Title: METHOD AND SYSTEM FOR DYNAMIC INTERLEAVER ADAPTATION SCHEME

Abstract: The present invention provides a method and apparatus for a new interleaver adaptation scheme that achieves “error free” and zero delay (interleaving - triangular) or near zero delay variation (interleaving - GCI), and with easier implementation but no additional memory required. The dummy insertion methods and systems embodiments of the invention provide an effective dummy byte insertion scheme for applications that require seamless on-line rate changes, e.g., SRA (seamless rate adaptation), DRR (dynamic rate re-partitioning) and adaptive INP (impulse noise protection).
For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.
METHOD AND SYSTEM FOR DYNAMIC INTERLEAVER ADAPTATION SCHEME

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims priority to U.S. Provisional Application 60/634,260, filed December 9, 2004, entitled Method and System For Dynamic Interleaver Adaptation Scheme in VDSL; and to U.S. Provisional Patent Application 60/643,309, filed January 13, 2005, entitled Dynamic Interleaver Reconfiguration; and to U.S. Provisional Patent Application 60/656,019, filed February 25, 2005, entitled Dynamic Interleaver Reconfiguration, each of the above-referenced applications being incorporated herein by reference in their entirety.

BACKGROUND OF THE INVENTION

[0002] The present invention relates generally to the field of telecommunications and, more particularly, to delivering services over telephone (switched) networks (POTS, PSTN) and leased point-to-point two-wire telephone type circuits, such as via modems or other forms of transceivers operating over twisted pair. An exemplary system might have a data signaling rate of 33600 bits/second. For example, multi-channel, multi-carrier communications systems such as discrete multi-tone (DMT) systems may be used.

[0003] The explosive growth of the Internet has created a demand for high data rates for business and residential users (SOHO - small office/house office) that rely on standard analog plain old telephone systems (POTS) that use a copper wire twisted pair to carry the information. The need for high-speed access to the home is increasing due to the availability of information, data, high-bandwidth video and the
like, such as from the world wide web. Because of such demand, higher speed modems are required; modems operating at rates of for example 33,600 bits/second are now in use. A multitude of competing communication technologies provide high-speed access to the home such as cable modems and digital subscriber line (xDSL) equipment. DSL equipment may utilize the existing analog POTS that use a copper wire twisted pair to carry the information. Because of bandwidth limitation (4 KHz), and power limitation of the telephone network, line coding schemes are used to encode digital signals into analog signals that convey the analog information over the analog telephone network. Such line coding schemes should avoid the undesirable bandwidth or power increase.

[0004] Line coding schemes manipulate the analog carrier signal, which has three attributes, amplitude, phase and frequency. One or more of such attributes may be manipulated by known modulation techniques. One such technique is quadrature amplitude modulation (QAM) whereby the carrier signal’s phase and amplitude is modulated to encode more data within a frequency bandwidth. One example of a QAM modulation system sends two bits of information per QAM symbol, where the digital values can be encoded and the corresponding amplitude and phase can be represented using the constellation.

[0005] Data from a personal computer or other equipment at the customer premise (CPE) are sent to a transmitter, which arranges the data into frame packets. The packetized signal is then quadrature amplitude modulation encoded and error encoded using trellis encoding to improve the noise immunity using a convolutional coder to select a sequence of subsets in a partitioned signal constellation. A numerical
symbol vector is trellis encoded. The trellis encoding starts with the most significant symbol and ends with the least significant symbol of the vector, a process which employs convolutional encoding that converts the input symbol to another symbol and then maps the encoded symbol to its corresponding 16 QAM signal constellation point.

[0006] VDSL, which is referenced generally and shall be used herein to include where appropriate progeny such as VDSL2, provides symmetric and asymmetric bandwidth and VDSL reference standards have been developed by International Telecommunications Union (ITU-T), American National Standards Institute (ANSI) T1/E1, and European Telecommunications Standards Institute (ETSI). Standards of particular interest are T1.424; ITU G.993.1; TS101 270-1; and TS101 270-2, all of which are incorporated herein by reference. Also, the dynamic interleaver adaptation scheme is intended for use with adopted VDSL triangular interleaver as well as with a soon to be adopted convolutional triangular interleaver for VDSL2.

[0007] A conventional multi-carrier VDSL system normally consists of a data-to-symbol converter, an IFFT (Inverse Fast Fourier Transform), a cyclic extension adder, D/A (digital-to-analog) converter for its transmitter, and an A/D (analog-to-digital) converter, a FFT (Fast Fourier Transform), a frequency domain equalizer, and a symbol-to-data converter for its receiver.

