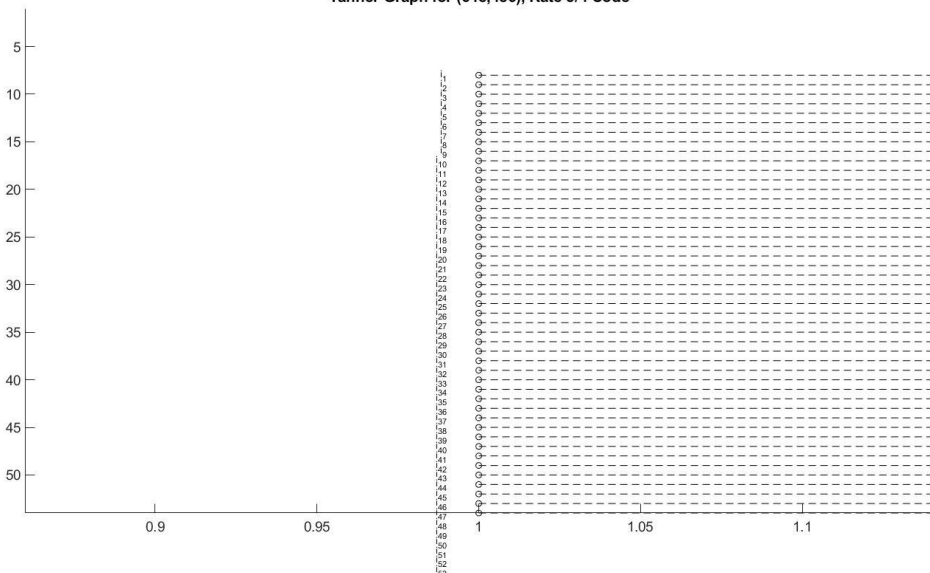
| Claim Language | Analysis |
|----------------|--|
| | <p>Tanner Graph for (648,486), Rate 3/4 Code</p>  <p><i>See also Appendix B.</i></p> <p>The encoding repeats each message bit one or more times, reflected by the green line segments in the exemplary Tanner Graph, as shown below:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

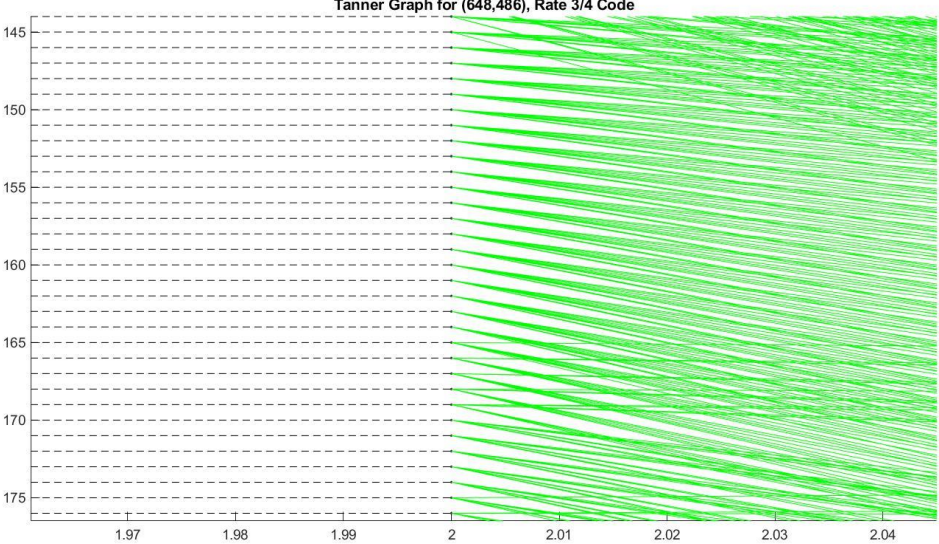
| Claim Language | Analysis |
|----------------|--|
| | <div><p>Tanner Graph for (648,486), Rate 3/4 Code</p></div> <p><i>See also Appendix B.</i></p> <p>The plurality of green line segments connected to each message bit confirms that each bit is repeated one or more times. Different fractions of the message bits are each repeated a different number of times, and the number of repeats for each message bit is irregular, as shown below:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>Tanner Graph for (648,486), Rate 3/4 Code</p> <p>Information bits repeated 6 times</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

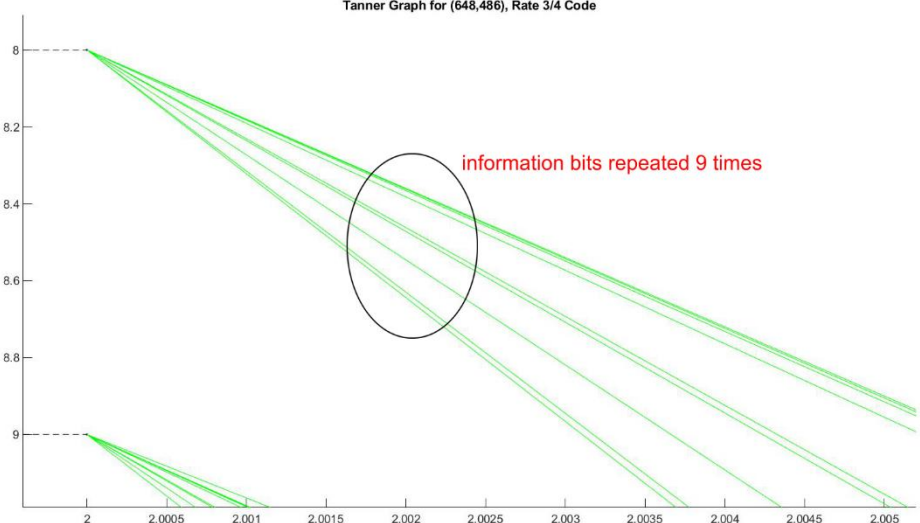
| Claim Language | Analysis |
|----------------|--|
| | <p>Tanner Graph for (648,486), Rate 3/4 Code</p>  <p>The repeats of the message bits are distributed in a random sequence:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

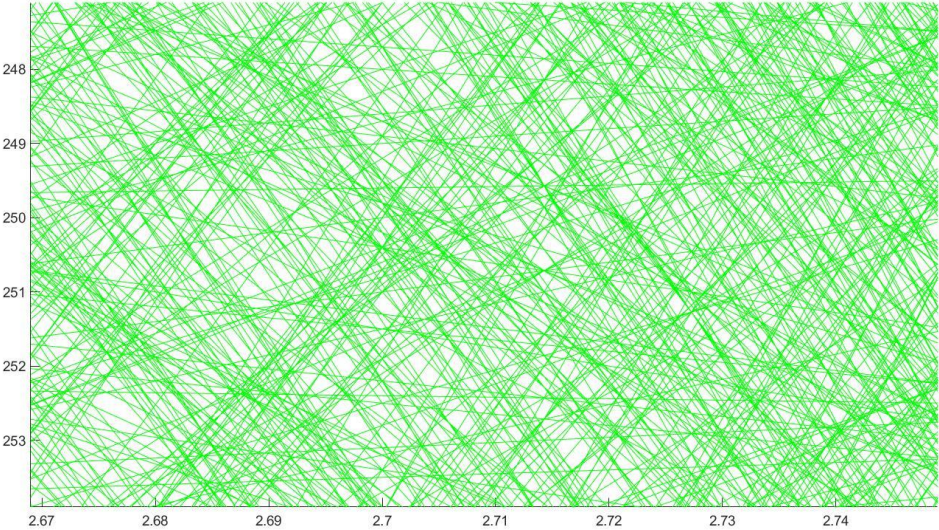
| Claim Language | Analysis |
|----------------|---|
| | <div><p>Tanner Graph for (648,486), Rate 3/4 Code</p></div> <p><i>See also Appendix B.</i></p> <p>As shown by their corresponding exemplary Tanner Graphs, each of the eleven other parity check matrices provided in the 802.11 standard—(1944, 974); (1296, 648); (648, 324); (1944, 1296); (1296, 864); (648, 432); (1944, 1458); (1296, 972); (1944, 1620); (1296, 1080); and (648,540)—define a code whose encoding involves generating a random sequence of bits that repeats each of the message bits one or more times with the repeats of the message bits being distributed in a random sequence, wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular. <i>See Appendix B.</i></p> <p>Further, in each of the twelve parity check matrices provided in the 802.11 standard, different fractions of the message bits are each repeated a different number of times. An example of two such bits for the (648, 486) code is shown above. More generally, the exemplary Tanner Graphs can be used to determine the number of times each message bit</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | |
|-------------------|---|-------------------|-----------------------------|---|----|---|----|---|-----|---|----|---|----|---|----|----|----|
| | <p>is repeated. For example, as illustrated in the exemplary Tanner Graph for the (648, 486) code, message bits may be repeated with the following frequencies:</p> <table><tr><th>Number of repeats</th><th>Number of such message bits</th></tr><tr><td>3</td><td>27</td></tr><tr><td>5</td><td>54</td></tr><tr><td>6</td><td>135</td></tr><tr><td>7</td><td>81</td></tr><tr><td>8</td><td>54</td></tr><tr><td>9</td><td>54</td></tr><tr><td>11</td><td>81</td></tr></table> <p>These frequencies are illustrative only; to the extent the Accused Products use alternative frequencies, they literally infringe so long as each information bit is repeated and at least two information bits are repeated a different number of times, even if the exact frequencies differ from those above.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> <p>For example, to the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe because they do not “repeat” information bits, <i>e.g.</i>, by construing “repeat” to require copies of information bits to be stored in a separate memory, the Accused Products would infringe under the doctrine of equivalents because each information bit is used multiple times in the first encoding, and appears in a plurality of sums of information bits at the check nodes. Whether or not the information bits are duplicated and separately stored in memory before they are summed or combined is an insubstantial difference: both scenarios accomplish substantially the same function of summing pluralities of information bits at the check nodes, in substantially the same way, <i>i.e.</i>, by directing each information bit to a plurality of check nodes to be</p> | Number of repeats | Number of such message bits | 3 | 27 | 5 | 54 | 6 | 135 | 7 | 81 | 8 | 54 | 9 | 54 | 11 | 81 |
| Number of repeats | Number of such message bits | | | | | | | | | | | | | | | | |
| 3 | 27 | | | | | | | | | | | | | | | | |
| 5 | 54 | | | | | | | | | | | | | | | | |
| 6 | 135 | | | | | | | | | | | | | | | | |
| 7 | 81 | | | | | | | | | | | | | | | | |
| 8 | 54 | | | | | | | | | | | | | | | | |
| 9 | 54 | | | | | | | | | | | | | | | | |
| 11 | 81 | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|--|
| | summed or combined, and accomplish substantially the same result of forming the first encoded data block. |
| <p>[5b] XOR summing in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of bits.</p> | <p>The Accused Products XOR sum in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of bits.</p> <p>For example, the Accused Products implement the IEEE Standards, which XOR sum in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of bits.</p> <p>As noted above in [4], the calculation of the parity bits per claim 1 is recursive, as required by:</p> $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$ <p>This recursive equation forms a linear sum of a predecessor parity bit, x_{j-1}, and “a” bits of the random sequence of bits. Further, as noted in [4] above, arithmetic in the 802.11 LDPC codes is performed using modulo two arithmetic including modulo two addition or its mathematical equivalent, XOR summing.</p> <p>In the example Tanner Graph for the (648, 486) code in [5a] above, “a” bits of the random sequence of bits are XOR summed at each check node, as illustrated by the green lines in the figure below. These “a” bits of the random sequence of bits are XOR summed in a linear fashion with a predecessor parity bit, as illustrated by the red line on the right of the figure below.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

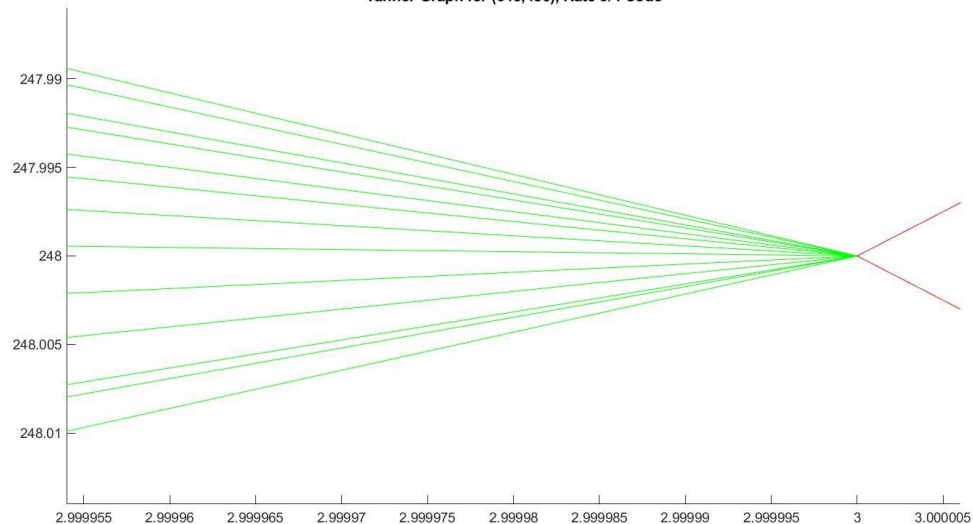
| Claim Language | Analysis |
|--|---|
| | <p align="center">Tanner Graph for (648,486), Rate 3/4 Code</p>  <p><i>See also</i> Appendix B.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 6 | |
| [6] The method of claim 5, wherein generating the random sequence of bits comprises coding the collection of | <p>The Accused Products practice the method of claim 5.</p> <p>See claim [5]-[5b].</p> <p>In the Accused Products, generating the random sequence of bits comprises coding the collection of message bits using a low-density generator matrix (LDGM) coder.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|---|
| message bits using a low-density generator matrix (LDGM) coder. | <p>For example, the Accused Products implement the IEEE Standards, and in said standards generating the random sequence of bits comprises coding the collection of message bits using a low-density generator matrix (LDGM) coder.</p> <p>As shown in [5a], the generation of the random sequence of bits can be represented using a Tanner Graph. This generation can also be described using a low density generator matrix (LDGM). An exemplary LDGM for the (648, 486) LDPC code ($R=3/4$) is shown below. The matrix is 486 by 162. Dots are used to show the locations in the LGDM that are equal to one. All other locations are equal to zero. Other LDGMs with the message bits (inputs) and check nodes (outputs) in a different order may also be used, and would satisfy the claim limitation.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

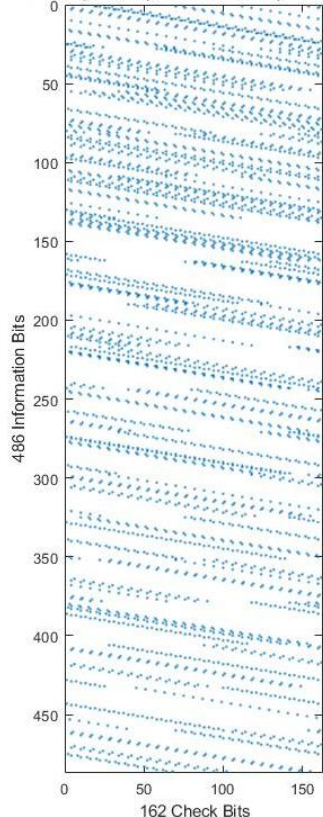
| Claim Language | Analysis | | | | | | |
|----------------|---|-------------|---------|--------|---------|--------|---------|
| | <div><p>LDGM for (648,486), Rate 3/4 Code (Density 4.5%)</p><p>Analysis of the 12 parity check matrices defined in the 802.11 standard shows the generator matrix used in connection with generating the random sequence of bits is low density. The following are the densities of exemplary LDGMs generated from the 12 parity check matrices defined in the 802.11 standard:</p><table><tr><th>LDGM Number</th><th>Density</th></tr><tr><td>LDGM 1</td><td>1.6204%</td></tr><tr><td>LDGM 2</td><td>2.0544%</td></tr></table></div> | LDGM Number | Density | LDGM 1 | 1.6204% | LDGM 2 | 2.0544% |
| LDGM Number | Density | | | | | | |
| LDGM 1 | 1.6204% | | | | | | |
| LDGM 2 | 2.0544% | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | |
|----------------|---|--------|--------|--------|---------|--------|----------|--------|---------|--------|--------|--------|---------|--------|----------|---------|---------|---------|----------|---------|---------|
| | <table border="1" data-bbox="1134 289 1543 667"> <tr><td>LDGM 3</td><td>2.572%</td></tr> <tr><td>LDGM 4</td><td>3.6574%</td></tr> <tr><td>LDGM 5</td><td>0.78447%</td></tr> <tr><td>LDGM 6</td><td>1.0272%</td></tr> <tr><td>LDGM 7</td><td>1.286%</td></tr> <tr><td>LDGM 8</td><td>1.7593%</td></tr> <tr><td>LDGM 9</td><td>0.52298%</td></tr> <tr><td>LDGM 10</td><td>0.6848%</td></tr> <tr><td>LDGM 11</td><td>0.82305%</td></tr> <tr><td>LDGM 12</td><td>1.0802%</td></tr> </table> <p><i>See also</i> Appendix B (showing corresponding exemplary LDGMs for all 12 parity check matrices defined in the 802.11 standard).</p> <p>An LDGM may sum “a” bits of the random sequence of bits, as in the above example, before accumulation with a predecessor parity bit. Or, alternatively, an LDGM may only irregularly repeat message bits and distribute in random sequence to generate the random sequence of bits as described in [5a]. In either case, the LDGM is used to generate the random sequence of bits.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> <p>For example, to the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe because they do not generate the random sequence of bits via a low density generator matrix (“LDGM”) coder, the accused products would infringe under the doctrine of equivalents because they generate parity bits by summing subsets of message bits.</p> <p>An LDGM is a way to represent a linear transform. It is known in the art that such transforms may be implemented in various ways. As an example, these implementations</p> | LDGM 3 | 2.572% | LDGM 4 | 3.6574% | LDGM 5 | 0.78447% | LDGM 6 | 1.0272% | LDGM 7 | 1.286% | LDGM 8 | 1.7593% | LDGM 9 | 0.52298% | LDGM 10 | 0.6848% | LDGM 11 | 0.82305% | LDGM 12 | 1.0802% |
| LDGM 3 | 2.572% | | | | | | | | | | | | | | | | | | | | |
| LDGM 4 | 3.6574% | | | | | | | | | | | | | | | | | | | | |
| LDGM 5 | 0.78447% | | | | | | | | | | | | | | | | | | | | |
| LDGM 6 | 1.0272% | | | | | | | | | | | | | | | | | | | | |
| LDGM 7 | 1.286% | | | | | | | | | | | | | | | | | | | | |
| LDGM 8 | 1.7593% | | | | | | | | | | | | | | | | | | | | |
| LDGM 9 | 0.52298% | | | | | | | | | | | | | | | | | | | | |
| LDGM 10 | 0.6848% | | | | | | | | | | | | | | | | | | | | |
| LDGM 11 | 0.82305% | | | | | | | | | | | | | | | | | | | | |
| LDGM 12 | 1.0802% | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|--|
| | <p>may represent the information in the LDGM via equations, physical connections to adders, arrays in memory and/or by other means. The LDGM may be expressed in alternative formats such as equations. Each column of the LDGM matrix specifies the message bits that must be repeated irregularly to form the random sequence of bits. Thus, an equation can be written for each column of the LDGM. That set of equations is an alternative and equivalent representation of the LDGM. The equations may be used directly or manipulated form to implement the transform specified by the LDGM. Information from the LDGM may be stored in forms such as equations, arrays or similar storage structures. Implementations may use repetition and/or summation of message bits in groups as specified by the LDGM (or alternative representations thereof). Looping may be used. For example, loops may operate over columns and/or rows of the LDGM. Implementations may use hard-coded operations thereby requiring little or no storage of the LDGM or information in memories. Alternatively, implementations may store the LDGM or information derived from the LDGM in memory thereby making the information available in whole or in part as the random sequence of bits is generated. Hybrid structure are also possible. Further, the generation of the random sequence of bits may be carried out using or assisted by custom or general purpose processors.</p> <p>Thus, whether or not the generation of the random sequence of bits is performed using an LDGM directly or using an equivalent hardware and/or software implementation is an insubstantial difference: each scenario accomplishes substantially the same function of generating a sum of message bits, in substantially the same way, <i>i.e.</i>, summing subsets of a plurality of message bits, and accomplishes substantially the same result of providing an efficient and effective linear error correction code.</p> |
| Claim 7 | |
| [7] The method of claim 5, wherein generating the random sequence of bits comprises: | The Accused Products practice the method of claim 5. See claim [5]-[5b]. |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|---|
| <p>[7a] producing a block of data bits, wherein different message bits are each repeated a different number of times in a sequence that matches the first sequence; and</p> | <p>The Accused Products produce a block of data bits, wherein different message bits are each repeated a different number of times in a sequence that matches the first sequence.</p> <p>For example, the Accused Products implement the IEEE Standards, which produce a block of data bits, wherein different message bits are each repeated a different number of times in a sequence that matches the first sequence.</p> <p>As explained in [5a] above, the parity check matrices of the 802.11 standards may be used to generate Tanner Graphs that reflect the codes' structure. The Tanner Graph representation of the 802.11 parity check matrices illustrates the production of a block of data bits in which different message bits are each repeated a different number of times in a sequence that matches the first sequence.</p> <p>For example, in the exemplary Tanner Graph shown below for the (648, 486) code ($R = 3/4$), the dashed green rectangle illustrates the repetition of each of the message bits a different number of times in a sequence that matches the first sequence.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

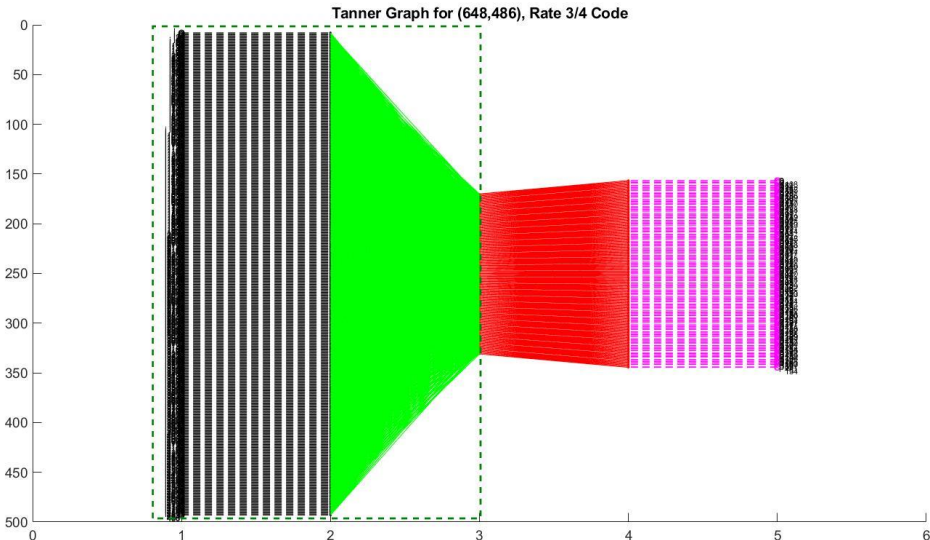
| Claim Language | Analysis |
|----------------|--|
| | <div><p>Tanner Graph for (648,486), Rate 3/4 Code</p><p>The plurality of green line segments connected to each message bit confirms that different message bits are each repeated a different number of times, as shown in the exemplary Tanner graph below:</p></div> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>Tanner Graph for (648,486), Rate 3/4 Code</p> <p>Information bits repeated 6 times</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

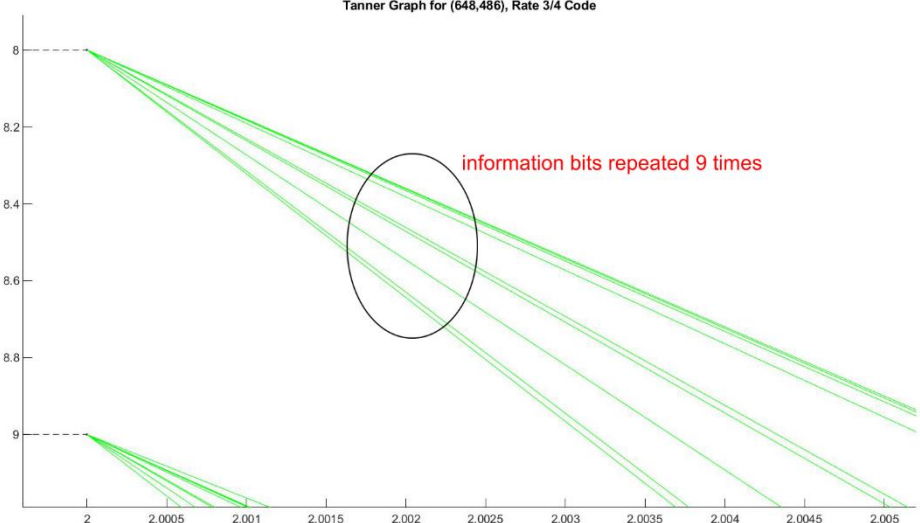
| Claim Language | Analysis |
|----------------|--|
| | <div><p>Tanner Graph for (648,486), Rate 3/4 Code</p><p>information bits repeated 9 times</p></div> <p>Further, in each of the twelve parity check matrices provided in the 802.11 standard, different fractions of the message bits are each repeated a different number of times. An example of two such bits for the (648, 486) code is shown above. More generally, the Tanner Graphs can be used to determine the number of times each message bit is repeated. For example, as illustrated in the exemplary Tanner Graph for the (648, 486) code, message bits may be repeated with the following frequencies:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | |
|-------------------|---|--|-------------------|-----------------------------|---|----|---|----|---|-----|---|----|---|----|---|----|----|----|
| | <table><tr><th>Number of repeats</th><th>Number of such message bits</th></tr><tr><td>3</td><td>27</td></tr><tr><td>5</td><td>54</td></tr><tr><td>6</td><td>135</td></tr><tr><td>7</td><td>81</td></tr><tr><td>8</td><td>54</td></tr><tr><td>9</td><td>54</td></tr><tr><td>11</td><td>81</td></tr></table> | | Number of repeats | Number of such message bits | 3 | 27 | 5 | 54 | 6 | 135 | 7 | 81 | 8 | 54 | 9 | 54 | 11 | 81 |
| Number of repeats | Number of such message bits | | | | | | | | | | | | | | | | | |
| 3 | 27 | | | | | | | | | | | | | | | | | |
| 5 | 54 | | | | | | | | | | | | | | | | | |
| 6 | 135 | | | | | | | | | | | | | | | | | |
| 7 | 81 | | | | | | | | | | | | | | | | | |
| 8 | 54 | | | | | | | | | | | | | | | | | |
| 9 | 54 | | | | | | | | | | | | | | | | | |
| 11 | 81 | | | | | | | | | | | | | | | | | |
| | <p>These frequencies are illustrative only; to the extent the Accused Products use alternative frequencies, they literally infringe so long as different message bits are each repeated a different number of times, even if the exact frequencies differ from those above.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> <p>For example, to the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe because they do not “repeat” information bits, <i>e.g.</i>, by construing “repeat” to require copies of information bits to be stored in a separate memory, the Accused Products would infringe under the doctrine of equivalents because each information bit is used multiple times in the first encoding, and appears in a plurality of sums of information bits at the check nodes. Whether or not the information bits are duplicated and separately stored in memory before they are summed or combined is an insubstantial difference: both scenarios accomplish substantially the same function of summing pluralities of information bits at the check nodes, in substantially the same way, <i>i.e.</i>, by directing each information bit to a plurality of check nodes to be summed or combined, and accomplish substantially the same result of forming the first encoded data block.</p> | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

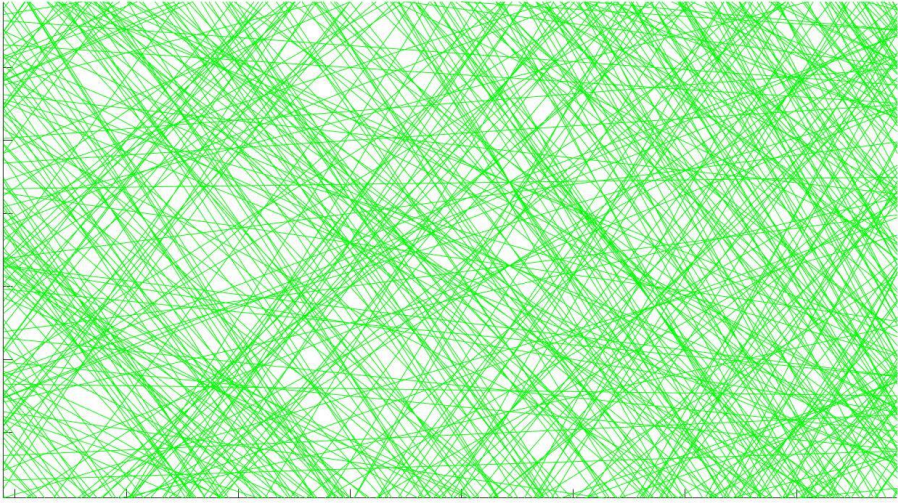
| Claim Language | Analysis |
|---|---|
| <p>[7b] randomly permuting the different bits to generate the random sequence.</p> | <p>The Accused Products randomly permute the different bits to generate the random sequence.</p> <p>For example, the Accused Products implement the IEEE Standards, which on information and belief includes randomly permuting the different bits to generate the random sequence.</p> <p>As explained in [7a] above, the parity check matrices of the 802.11 standards may be used to generate Tanner Graphs that reflect the codes' structure. The Tanner Graph representation of the 802.11 parity check matrices illustrates randomly permuting the different bits to generate the random sequence.</p> <p>For example, in the zoomed-in version of the exemplary Tanner Graph shown below for the (648, 486) code ($R = 3/4$), the green lines illustrate randomly permuting the different bits to generate the random sequence.</p> <div data-bbox="877 792 1803 1331"><p style="text-align: center;">Tanner Graph for (648,486), Rate 3/4 Code</p></div> <p><i>See also Appendix B.</i></p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|---|
| | <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 8 | |
| <p>[8] The method of claim 1, further comprising transmitting the sequence of parity bits.</p> | <p>The Accused Products practice the method of claim 1.</p> <p>See <i>supra</i> [1]-[1c].</p> <p>The Accused Products transmit the sequence of parity bits.</p> <p>For example, the Accused Products implement the IEEE Standards, which transmit the sequence of parity bits.</p> <p>For example, the Accused Products transmit a codeword “c” of size “n” that includes “k” information bits and “n-k” parity bits in a transmission data stream.</p> <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $c=(i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, c, of size n, $c=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $H \times c^T = 0$, where H is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>20.3.11.6.6 LDPC parser</p> <p>The succession of LDPC codewords that result from the encoding process of 20.3.11.6.5 shall be converted into a bitstream in sequential fashion. Within each codeword, bit i_0 is ordered first. The parsing of this encoded data stream into spatial streams shall follow exactly the parsing rules defined for the BCC encoder, as defined in 20.3.11.7.1. However, the frequency interleaver of 20.3.11.7.3 is bypassed.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.6 (emphasis added); <i>see also</i> IEEE802.11-2012 at § 20.3.11.7.6; IEEE 802.11-2020 at § 19.3.11.7.6.</p> <p>Figure 20-13—LDPC PDU encoding padding and puncturing of a single codeword</p> <p>IEEE 802.11n-2009 at Figure 20-13; <i>see also</i> IEEE 802.11-2012 at Figure 20-13; IEEE 802.11-2020 at Figure 19-13.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

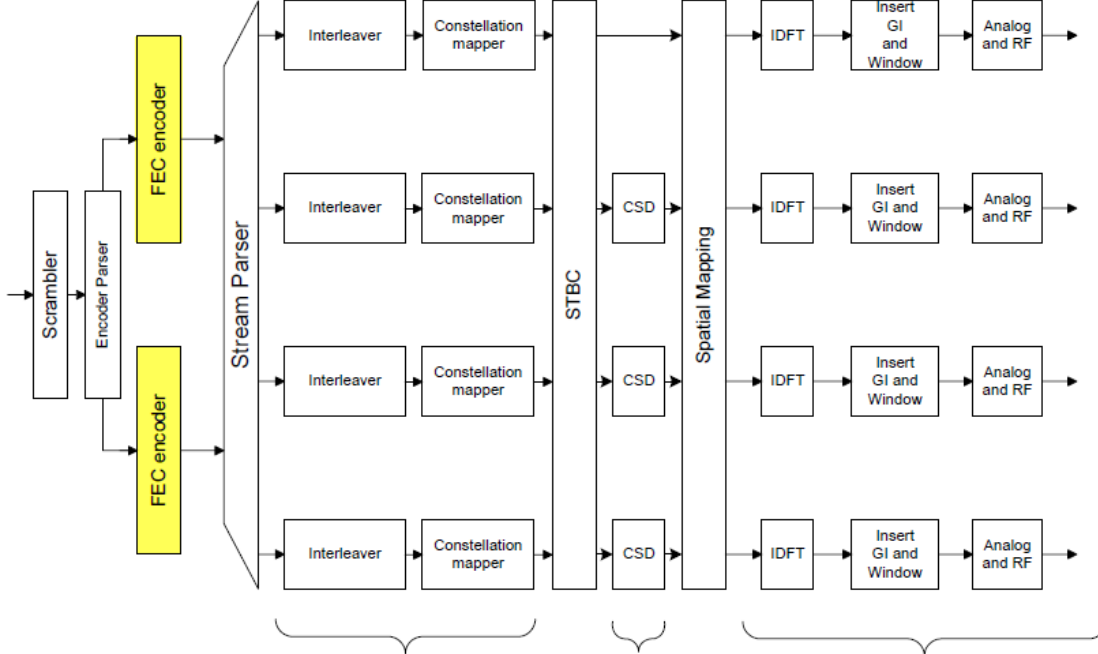
| Claim Language | Analysis |
|----------------|--|
| |  <p>The diagram illustrates a transmitter system architecture. It begins with an input signal entering a 'Scrambler' block. The output of the scrambler goes to an 'Encoder Parser' block. From the encoder parser, the signal is split into two parallel paths, each passing through a 'FEC encoder' block (highlighted in yellow). The outputs of these encoders are then fed into a 'Stream Parser' block. The stream parser outputs four parallel streams, each consisting of an 'Interleaver' block followed by a 'Constellation mapper' block. These four streams are then processed by a 'STBC' (Space-Time Block Coding) block. The output of the STBC block is then fed into a 'Spatial Mapping' block. The output of the spatial mapping block is then processed by four parallel 'IDFT' blocks. The outputs of the IDFT blocks are then processed by four parallel 'Insert GI and Window' blocks. Finally, the outputs of these blocks are fed into four parallel 'Analog and RF' blocks, which produce the final output signals.</p> <p style="text-align: center;"> N_{SS} Spatial Streams N_{STB} Space Time Streams N_{TX} Transmit Chains </p> <p>NOTES</p> <ul style="list-style-type: none"> —There may be 1 or 2 FEC encoders when BCC encoding is used. —The stream parser may have 1, 2, 3 or 4 outputs. —When LDPC encoding is used, the interleavers are not used —When STBC is used, the STBC block has more outputs than inputs. —When spatial mapping is used, there may be more transmit chains than space time streams. —The number of inputs to the spatial mapper may be 1, 2, 3, or 4. <p>IEEE 802.11n-2009 at Fig. 20-3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at Fig. 20-3; IEEE 802.11-2020 at Fig. 19-3.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|--|
| Claim 10 | |
| <p>[10] The method of claim 8, wherein transmitting the sequence of parity bits comprises transmitting the sequence of parity bits as part of a systematic code.</p> | <p>The Accused Products practice the method of claim 8.</p> <p>See claim [8].</p> <p>In the Accused Products, transmitting the sequence of parity bits comprises transmitting the sequence of parity bits as part of a systematic code.</p> <p>For example, the Accused Products implement the IEEE Standards, and in said standards transmitting the sequence of parity bits comprises transmitting the sequence of parity bits as part of a systematic code.</p> <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $c=(i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, c, of size n, $c=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $H \times c^T = 0$, where H is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 11 | |
| <p>[11] A device comprising:</p> | <p>To the extent this preamble is construed to be limiting, the Accused Products are devices.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

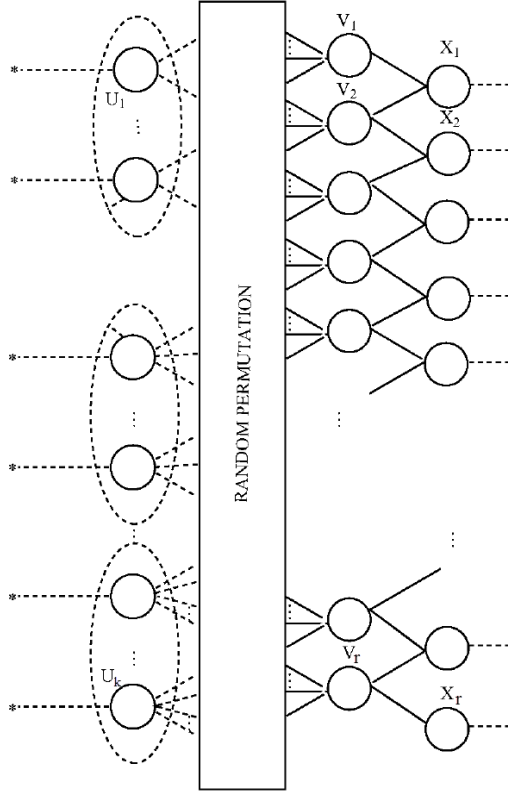
| Claim Language | Analysis |
|---|---|
| | <p>Further, each Accused Product is a device capable of encoding a signal.</p> <p>See claim [1].</p> |
| <p>[11a] an encoder configured to receive a collection of message bits and encode the message bits to generate a collection of parity bits in accordance with the following Tanner graph:</p>  | <p>The Accused Products include an encoder configured to receive a collection of message bits.</p> <p>See claim [1a].</p> <p>The Accused Products include an encoder configured to encode the message bits to generate a collection of parity bits in accordance with the claimed Tanner graph.</p> <p>For example, the Accused Products implement the IEEE Standards, and such implementation and/or compliance includes an encoder configured to encode the message bits to generate a collection of parity bits in accordance with the claimed Tanner graph.</p> <p>As shown below, the parity check matrices of the LDPC codes in the 802.11 standard use a dual diagonal pattern. This dual diagonal pattern permits efficient encoding.</p> <p>The 802.11 standard defines a codeword $\mathbf{c} = (i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, with information bits i and parity bits p. The parity bits p may be determined from a parity check matrix \mathbf{H} by computing $\mathbf{H} \times \mathbf{c}^T = 0$.</p> <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c} = (i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, \mathbf{c}, of size n, $\mathbf{c} = (i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = 0$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|----|----|----|---|---|----|---|----|---|----|----|---|----|---|---|---|---|----|---|---|---|---|---|---|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|----|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|----|---|----|---|---|----|---|---|---|---|---|---|---|----|---|----|---|---|---|---|----|----|----|---|---|---|----|---|----|---|----|---|---|---|---|---|---|---|---|----|----|----|---|----|----|---|----|---|---|---|----|---|---|---|---|---|---|---|---|---|---|
| | <p>The parity check matrix H for each block size and code rate is defined in Tables R.1 to R.3 of the 802.11n-2009 amendment, and in Tables F-1 to F-3 of the 802.11-2012 standard and the 802.11-2020 standard.</p> <p>Table R.1 defines the matrix prototypes of the parity-check matrices for a codeword block length $n=648$ bits, with a subblock size $Z=27$ bits.</p> <p style="text-align: center;">Table R.1—Matrix prototypes for codeword block length $n=648$ bits, subblock size is $Z = 27$ bits</p> <p style="text-align: center;">* * *</p> <div style="border: 1px solid black; padding: 5px;"><p>(c) Coding rate $R = 3/4$.</p><table><tr><td>16</td><td>17</td><td>22</td><td>24</td><td>9</td><td>3</td><td>14</td><td>-</td><td>4</td><td>2</td><td>7</td><td>-</td><td>26</td><td>-</td><td>2</td><td>-</td><td>21</td><td>-</td><td>1</td><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>12</td><td>12</td><td>3</td><td>3</td><td>26</td><td>6</td><td>21</td><td>-</td><td>15</td><td>22</td><td>-</td><td>15</td><td>-</td><td>4</td><td>-</td><td>-</td><td>16</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>18</td><td>26</td><td>16</td><td>22</td><td>23</td><td>9</td><td>-</td><td>0</td><td>-</td><td>4</td><td>-</td><td>4</td><td>-</td><td>8</td><td>23</td><td>11</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td></tr><tr><td>9</td><td>7</td><td>0</td><td>1</td><td>17</td><td>-</td><td>-</td><td>7</td><td>3</td><td>-</td><td>3</td><td>23</td><td>-</td><td>16</td><td>-</td><td>-</td><td>21</td><td>-</td><td>0</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td></tr><tr><td>24</td><td>5</td><td>26</td><td>7</td><td>1</td><td>-</td><td>-</td><td>15</td><td>24</td><td>15</td><td>-</td><td>8</td><td>-</td><td>13</td><td>-</td><td>13</td><td>-</td><td>11</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td></tr><tr><td>2</td><td>2</td><td>19</td><td>14</td><td>24</td><td>1</td><td>15</td><td>19</td><td>-</td><td>21</td><td>-</td><td>2</td><td>-</td><td>24</td><td>-</td><td>3</td><td>-</td><td>2</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td></tr></table></div> <p>IEEE 802.11n-2009 at Annex R, Table R.1; <i>see also</i> IEEE 802.11-2012 at Annex F, Table F-1; IEEE 802.11-2020 at Annex F, Table F-1.</p> <p>Each of the parity-check matrices can be partitioned into square subblocks (submatrices) of size $Z \times Z$. These submatrices are either cyclic-permutations of the identity matrix or null submatrices.</p> <p>The cyclic-permutation matrix P_i is obtained from the $Z \times Z$ identity matrix by cyclically shifting the columns to the right by i elements. The matrix P_0 is the $Z \times Z$ identity matrix. Figure 20-12 illustrates examples (for a subblock size of 8×8) of cyclic-permutation matrices P_i.</p> | 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | - | 0 | 0 | - | 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | - | 0 | 0 | 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 |
| 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | - | 0 | 0 | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | - | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | $P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, P_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, P_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$ <p align="center">Figure 20-12—Examples of cyclic-permutation matrices with Z=8</p> <p>Table R.1 displays the “matrix prototypes” of parity-check matrices for all four coding rates at block length $n=648$ bits. The integer i denotes the cyclic-permutation matrix P_i, as illustrated in Figure 20-12. Vacant entries of the table denote null (zero) submatrices.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.4; <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.4; IEEE 802.11-2020 at § 19.3.11.7.4.</p> <p>The parity check matrices may also be represented as a combination of two matrices, \mathbf{H}_1 and \mathbf{H}_2, where $\mathbf{H} = [\mathbf{H}_1 \mathbf{H}_2]$, \mathbf{H}_1 is an $(n-k) \times k$ matrix, and \mathbf{H}_2 is an $(n-k) \times (n-k)$ matrix, such that $\mathbf{H}_1 \times i^T + \mathbf{H}_2 \times p^T = 0$, $i = (i_0, i_1, \dots, i_{(k-1)})$ and $p = (p_0, p_1, \dots, p_{(n-k-1)})$.</p> <p>The parity check matrices of the 802.11 standards may be used to generate Tanner Graphs that reflect the codes’ structure. The Tanner Graph representation of the 802.11 parity check matrices illustrates encoding of the message bits in the collection.</p> <p>For example, in the exemplary Tanner Graph shown below for the (648, 486) code ($R = 3/4$) meets this limitation.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

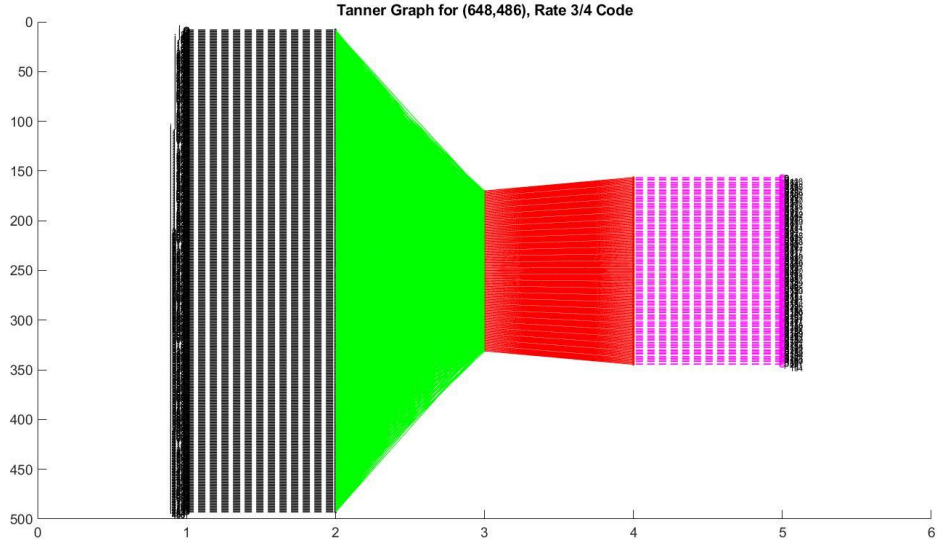
| Claim Language | Analysis |
|----------------|--|
| | <div><p>Tanner Graph for (648,486), Rate 3/4 Code</p><p>The graph shows a sequence of bit positions from 0 to 500 on the y-axis and parity check matrices from 1 to 6 on the x-axis. A black region on the left (x=1) represents message bits. A green trapezoidal region (x=2) represents the LDPC encoder's internal structure. A red trapezoidal region (x=3) represents the LDPC decoder's internal structure. A magenta region (x=4) represents the parity check matrices. The regions are connected by lines, showing the flow of information through the LDPC code.</p></div> <p><i>See also</i> Appendix B (showing exemplary Tanner Graph representation for all 12 parity check matrices defined in the 802.11 standards).</p> <p>The area on the left of the figure above, in black, represents the message bits that are received by the LDPC encoder. The message bits are repeated irregularly, as illustrated by the following zoomed-in versions of the same exemplary Tanner Graph:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

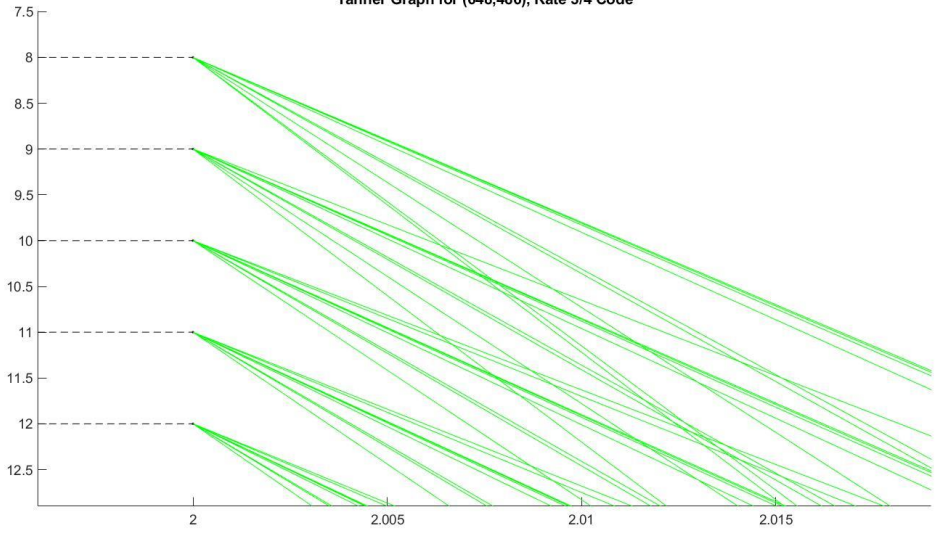
| Claim Language | Analysis |
|----------------|---|
| | <p>Tanner Graph for (648,486), Rate 3/4 Code</p>  |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

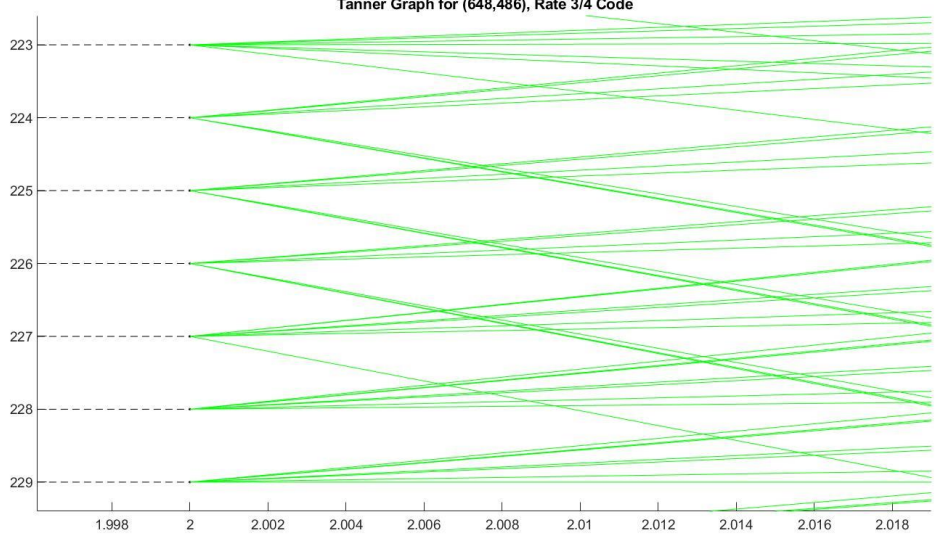
| Claim Language | Analysis |
|----------------|---|
| | <p>Tanner Graph for (648,486), Rate 3/4 Code</p>  |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

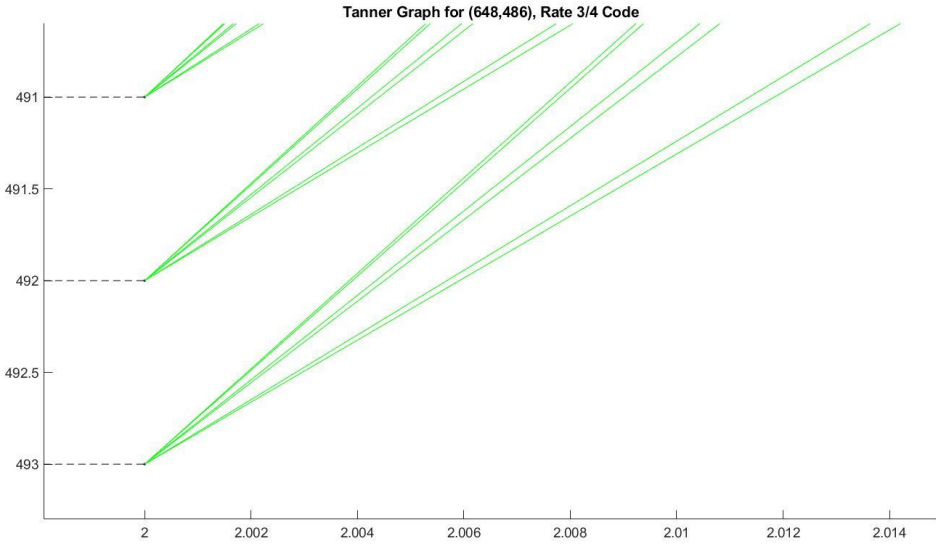
| Claim Language | Analysis |
|----------------|--|
| | <div><p>Tanner Graph for (648,486), Rate 3/4 Code</p><p>Further, the green area shown in the zoomed-in version of the same exemplary Tanner Graph illustrates a random permutation of the repeated message bits:</p></div> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

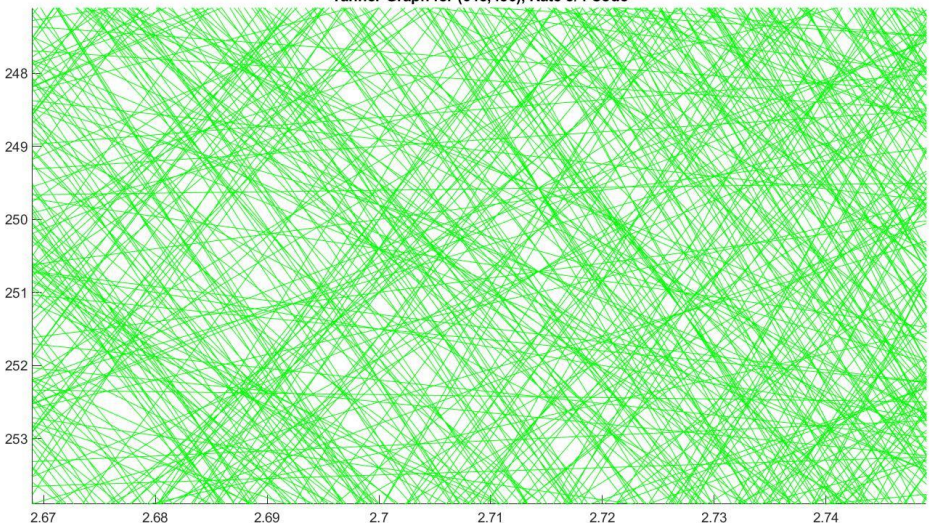
| Claim Language | Analysis |
|----------------|--|
| | <div><p>Tanner Graph for (648,486), Rate 3/4 Code</p><p>The red edges between the nodes in the exemplary Tanner Graph corresponding to the nodes marked with “x” and “v” in the claim language illustrate accumulation functionality performed by an accumulator. Zoomed-in examples of the same exemplary Tanner Graph are shown below:</p></div> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

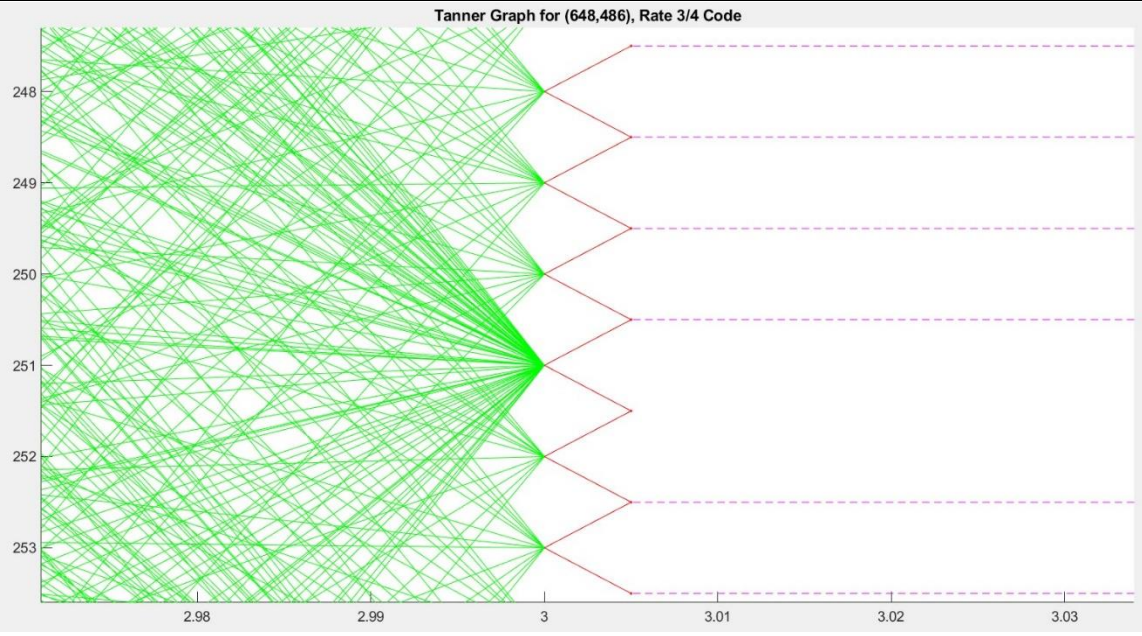
| Claim Language | Analysis |
|----------------|---|
| | <p>Tanner Graph for (648,486), Rate 3/4 Code</p>  |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

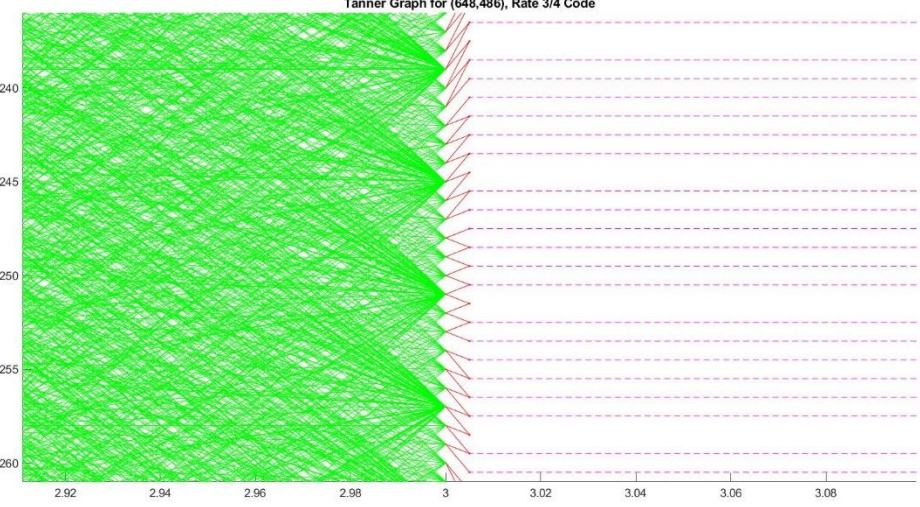
| Claim Language | Analysis |
|----------------|---|
| | <p>Tanner Graph for (648,486), Rate 3/4 Code</p>  |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

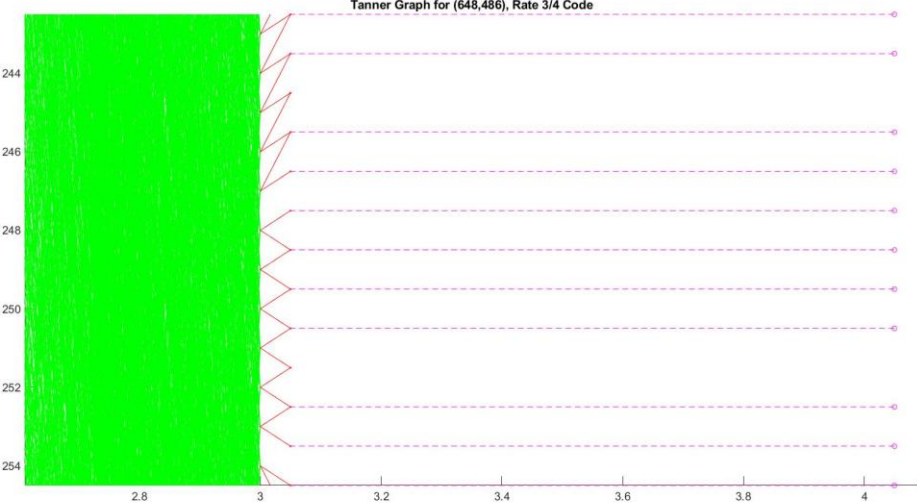
| Claim Language | Analysis |
|----------------|---|
| | <div></div> <p>The collection of parity bits generated from the LDPC encoder corresponding to the exemplary Tanner Graph above is illustrated on the right side of the figure zoomed-in versions of the exemplary Tanner Graph shown below:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <div><p>Tanner Graph for (648,486), Rate 3/4 Code</p></div> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

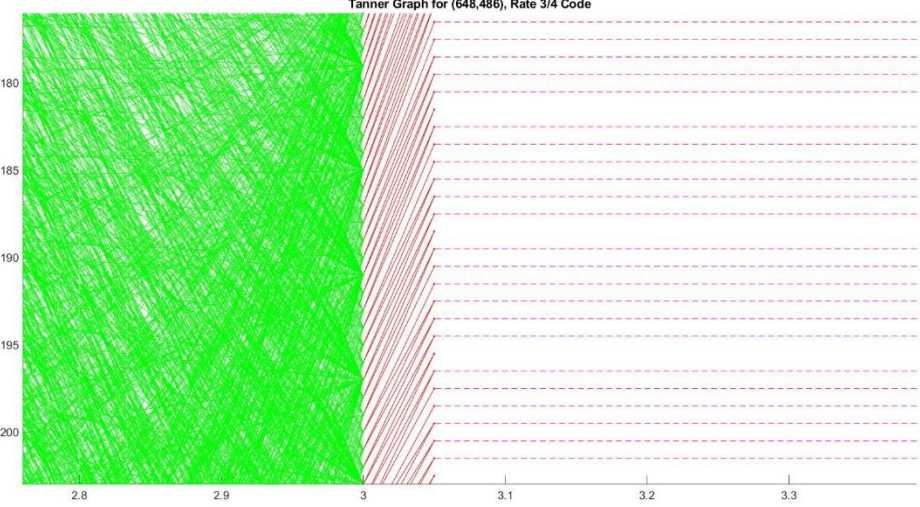
| Claim Language | Analysis |
|----------------|---|
| | <p>Tanner Graph for (648,486), Rate 3/4 Code</p>  |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|---|
| | <div data-bbox="877 310 1801 816"> <p align="center">Tanner Graph for (648,486), Rate 3/4 Code</p> </div> <p>As shown by their corresponding exemplary Tanner Graphs, each of the eleven other parity check matrices provided in the 802.11 standard—(1944, 974); (1296, 648); (648, 324); (1944, 1296); (1296, 864); (648, 432); (1944, 1458); (1296, 972); (1944, 1620); (1296, 1080); and (648,540)—define a code whose encoding involves generating a collection of parity bits in accordance with the Tanner graph shown above. <i>See</i> Appendix B.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 12 | |
| [12] The device of claim 11, wherein the encoder is configured to generate | The Accused Products comprise the device of claim 11. |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|--|
| <p>the collection of parity bits as if a number of inputs into nodes v_i was not constant.</p> | <p>See claim [11]-[11a].</p> <p>In the Accused Products, the encoder is configured to generate the collection of parity bits as if a number of inputs into nodes v_i was not constant.</p> <p>For example, the Accused Products implement the IEEE Standards, and such implementation and/or compliance includes an encoder configured to generate the collection of parity bits as if a number of inputs into nodes v_i was not constant.</p> <p>As noted in [11a], the parity check matrices may be used to generate a Tanner Graph that reflects the code structure. The Tanner Graph representation of the 802.11 parity check matrices show that the corresponding encoder is configured to generate the collection of parity bits as if a number of inputs into nodes v_i was not constant.</p> <p>For example, in the zoomed-in portion of the exemplary Tanner Graph shown below for the (648, 486) code ($R = 3/4$), the solid, black ovals illustrate that the number of inputs into nodes v_i is not constant.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

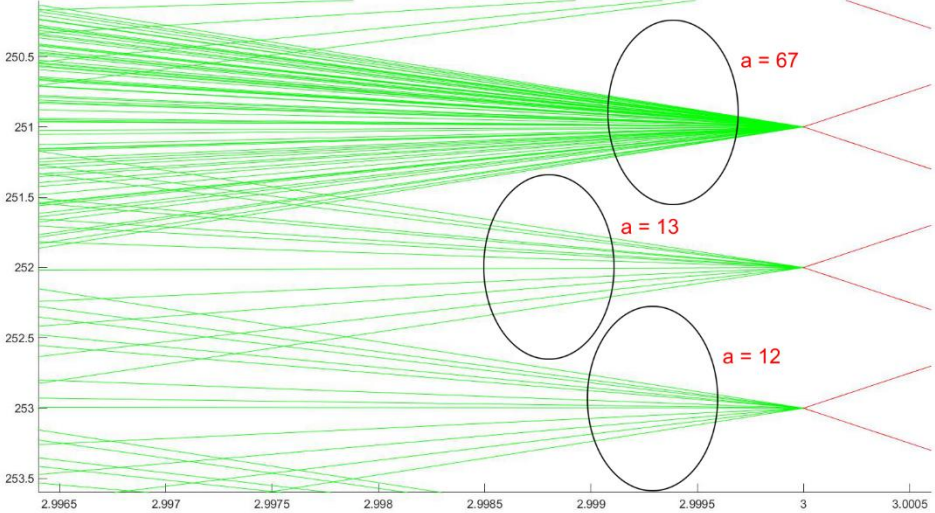
| Claim Language | Analysis |
|----------------|--|
| | <p data-bbox="1234 315 1486 331">Tanner Graph for (648,486), Rate 3/4 Code</p>  <p data-bbox="751 911 1923 1013">Every code in the 802.11 Standards may be literally implemented the number of inputs into nodes v_i not being constant. The following list shows the codes and the varying number of inputs used:</p> <ul data-bbox="871 1036 1150 1382" style="list-style-type: none">(1944, 974) – 5, 6, 57(1296, 648) – 5, 6, 57(648, 324) – 5, 6, 53(1944, 1296) – 9, 65(1296, 864) – 9, 69(648, 432) – 9, 63(1944, 1458) – 12, 72 |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|--|
| | <p>(1296, 972) - 12, 13, 73</p> <p>(648, 486) - 12, 13, 67</p> <p>(1944, 1620) – 17, 18, 70</p> <p>(1296, 1080) – 19, 74</p> <p>(648, 540) – 20, 75</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 13 | |
| [13] The device of claim 11, wherein the encoder comprises: | <p>The Accused Products comprise the device of claim 11.</p> <p>See claim [11]-[11a].</p> |
| [13a] a low-density generator matrix (LDGM) coder configured to perform an irregular repeat on message bits having a first sequence in a source data stream to output a random sequence of repeats of the message bits; and | <p>The encoders of the Accused Products include a low-density generator matrix (LDGM) coder configured to perform an irregular repeat on message bits having a first sequence in a source data stream to output a random sequence of repeats of the message bits.</p> <p>For example, the Accused Products implement the IEEE Standards, and such implementation and/or compliance includes an encoder with a low-density generator matrix (LDGM) coder configured to perform an irregular repeat on message bits having a first sequence in a source data stream to output a random sequence of repeats of the message bits.</p> <p>As shown in [11a], an irregular repeat on message bits can be represented using a Tanner Graph. This operation can also be described using a low density generator matrix (LDGM). An exemplary LDGM for the (648, 486) LDPC code (R=3/4) is shown below. The matrix is 486 by 162. Dots are used to show the locations in the LGDM that are equal to one. All</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

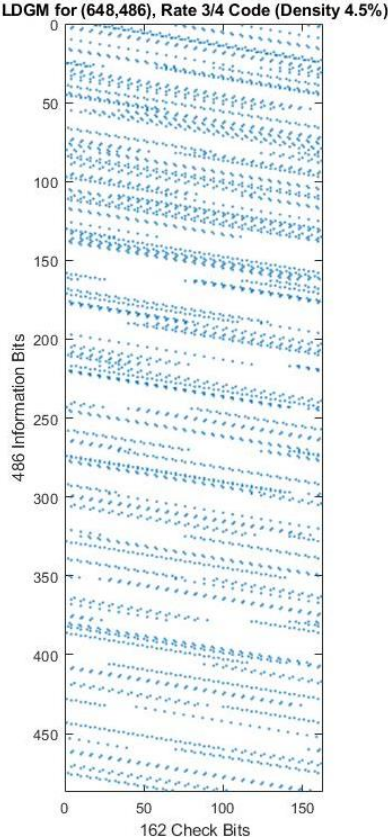
| Claim Language | Analysis |
|----------------|--|
| | <p>other locations are equal to zero. The number of times a message bit is repeated corresponds to the number of “ones” in the corresponding row of the LDGM. As shown in the LDGM below, message bits are irregularly repeated. Other LDGMs with the message bits (inputs) and check nodes (outputs) in a different order may also be used, and would satisfy the claim limitation.</p>  <p>The figure is a scatter plot titled "LDGM for (648,486), Rate 3/4 Code (Density 4.5%)". The vertical axis is labeled "486 Information Bits" and ranges from 0 to 450 with major ticks every 50 units. The horizontal axis is labeled "162 Check Bits" and ranges from 0 to 150 with major ticks every 50 units. The plot area is filled with a dense, irregular pattern of small 'x' marks, representing the positions of ones in the LDGM matrix. The distribution of these marks is non-uniform across the matrix, illustrating the irregular repetition of message bits.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|--|-------------|---------|--------|---------|--------|---------|--------|--------|--------|---------|--------|----------|--------|---------|--------|--------|--------|---------|--------|----------|---------|---------|---------|----------|---------|---------|
| | <p>See also Appendix B (showing corresponding exemplary LDGMs for all 12 parity check matrices defined in the 802.11 standard).</p> <p>Analysis of the 12 parity check matrices defined in the 802.11 standard shows the generator matrix is low density. The following are the densities of exemplary LDGMs generated from the 12 parity check matrices defined in the 802.11 standard:</p> <table><tr><th>LDGM Number</th><th>Density</th></tr><tr><td>LDGM 1</td><td>1.6204%</td></tr><tr><td>LDGM 2</td><td>2.0544%</td></tr><tr><td>LDGM 3</td><td>2.572%</td></tr><tr><td>LDGM 4</td><td>3.6574%</td></tr><tr><td>LDGM 5</td><td>0.78447%</td></tr><tr><td>LDGM 6</td><td>1.0272%</td></tr><tr><td>LDGM 7</td><td>1.286%</td></tr><tr><td>LDGM 8</td><td>1.7593%</td></tr><tr><td>LDGM 9</td><td>0.52298%</td></tr><tr><td>LDGM 10</td><td>0.6848%</td></tr><tr><td>LDGM 11</td><td>0.82305%</td></tr><tr><td>LDGM 12</td><td>1.0802%</td></tr></table> <p>An LDGM may sum “a” bits of the random sequence of repeats of the message bits, as in the above example, before XOR summation with a predecessor parity bit. Or, alternatively, an LDGM may only irregularly repeat message bits and distribute in random sequence to generate the random sequence of repeats of the message bits, without summation of “a” bits.</p> <p>Further, the message bits have a first sequence. More specifically, the sequence of the message bits is defined as having a first sequence as part of the specification of the systematic code. Each LDPC codeword is of the form $\mathbf{c}=(i_0,i_1,\dots,i_{(k-1)},p_0,p_1,\dots,p_{(n-k-1)})$, where the message bits have a first sequence a listed from i_0 to $i_{(k-1)}$. The subscripts 0 to k-1 specify the individual message bits and index their order.</p> | LDGM Number | Density | LDGM 1 | 1.6204% | LDGM 2 | 2.0544% | LDGM 3 | 2.572% | LDGM 4 | 3.6574% | LDGM 5 | 0.78447% | LDGM 6 | 1.0272% | LDGM 7 | 1.286% | LDGM 8 | 1.7593% | LDGM 9 | 0.52298% | LDGM 10 | 0.6848% | LDGM 11 | 0.82305% | LDGM 12 | 1.0802% |
| LDGM Number | Density | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 1 | 1.6204% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 2 | 2.0544% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 3 | 2.572% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 4 | 3.6574% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 5 | 0.78447% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 6 | 1.0272% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 7 | 1.286% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 8 | 1.7593% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 9 | 0.52298% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 10 | 0.6848% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 11 | 0.82305% | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LDGM 12 | 1.0802% | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | <p>The message bits are taken from (and thus “in”) a “source data stream.” The data stream is passed through to the MAC layer and then subsequently to the PHY layer. The PHY layer contains an LDPC encoder that receives the source data stream and then parses the data stream into blocks of message bits that are processed to generate a sequence of LDPC codewords. The 802.11 specification specifically describes how to take the message bits from the source data stream and parse the message bits into groups of message bits with specific first sequences. “A succession of LDPC codewords” results from LDPC encoding of the “message bits” in the “source data stream.” <i>See, e.g.</i>, 802.11n-2009 at § 20.3.11.6.2-6; 802.11-2012 at § 20.3.11.7.2-6; 802.11-2020 at § 19.3.11.7.2-6.</p> <p>20.3.11.6.6 LDPC parser</p> <p>The succession of LDPC codewords that result from the encoding process of 20.3.11.6.5 shall be converted into a bitstream in sequential fashion. Within each codeword, bit i_0 is ordered first. The parsing of this encoded data stream into spatial streams shall follow exactly the parsing rules defined for the BCC encoder, as defined in 20.3.11.7.1. However, the frequency interleaver of 20.3.11.7.3 is bypassed.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.6 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.6; IEEE 802.11-2020 at § 19.3.11.7.6.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> <p>For example, to the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe because they do not generate the random sequence of bits via a low density generator matrix (“LDGM”) coder, the accused products would infringe under the doctrine of equivalents because they generate parity bits by combining a plurality of message bits with previous parity bits.</p> <p>An LDGM is a way to represent a linear transform. It is known in the art that such transforms may be implemented in various ways. As an example, these implementations</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p>may represent the information in the LDGM via equations, physical connections to adders, arrays in memory and/or by other means. The LDGM may be expressed in alternative formats such as equations. Each column of the LDGM matrix specifies the message bits that must be repeated irregularly to form the random sequence of bits. Thus, an equation can be written for each column of the LDGM. That set of equations is an alternative and equivalent representation of the LDGM. The equations may be used directly or manipulated form to implement the transform specified by the LDGM. Information from the LDGM may be stored in forms such as equations, arrays or similar storage structures. Implementations may use repetition and/or summation of message bits in groups as specified by the LDGM (or alternative representations thereof). Looping may be used. For example, loops may operate over columns and/or rows of the LDGM. Implementations may use hard-coded operations thereby requiring little or no storage of the LDGM or information in memories. Alternatively, implementations may store the LDGM or information derived from the LDGM in memory thereby making the information available in whole or in part as the random sequence of bits is generated. Hybrid structure are also possible. Further, the generation of the random sequence of bits may be carried out using or assisted by custom or general purpose processors.</p> <p>Thus, whether or not the generation of the random sequence of bits is performed using an LDGM directly or using an equivalent hardware and/or software implementation is an insubstantial difference: each scenario accomplishes substantially the same function of generating a sum of message bits, in substantially the same way, <i>i.e.</i>, summing subsets of a plurality of message bits, and accomplishes substantially the same result of providing an efficient and effective linear error correction code.</p> <p>Further, to the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe because they do not “repeat” information bits, <i>e.g.</i>, by construing “repeat” to require copies of information bits to be stored in a separate memory, the Accused Products would infringe under the doctrine of equivalents because each information bit is used multiple times in the first encoding, and appears in a plurality of sums of information bits at the check nodes. Whether or not the information bits are duplicated and separately stored in memory before they are summed or combined is an</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|---|
| | insubstantial difference: both scenarios accomplish substantially the same function of summing pluralities of information bits at the check nodes, in substantially the same way, <i>i.e.</i> , by directing each information bit to a plurality of check nodes to be summed or combined, and accomplish substantially the same result of forming the first encoded data block. |
| [13b] an accumulator configured to XOR sum in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of repeats of the message bits. | <p>The encoders of the Accused Products include an accumulator configured to XOR sum in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of repeats of the message bits.</p> <p>For example, the Accused Products implement the IEEE Standards, and such implementation and/or compliance includes an encoder with an accumulator configured to XOR sum in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of repeats of the message bits.</p> <p>As noted in [11a], the parity check matrices of the LDPC codes in the 802.11 standard use a dual diagonal pattern. This dual diagonal pattern permits efficient encoding, including an accumulation operation.</p> <p>Also, as noted in [11a], the parity check matrices of the 802.11 standards may be used to generate Tanner Graphs that reflect the codes’ structure. For example, in the exemplary Tanner Graph shown below for the (648, 486) code ($R = 3/4$) illustrates an accumulator configured to XOR sum in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of repeats of the message bits.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

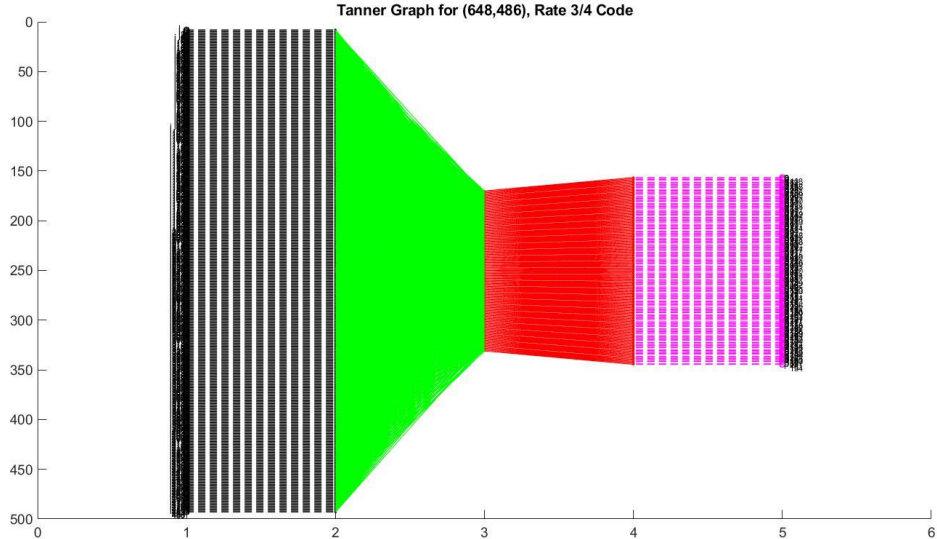
| Claim Language | Analysis |
|----------------|---|
| | <div><p>Tanner Graph for (648,486), Rate 3/4 Code</p><p>The figure is a Tanner Graph representation for a (648,486) code with a rate of 3/4. The vertical axis represents the number of nodes, ranging from 0 to 500. The horizontal axis represents the stages of the graph, ranging from 0 to 6. The graph is divided into several colored regions: a green region on the left, a red region in the middle, and a pink region on the right. The green region is bounded by a vertical line at stage 1 and a diagonal line that slopes downwards from left to right. The red region is bounded by a vertical line at stage 3 and a diagonal line that slopes upwards from left to right. The pink region is bounded by a vertical line at stage 5 and a diagonal line that slopes downwards from left to right. The graph shows a series of vertical lines representing nodes at each stage, with connections between them forming a sparse, bipartite-like structure.</p></div> <p><i>See also</i> Appendix B (showing exemplary Tanner Graph representation for all 12 parity check matrices defined in the 802.11 standards).</p> <p>More specifically, the Tanner Graph representations of the 802.11 parity check matrices reflect that “a” bits of the random sequence of repeats of the message bits and a predecessor parity bit are XOR summed. For example, in the zoomed in portion of the exemplary Tanner Graph shown below for the (648, 486) code, “a” bits of the random sequence of bits are XOR summed at each check node, as illustrated by the green lines in the figure below. These “a” bits of the random sequence of bits are XOR summed in a linear fashion with a predecessor parity bit, as illustrated by the red line on the right of the figure below.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

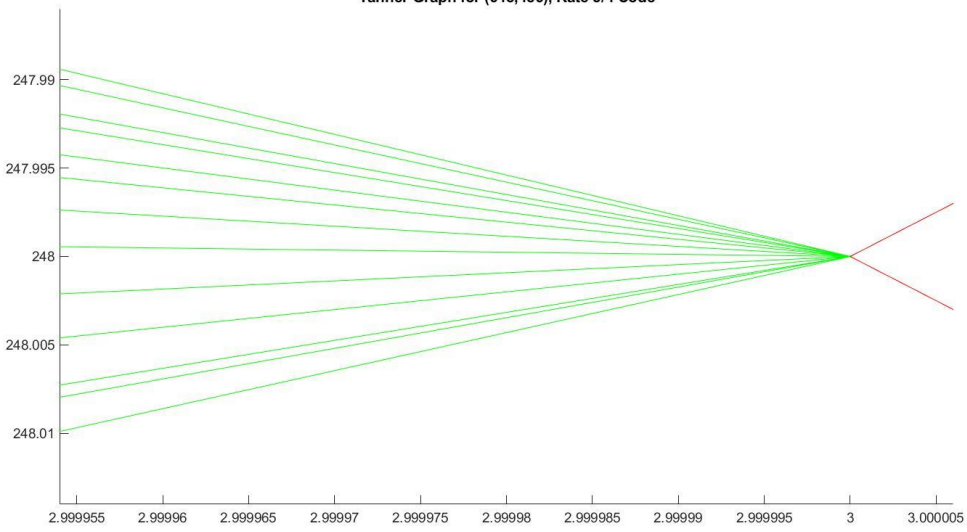
| Claim Language | Analysis |
|----------------|--|
| | <p style="text-align: center;">Tanner Graph for (648,486), Rate 3/4 Code</p>  <p><i>See also</i> Appendix B.</p> <p>Further, arithmetic in the 802.11 LDPC codes is performed using modulo two arithmetic including modulo two addition or its mathematical equivalent, XOR summing.</p> <p>For example, Annex T of the 802.11n-2009 amendment discloses a “waveform generator tool [that was] written to model the PHY transmission process described in Clause 17, Clause 19, and Clause 20.” <i>See</i> 802.11n-2009 at Annex T, § T.1. The waveform generator tool is written in Matlab format and is available in IEEE 802.11 document 11-06/1715. <i>See id.</i>; <i>see also</i> IEEE 802.11 11-06/1715. The waveform generator tool includes a function named <code>ldpc_encode(inp,ldpc_params)</code> that performs LDPC encoding. <i>See</i> IEEE doc. 11-06/1715 pp. 14-16. This function contains the lines:</p> <pre>% generate parity parity = bitand((generator_matrix* ldpc_bits) , 1);</pre> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|---|
| | <p>This line shows that the parity bits of the 802.11 LDPC codes are generated with XOR summing. Specifically, the <code>bitand()</code> function in Matlab performs a bitwise AND operation. In the code above, a generator matrix is multiplied by the <code>ldpc_bits</code>, which are the information bits as indicated in the preceeding lines of the Matlab code. The result of this multiplication is then bitwise AND'ed with 1, which means the least significant bit is maintained. Thus, the parity bit generation is performed using XOR summing, or, equivalently, mod-2 summing.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> <p>For example, to the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe because they do not “repeat” information bits, <i>e.g.</i>, by construing “repeat” to require copies of information bits to be stored in a separate memory, the Accused Products would infringe under the doctrine of equivalents because each information bit is used multiple times in the first encoding, and appears in a plurality of sums of information bits at the check nodes. Whether or not the information bits are duplicated and separately stored in memory before they are summed or combined is an insubstantial difference: both scenarios accomplish substantially the same function of summing pluralities of information bits at the check nodes, in substantially the same way, <i>i.e.</i>, by directing each information bit to a plurality of check nodes to be summed or combined, and accomplish substantially the same result of forming the first encoded data block.</p> |
| Claim 14 | |
| [14] The device of claim 12, wherein the accumulator comprises a recursive convolutional coder. | The Accused Products comprise the device of claim 12. See claim [12]. |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|---|
| | <p>In the Accused Products the accumulator comprises a recursive convolutional coder. An accumulator is a recursive convolutional encoder with a transfer function of $1/(1+D)$. Specifically, the transfer function $1/(1+D)$ reflects a system where $f(n) = f(n-1) + x$, where $f(n)$ describes the output of the accumulator, and x describes the input to the accumulator. Performing a convolution with the transfer function $1/(1+D)$ on an input stream generates an output that is the accumulation of the input stream.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 15 | |
| <p>[15] The device of claim 14, wherein the recursive convolutional coder comprises a truncated rate-1 recursive convolutional coder.</p> | <p>The Accused Products comprise the device of claim 14.</p> <p>See claim [14].</p> <p>In the Accused Products the recursive convolutional coder comprises a truncated rate-1 recursive convolutional coder.</p> <p>As shown in [14], the accumulator in the Accused Products is a recursive convolutional encoder with transfer function $1/(1+D)$. The transfer function $1 / (1 + D)$ has a rate of one. In addition, the accumulator in the Accused Products has a rate of one, because it outputs exactly one bit for each input bit. In other words, for each sum of information bits, the accumulator accumulates and outputs one parity bit.</p> <p>Further, the accumulator in the Accused Products is a truncated rate-1 recursive convolutional coder because it only operates on the number of bits corresponding to the block length, and, specifically, the number of parity bits corresponding to that block length.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|--|
| | contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents. |
| Claim 16 | |
| [16] The device of claim 14, wherein the recursive convolutional coder has a transfer function of $1/(1 + D)$. | <p>The Accused Products comprise the device of claim 14.</p> <p>See claim [14].</p> <p>In the Accused Products the recursive convolutional coder has a transfer function of $1/(1+D)$. <i>See</i> [14].</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 17 | |
| [17] The device of claim 12, further comprising a second accumulator configured to determine a second sequence of parity bits that defines a second condition that constrains the random sequence of repeats of the message bits. | <p>The Accused Products comprise the device of claim 12.</p> <p>See claim [12].</p> <p>The Accused Products include a second accumulator configured to determine a second sequence of parity bits that defines a second condition that constrains the random sequence of repeats of the message bits.</p> <p>For example, the Accused Products implement the IEEE Standards, and such implementation and/or compliance includes a second accumulator configured to determine a second sequence of parity bits that defines a second condition that constrains the random sequence of repeats of the message bits.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

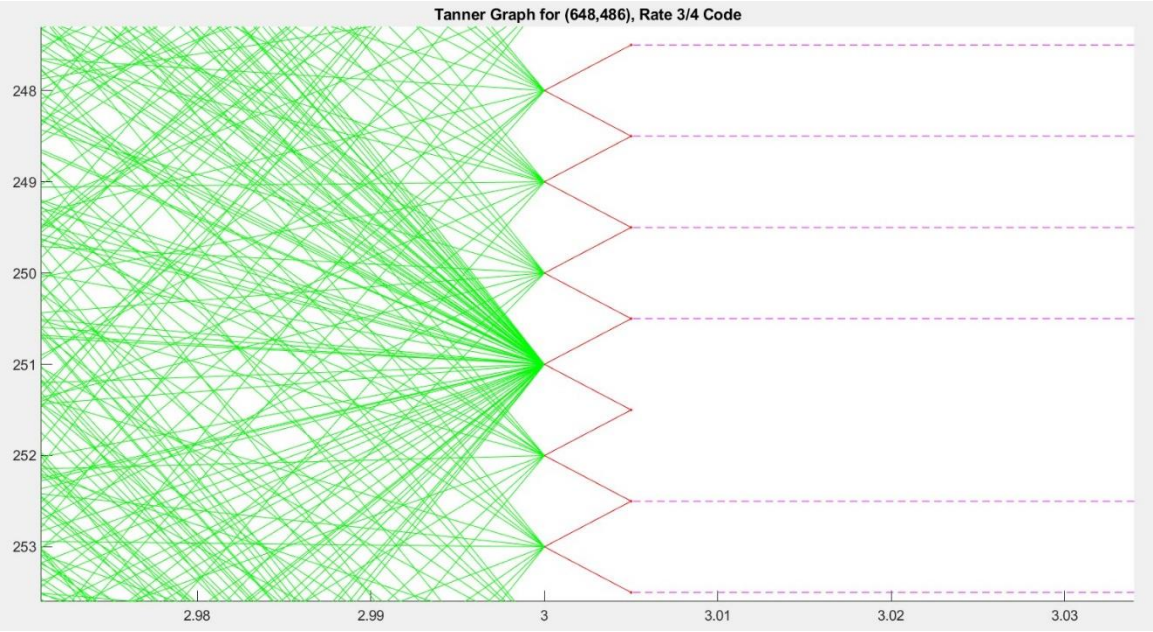
| Claim Language | Analysis |
|----------------|---|
| | <p>As noted in [11a], in the exemplary Tanner Graph corresponding to the (648, 486) code, the red edges between the check nodes and the parity bit nodes illustrate the accumulation functionality performed by an accumulator. A zoomed-in example of the same exemplary Tanner Graph is shown below illustrating an accumulator configured to determine a sequence of parity bits that defines a condition that constrains the random sequence of repeats of the message bits:</p>  <p>Further, the sub-matrix structure of the LDPC parity check matrices of the 802.11 standards is such that an efficient LDPC encoder based thereon can calculate Z parity bits in parallel.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|----|----|----|---|---|----|---|----|---|----|----|---|----|---|---|---|---|----|---|---|---|---|---|---|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|----|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|----|---|----|---|---|----|---|---|---|---|---|---|---|----|---|----|---|---|---|---|----|----|----|---|---|---|----|---|----|---|----|---|---|---|---|---|---|---|---|----|----|----|---|----|----|---|----|---|---|---|----|---|---|---|---|---|---|---|---|---|---|
| | <p>Table R.1 defines the matrix prototypes of the parity-check matrices for a codeword block length $n=648$ bits, with a subblock size $Z=27$ bits.</p> <p style="text-align: center;">Table R.1—Matrix prototypes for codeword block length $n=648$ bits, subblock size is $Z = 27$ bits</p> <p style="text-align: center;">* * *</p> <div style="border: 1px solid black; padding: 10px;"><p>(c) Coding rate $R = 3/4$.</p><table><tr><td>16</td><td>17</td><td>22</td><td>24</td><td>9</td><td>3</td><td>14</td><td>-</td><td>4</td><td>2</td><td>7</td><td>-</td><td>26</td><td>-</td><td>2</td><td>-</td><td>21</td><td>-</td><td>1</td><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>12</td><td>12</td><td>3</td><td>3</td><td>26</td><td>6</td><td>21</td><td>-</td><td>15</td><td>22</td><td>-</td><td>15</td><td>-</td><td>4</td><td>-</td><td>-</td><td>16</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td><td>-</td></tr><tr><td>25</td><td>18</td><td>26</td><td>16</td><td>22</td><td>23</td><td>9</td><td>-</td><td>0</td><td>-</td><td>4</td><td>-</td><td>4</td><td>-</td><td>8</td><td>23</td><td>11</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td></tr><tr><td>9</td><td>7</td><td>0</td><td>1</td><td>17</td><td>-</td><td>-</td><td>7</td><td>3</td><td>-</td><td>3</td><td>23</td><td>-</td><td>16</td><td>-</td><td>-</td><td>21</td><td>-</td><td>0</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td></tr><tr><td>24</td><td>5</td><td>26</td><td>7</td><td>1</td><td>-</td><td>-</td><td>15</td><td>24</td><td>15</td><td>-</td><td>8</td><td>-</td><td>13</td><td>-</td><td>13</td><td>-</td><td>11</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td></tr><tr><td>2</td><td>2</td><td>19</td><td>14</td><td>24</td><td>1</td><td>15</td><td>19</td><td>-</td><td>21</td><td>-</td><td>2</td><td>-</td><td>24</td><td>-</td><td>3</td><td>-</td><td>2</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td></tr></table></div> <p>IEEE 802.11n-2009 at Annex R, Table R.1; <i>see also</i> IEEE 802.11-2012 at Annex F, Table F-1; IEEE 802.11-2020 at Annex F, Table F-1.</p> <p>Each of the parity-check matrices can be partitioned into square subblocks (submatrices) of size $Z \times Z$. These submatrices are either cyclic-permutations of the identity matrix or null submatrices.</p> <p>The cyclic-permutation matrix P_i is obtained from the $Z \times Z$ identity matrix by cyclically shifting the columns to the right by i elements. The matrix P_0 is the $Z \times Z$ identity matrix. Figure 20-12 illustrates examples (for a subblock size of 8×8) of cyclic-permutation matrices P_i.</p> | 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | - | 0 | 0 | - | 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | - | 0 | 0 | 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 |
| 16 | 17 | 22 | 24 | 9 | 3 | 14 | - | 4 | 2 | 7 | - | 26 | - | 2 | - | 21 | - | 1 | 0 | - | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 12 | 12 | 3 | 3 | 26 | 6 | 21 | - | 15 | 22 | - | 15 | - | 4 | - | - | 16 | - | 0 | 0 | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | 18 | 26 | 16 | 22 | 23 | 9 | - | 0 | - | 4 | - | 4 | - | 8 | 23 | 11 | - | - | - | 0 | 0 | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | 7 | 0 | 1 | 17 | - | - | 7 | 3 | - | 3 | 23 | - | 16 | - | - | 21 | - | 0 | - | - | 0 | 0 | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | 5 | 26 | 7 | 1 | - | - | 15 | 24 | 15 | - | 8 | - | 13 | - | 13 | - | 11 | - | - | - | - | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 2 | 19 | 14 | 24 | 1 | 15 | 19 | - | 21 | - | 2 | - | 24 | - | 3 | - | 2 | 1 | - | - | - | - | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | $P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, P_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, P_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$ <p align="center">Figure 20-12—Examples of cyclic-permutation matrices with Z=8</p> <p>Table R.1 displays the “matrix prototypes” of parity-check matrices for all four coding rates at block length $n=648$ bits. The integer i denotes the cyclic-permutation matrix P_i, as illustrated in Figure 20-12. Vacant entries of the table denote null (zero) submatrices.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.4; <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.4; IEEE 802.11-2020 at § 19.3.11.7.4.</p> <p>For example, Z accumulators can generate Z parity bits in parallel and output those Z parity bits simultaneously. Groups of Z parity bits can be output sequentially such that individual parity bits from each of the Z accumulators its output immediately following its respective preceding parity bit. Mathematically, this can be represented as vector-based operations.</p> <p>The parity check matrices may be used to generate Tanner Graphs that reflect this parallel relationship. For example, the exemplary Tanner Graph for the (648, 486) code includes the following Z=27 sequences of parity bits:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

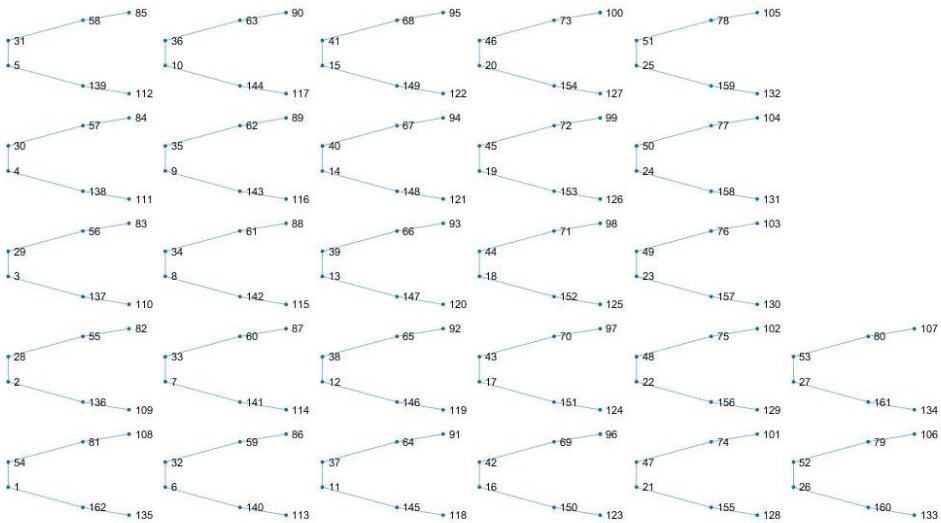
| Claim Language | Analysis |
|----------------|---|
| |  <p>Some of the same sequences of parity bits are shown in the zoomed-in version of the exemplary Tanner Graph for the (648, 486) code below:</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---------------------------|--|
| | <div data-bbox="877 310 1806 816" data-label="Figure"> </div> <p data-bbox="751 878 1925 1019">Thus, the Accused Products compies a second accumulator similar to and operating in parallel to the first accumulator in [11a] that is configured to determine a second sequence of parity bits that defines a second condition that contrains the random sequence of repeats of the message bits.</p> <p data-bbox="751 1040 1925 1219">On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 18 | |
| [18] A device comprising: | To the extent this preamble is construed to be limiting, the Accused Products are devices. See [11]. |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|--|
| <p>[18a] a message passing decoder configured to decode a received data stream that includes a collection of parity bits, the message passing decoder comprising two or more check/variable nodes operating in parallel to receive messages from neighboring check/variable nodes and send updated messages to the neighboring variable/check nodes, wherein the message passing decoder is configured to decode the received data stream that has been encoded in accordance with the following Tanner graph:</p> | <p>The Accused Products include a message passing decoder configured to decode a received data stream that includes a collection of parity bits, the message passing decoder comprising two or more check/variable nodes operating in parallel to receive messages from neighboring check/variable nodes and send updated messages to the neighboring variable/check nodes, wherein the message passing decoder is configured to decode the received data stream that has been encoded in accordance with the claimed Tanner graph.</p> <p>For example, the Accused Products implement the IEEE Standards, and such implementation and/or compliance includes a message passing decoder configured to decode a received data stream that includes a collection of parity bits, the message passing decoder comprising two or more check/variable nodes operating in parallel to receive messages from neighboring check/variable nodes and send updated messages to the neighboring variable/check nodes, wherein the message passing decoder is configured to decode the received data stream that has been encoded in accordance with the claimed Tanner graph.</p> <p>The Accused Products generate a data stream encoded in accordance with the claimed Tanner graph. <i>See, e.g.</i> claim [11].</p> <p>The Accused Products further comprise a message passing decoder configured to decode a received data stream that includes a collection of parity bits</p> <p>For example, documents from the 802.11 Working Group confirm that decoding of LDPC codes is performed using iterative decoding techniques, such as belief propagation, which is a type of message passing algorithm.</p> <p><i>See, e.g.:</i></p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

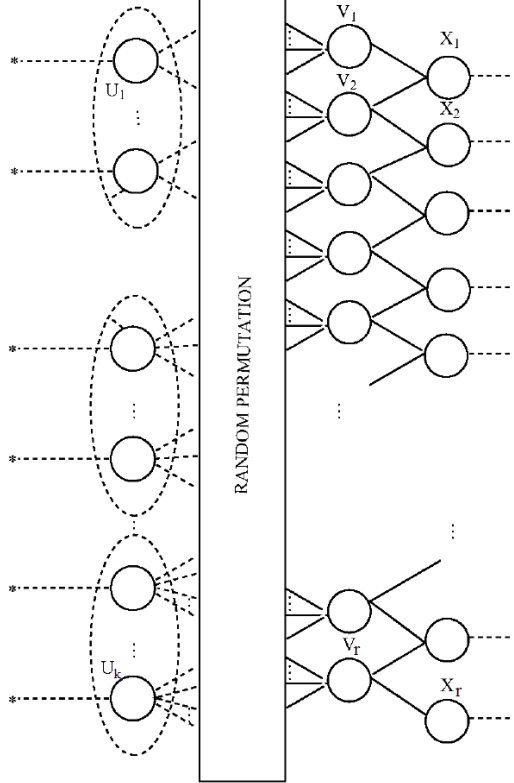
| Claim Language | Analysis |
|--|---|
|  | <div style="display: flex; justify-content: space-between; font-size: small;"> September, 2004 doc.: IEEE 802.11-04/992 </div> <h3 style="text-align: center;">Low-Density Parity-Check (LDPC) Codes</h3> <ul style="list-style-type: none"> • What is a <u>LDPC</u> code? <ul style="list-style-type: none"> – A <u>LDPC</u> code is simply a block code defined by a parity-check matrix that has a low density of ones (i.e. mostly zeros) – Decoding is done iteratively using Belief Propagation (BP) – passing of extrinsic information between <u>codeword</u> elements and parity check equations • Why do you want to use <u>LDPC</u> codes? <ul style="list-style-type: none"> – Best performing forward error correction code available – Designs have approached capacity within <u>0.0045dB</u> – Structured designs offer the great performance with faster convergence and attractive silicon solutions – For <u>802.11n</u>, structured <u>LDPC</u> is a viable and attractive solution with significant gains over the legacy <u>FEC</u> system <div style="display: flex; justify-content: space-between; font-size: x-small; margin-top: 10px;"> Submission Victor Stolpman et. al </div> <p style="margin-top: 10px;">Stolpman <i>et al.</i>, “Irregular Structured LDPC Codes and Structured Puncturing,” IEEE 802.11 doc. 802.11-04/992 (2004), at slide 3.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <div>September, 2004<div>doc.: IEEE 802.11-04/992</div></div> <div>Layered Belief Propagation</div> <div><ul style="list-style-type: none">Parity-check matrix is partitioned into layers and messages are passed betweenSpeeds convergence time significantly => High performance with low latencySignificant reduction in memory requirements (75% reduction)Most structured LDPC codes can implement layered-BP in cost effective solutions</div> <div><div>H =</div><div><div><div>0010000000001000000</div><div>1000000000100000000</div><div>0100000000001000000</div><div>1000100000100000000</div><div>0100010000100010000</div><div>0011000000001100000</div><div>0001001000000000001</div><div>0000100100000000010</div><div>0000010010000000010</div><div>0000001000000010001</div><div>0000000100000010000</div><div>0000000010000001100</div><div>0000000001000010000</div><div>00000000001000100</div></div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div></div><div>Submission<div>Victor Stolpman et. al</div></div><div><div>Id. at slide 19.</div></div></div> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <p><u>May 2005</u> <u>doc.: IEEE 802.11-05/0431r0</u></p> <p>Responses to Complexity Comments (2)</p> <p>– In relative terms, the decoding complexity of the <u>TGn Sync LDPC</u> proposal can be readily compared to that of the <u>WWiSE</u> proposal. Both codes accommodate decoding by Layered Belief Propagation. The decoding complexity has proportionalities with the block sizes and the ones density (or maximum degree) of the matrices. The <u>TGn Sync LDPC</u> proposal uses a smaller maximum block size as well as lower matrix densities (i.e., lower maximum degrees) than the <u>WWiSE</u> proposal. The base matrix column dimensions are also smaller. These factors can be expected to result in less complexity for solutions based on the <u>TGn Sync</u> proposal than for those using the <u>WWiSE</u> proposal.</p> <hr/> <p>Submission Slide 11 Bjorn A. Bjerke, Qualcomm</p> <p>Bjerke <i>et al.</i>, “Responses to reasons and cures comments on TGn Sync LDPC codes,” IEEE 802.11 doc. 802.11-05/0431r0 (2005) at slide 11.</p> <p>Message passing decoders naturally lend themselves to parallel processing. The 802.11 LDPC codes are intended for very fast decoding (<i>e.g.</i>, high throughput and very high throughput). On information and belief, the message passing decoders in the Accused Products comprise check/variable nodes operating in parallel in order to meet the timing requirements of the 802.11 standard.</p> <p>In addition, documents from the 802.11 Working Group propose decoding implementations using parallel decoding structures.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | <p data-bbox="751 293 871 326"><i>See, e.g.,</i></p> <div data-bbox="884 358 978 383"><u>July 2004</u></div> <div data-bbox="1541 358 1803 383"><u>doc.: IEEE 802.11-04/0714r0</u></div> <h2 data-bbox="1178 427 1503 483">Introduction</h2> <h3 data-bbox="890 560 1304 609">Advantages of LDPC</h3> <ul data-bbox="945 630 1755 943" style="list-style-type: none"><li data-bbox="945 630 1570 670">– Capacity approaching performance<li data-bbox="945 686 1755 781">– Parallelizability of decoding, suitable for high speed implementation<li data-bbox="945 800 1713 943">– Flexibility: LDPC is simply a kind of linear block code and its rate can be adjusted by puncturing, shortening, etc. <p data-bbox="1776 1062 1793 1086">3</p> <p data-bbox="751 1117 1919 1182">Du <i>et al.</i>, “LDPC for MIMO Systems,” IEEE 802.11 doc. 802.11-04/0714r0 (2004) at slide 3.</p> <p data-bbox="751 1206 1892 1349"><i>See also</i> Sun <i>et al.</i>, “High Throughput, Parallel, Scalable LDPC Encoder/Decoder Architecture for OFDM Systems,” 2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software, pp. 39-42 (describing belief propagation decoder using parallel decoding operations)..</p> <p data-bbox="751 1369 1871 1433">On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|--|
| | <p>contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 19 | |
| <p>[19] The device of claim 18, wherein the message passing decoder is configured to decode the received data stream that includes the message bits.</p> | <p>The Accused Products comprise the device of claim 18.</p> <p>See claim [18].</p> <p>In the Accused Products, the message passing decoder is configured to decode the received data stream that includes the message bits.</p> <p>For example, the Accused Products implement the IEEE Standards, and such implementation and/or compliance includes a message passing decoder configured to decode the received data stream that includes the message bits.</p> <p>The LDPC encoder for 802.11 generates a codeword that includes the message bits. Each LDPC codeword is of the form $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, where the message bits have a first sequence a listed from i_0 to $i_{(k-1)}$. Thus, the codeword received by the decoder includes the message bits.</p> <p>20.3.11.6.3 LDPC encoder</p> <p>For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)})$, of size k, into a codeword, \mathbf{c}, of size n, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = \mathbf{0}$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.</p> <p>IEEE 802.11n-2009 at § 20.3.11.6.3 (emphasis added); <i>see also</i> IEEE 802.11-2012 at § 20.3.11.7.3; IEEE 802.11-2020 at § 19.3.11.7.3.</p> <p>An LDPC decoder for 802.11 is designed to decode a received data stream that includes the message bits.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

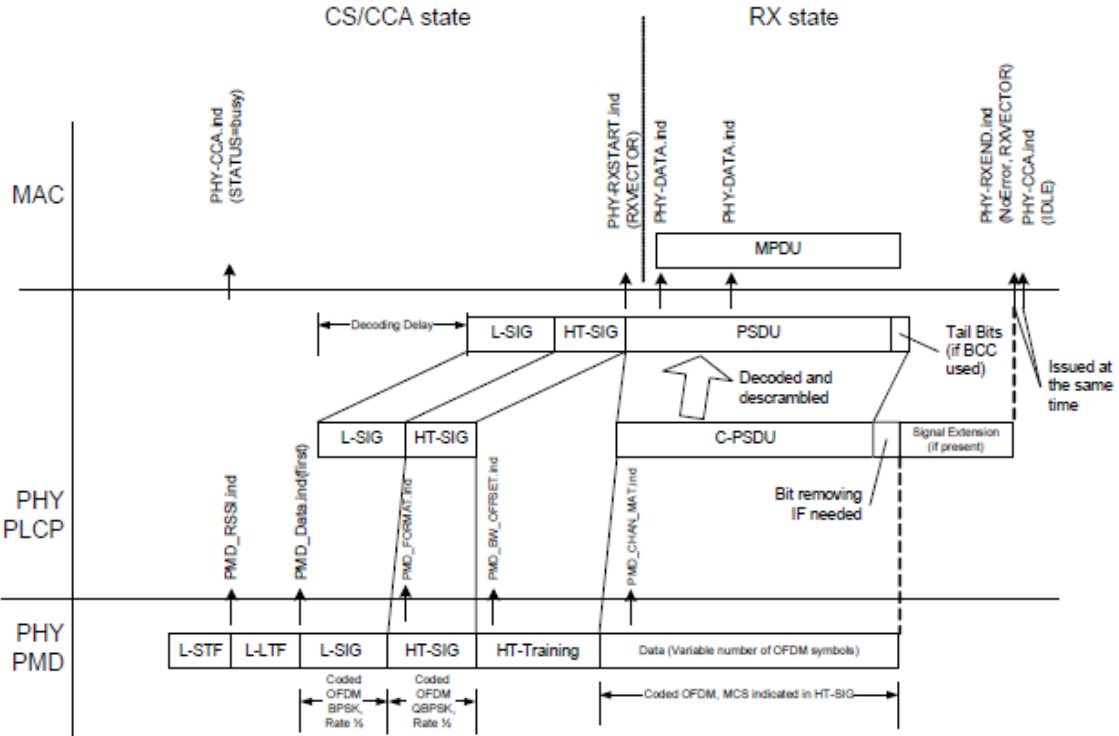
| Claim Language | Analysis |
|----------------|---|
| | <p>See, generally, IEEE 802.11n-2009 at § 20.3.24 (“PLCP receive procedure”); see also IEEE 802.11-2012 at § 20.3.23 (same); 802.11-2020 at § 19.3.21.</p>  <p>The diagram illustrates the PLCP receive procedure across three layers: MAC, PHY PLCP, and PHY PMD. At the PHY PMD layer, the sequence is L-STF, L-LTF, L-SIG, HT-SIG, HT-Training, and Data (Variable number of OFDM symbols). At the PHY PLCP layer, L-SIG and HT-SIG are received from the PMD. The C-PSDU is received after HT-Training. At the MAC layer, the PSDU is received from the PLCP. The MPDU is then processed. Key events and indicators include: - PHY-CCA.ind (STATUS=busy) at the start. - PHY-RXSTART.ind (RXVECTOR) and PHY-DATA.ind at the start of the PSDU. - PHY-RXEND.ind (NoError, RXVECTOR) and PHY-CCA.ind (IDLE) at the end. Processing steps include a decoding delay, decoding and descrambling of the C-PSDU, and bit removing if needed. Timing markers include PMD_RSSI.ind, PMD_Data.ind(first), PMD_FORMAT.ind, PMD_BW_OFFSET.ind, and PMD_CHAN_MAT.ind. Coded OFDM symbols are indicated for L-SIG, HT-SIG, and the Data portion. Tail Bits (if BCC used) and Signal Extension (if present) are also shown." data-bbox="375 240 905 690"/> <p>IEEE 802.11n-2009 at Figure 20-23; see also IEEE 802.11-2012 at Figure 20-26; IEEE 802.11-2020 at Figure 19-26.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> </p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|--|---|
| Claim 20 | |
| <p>[20] The device of Claim 18, wherein the message passing decoder is configured to decode the received data stream as if a number of inputs into nodes v_i was not constant.</p> | <p>The Accused Products comprise the device of claim 18.</p> <p>See claim [18].</p> <p>In the Accused Products, the message passing decoder is configured to decode the received data stream as if a number of inputs into nodes v_i was not constant.</p> <p>For example, the Accused Products implement the IEEE Standards, and such implementation and/or compliance includes a message passing decoder configured to decode the received data stream as if a number of inputs into nodes v_i was not constant.</p> <p>For example, the 802.11 LDPC codes may be represented as Tanner Graphs where the number of inputs into nodes v_i are not constant. <i>See</i> [3]. The decoders in the Accused Products must be configured to decode a received data stream encoded according to such constraints.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |
| Claim 21 | |
| <p>[21] The device of claim 18, wherein the message passing decoder is configured to decode in the linear time at rates that approach a capacity of a channel.</p> | <p>The Accused Products comprise the device of claim 18.</p> <p>See claim [18].</p> <p>In the Accused Products, the message passing decoder is configured to decode in the linear time at rates that approach a capacity of a channel.</p> <p>For example, the complexity of the decoder is measured relative to the block length, n. As shown in [25c], the decoder used in the Accused Products is an iterative decoder. For an</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

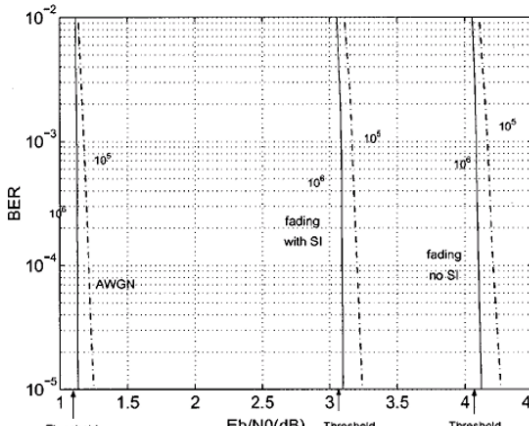
| Claim Language | Analysis |
|----------------|---|
| | <p>encoded data stream containing a codeword with k information bits and $n-k$ parity bits, the decoder operates on k information nodes, $n-k$ check nodes, and $n-k$ parity nodes, with the number of operations per iteration approximately constant and the number of iterations per node approximately constant. Thus, the complexity is proportional to $k + n-k + n-k = 2n-k$, which is linear with respect to n. Accordingly, the decoding happens in linear time.</p> <p>In addition, documents from the 802.11 working group describing the proposed decoding for LDPC codes confirm that LDPC decoders for 802.11 LDPC codes decode at rates that approach a capacity of a channel.</p> <div style="text-align: center;"> <p>November 2003 doc.: IEEE 802.11-03/0865r1</p> <h3>Large Block LDPCs in Fading</h3>  <p>For large block sizes, In this case 10^5 and 10^6, LDPCs perform extremely close to capacity.</p> <p>For a code with $R = \frac{1}{2}$ in AWGN, $C = \sim 1.2$ dB E_b/N_0 (BICO).</p> </div> <p>Fig. 3. Comparison of thresholds and simulation results for rate-1/2, (3,6) regular LDPC codes on the AWGN channel, and the uncorrelated Rayleigh fading channels with or without SI. The codes used in simulations are of block size 10^5 and 10^6.</p> <div style="display: flex; justify-content: space-between; font-size: small;"> Submission Slide 10 Eric Jacobsen, Intel </div> <p>Jacobsen, “LDPC FEC for IEEE 802.11n Applications,” IEEE 802.11 doc. 802.11-03/0865r1 (2003) at slide 10.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <div data-bbox="886 310 972 331">July 2004</div> <div data-bbox="1520 310 1766 331">doc.: IEEE 802.11-04/0714r0</div> <div data-bbox="1165 375 1482 428"><h2>Introduction</h2></div> <div data-bbox="890 501 1289 548"><h3>Advantages of LDPC</h3></div> <div data-bbox="940 566 1722 870"><ul style="list-style-type: none">– Capacity approaching performance– Parallelizability of decoding, suitable for high speed implementation– Flexibility: LDPC is simply a kind of linear block code and its rate can be adjusted by puncturing, shortening, etc.</div> <div data-bbox="1745 985 1759 1005">3</div> <div data-bbox="753 1037 1921 1102"><p>Du <i>et al.</i>, “LDPC for MIMO Systems,” IEEE 802.11 doc. 802.11-04/0714r0 (2004) at slide 3.</p></div> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <div data-bbox="877 302 1772 326"> <div>September, 2004</div> <div>doc.: IEEE 802.11-04/992</div> </div> <p align="center">Low-Density Parity-Check (LDPC) Codes</p> <ul style="list-style-type: none"> • What is a LDPC code? <ul style="list-style-type: none"> – <u>A LDPC code is simply a block code</u> defined by a parity-check matrix that has a low density of ones (i.e. mostly zeros) – Decoding is done iteratively using Belief Propagation (BP) – passing of extrinsic information between codeword elements and parity check equations • Why do you want to use LDPC codes? <ul style="list-style-type: none"> – <u>Best performing forward error correction code available</u> – Designs have approached capacity within 0.0045dB – Structured designs offer the great performance with faster convergence and attractive silicon solutions – For 802.11n, structured LDPC is a viable and attractive solution with significant gains over the legacy FEC system <div data-bbox="884 943 1791 961"> <div>Submission</div> <div>Victor Stolpman et. al</div> </div> <p>Stolpman <i>et al.</i>, “Irregular Structured LDPC Codes and Structured Puncturing,” IEEE 802.11 doc. 802.11-04/992 (2004), at slide 3.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|---|---|
| Claim 22 | |
| <p>[22] The device of claim 18, wherein the message passing decoder comprises a belief propagation decoder.</p> | <p>The Accused Products comprise the device of claim 18.</p> <p>See claim [18].</p> <p>In the Accused Products, the message passing decoder comprises a belief propagation decoder.</p> <p>For example, the Accused Products implement the IEEE Standards, and such implementation and/or compliance includes a message passing decoder comprising a belief propagation decoder.</p> <p>For example, documents from the 802.11 Working Group state that decoding of LDPC codes is performed using iterative decoding techniques, such as belief propagation.</p> <p><i>See, e.g.:</i></p> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|---|
| | <div>September, 2004doc.: IEEE 802.11-04/992</div> <div>Low-Density Parity-Check (LDPC) Codes</div> <div><ul style="list-style-type: none">What is a <u>LDPC</u> code?<ul style="list-style-type: none">A <u>LDPC</u> code is simply a block code defined by a parity-check matrix that has a low density of ones (i.e. mostly zeros)Decoding is done iteratively using Belief Propagation (BP) – passing of extrinsic information between <u>codeword</u> elements and parity check equationsWhy do you want to use <u>LDPC</u> codes?<ul style="list-style-type: none">Best performing forward error correction code availableDesigns have approached capacity within <u>0.0045dB</u>Structured designs offer the great performance with faster convergence and attractive silicon solutionsFor <u>802.11n</u>, structured <u>LDPC</u> is a viable and attractive solution with significant gains over the legacy <u>FEC</u> system</div> <div>SubmissionVictor Stolpman et. al</div> <div>Stolpman <i>et al.</i>, “Irregular Structured LDPC Codes and Structured Puncturing,” IEEE 802.11 doc. 802.11-04/992 (2004), at slide 3.</div> |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | <div><div>September, 2004<div>doc.: IEEE 802.11-04/992</div></div><div>Layered Belief Propagation</div><div><ul style="list-style-type: none">Parity-check matrix is partitioned into layers and messages are passed betweenSpeeds convergence time significantly \Rightarrow High performance with low latencySignificant reduction in memory requirements (75% reduction)Most structured LDPC codes can implement layered-BP in cost effective solutions</div><div><div><div>H =</div><div><table><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table></div><div><div>Submission</div><div>Victor Stolpman et. al</div></div></div><div><div>Id.</div><div>at slide 19.</div></div></div></div> | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exhibit 2 – Preliminary Claim Chart For U.S. Patent No. 7,421,032

| Claim Language | Analysis |
|----------------|--|
| | <p><u>May 2005</u> <u>doc.: IEEE 802.11-05/0431r0</u></p> <p>Responses to Complexity Comments (2)</p> <p>– In relative terms, the decoding complexity of the <u>TGn Sync LDPC</u> proposal can be readily compared to that of the <u>WWiSE</u> proposal. Both codes accommodate decoding by Layered Belief Propagation. The decoding complexity has proportionalities with the block sizes and the ones density (or maximum degree) of the matrices. The <u>TGn Sync LDPC</u> proposal uses a smaller maximum block size as well as lower matrix densities (i.e., lower maximum degrees) than the <u>WWiSE</u> proposal. The base matrix column dimensions are also smaller. These factors can be expected to result in less complexity for solutions based on the <u>TGn Sync</u> proposal than for those using the <u>WWiSE</u> proposal.</p> <hr/> <p>Submission Slide 11 Bjorn A. Bjerke, Qualcomm</p> <p>Bjerke <i>et al.</i>, “Responses to reasons and cures comments on TGn Sync LDPC codes,” IEEE 802.11 doc. 802.11-05/0431r0 (2005) at slide 11.</p> <p>On information and belief, products made or sold by Samsung that implement the IEEE Standards do so in such a way that infringes this limitation. To the extent that Samsung contends that their LDPC encoding and/or decoding implementations do not literally infringe the limitations of this claim, such differences are insubstantial and thus the claim limitations are satisfied under the doctrine of equivalents.</p> |