SUMMARY OF INVENTION

[0008] One embodiment of the present invention is a communications system having an adaptive interleaver scheme that comprises an interleaver adapted to receive
data and to interleave the received data to thereby generate interleaved data, and a de-interleaver adapted to receive the interleaved data and to de-interleave the received interleaved data to thereby generate data essentially equivalent to the received data.

The interleaver defines a dummy insertion pattern in the received data prior to reordering of the received data to thereby form transition data to effect interleaver depth (D) and interleaver shape and to accommodate interleaver line rates to minimize delay variations and to maintain impulse noise protection (INP).

[0009] Another embodiment of the present invention is a communications system that comprises an encoder adapted to receive and encode a signal and generate an encoded signal, an interleaver adapted to receive and to interleave the encoded signal to thereby generate an interleaved encoded signal, the interleaver defining a dummy insertion pattern in the received signal prior to reordering of the received signal to thereby form a transition signal to effect interleaver depth (D) and interleaver shape and to accommodate interleaver line rates to minimize delay variations and to maintain impulse noise protection (INP), a modulator that modulates the interleaved encoded signal to generate an encoded, interleaved, modulated signal, a demodulator receiving and demodulating the modulated interleaved encoded signal and generating a demodulated interleaved encoded signal, a de-interleaver adapted to receive and deinterleave the demodulated interleaved encoded signal and generate a deinterleaved demodulated encoded signal, and a decoder adapted to decode the deinterleaved demodulated encoded signal and generate an output signal essentially equivalent to the received signal.
Another embodiment of the present invention is a communications system having an adaptive interleaver scheme comprising an interleaver means for receiving data and interleaving the received data to thereby generate interleaved data, and a de-interleaver means for receiving the interleaved data and deinterleaving the received interleaved data to thereby generate data essentially equivalent to the received data. The interleaver means defines a dummy insertion pattern in the received data prior to reordering of the received data to thereby form transition data to effect interleaver depth (D) and interleaver shape and to accommodate interleaver line rates to minimize delay variations and to maintain impulse noise protection (INP).

Another embodiment of the present invention is a communications system comprising: an encoder adapted to receive and encode a signal and generate an encoded signal; an interleaver means for receiving and interleaving the encoded signal to thereby generate an interleaved encoded signal, the interleaver means defining a dummy insertion pattern in the received signal prior to reordering of the received signal to thereby form a transition signal to effect interleaver depth (D) and interleaver shape and to accommodate interleaver line rates to minimize delay variations and to maintain impulse noise protection (INP); a modulator that modulates the interleaved encoded signal to generate an encoded, interleaved, modulated signal;

a demodulator receiving and demodulating the modulated interleaved encoded signal and generating a demodulated interleaved encoded signal; a de-interleaver means for receiving and deinterleaving the demodulated interleaved encoded signal and generate a deinterleaved demodulated encoded signal; and a
decoder adapted to decode the deinterleaved demodulated encoded signal and generate an output signal essentially equivalent to the received signal.

[0013] Another embodiment of the present invention is a method of performing an adaptive interleaver scheme comprising: receiving data at an interleaver and interleaving the received data to thereby generate interleaved data;

[0014] receiving at a deinterleaver the interleaved data and deinterleaving the received interleaved data to thereby generate data essentially equivalent to the received data; and defining at the interleaver a dummy insertion pattern in the received data prior to reordering of the received data to thereby form transition data to effect interleaver depth (D) and interleaver shape and to accommodate interleaver line rates to minimize delay variations and to maintain impulse noise protection (INP).

[0015] One inventive aspect provides a method and apparatus for a new interleaver adaptation scheme that achieves "error free" and zero delay (interleaving - triangular) or near zero delay variation (interleaving - GCI), and with easier implementation but no additional memory required.

[0016] The invention is disclosed or use in both directions, decrease in rate M and increase in rate M. Although the invention may be described in the particular embodiments of decrease in rate M4 to M3 and increase in rate M3 to M4, it should be understood that those skilled in the art would appreciate and know how to adopt the formulas and schemes herein disclosed to effect other rate increases/decreases. The invention relates to transceivers and applies in data transmitted in a bi-directional manner.
A first embodiment of the present invention provides a communications system having a dynamic interleaver adaptation scheme, the system comprising: an interleaver adapted to receive data and to interleave the received data to thereby generate interleaved data; and a de-interleaver adapted to receive the interleaved data and to de-interleave the received interleaved data to thereby generate data essentially equivalent to the received data. The interleaver inserting dummy data in the received data to transform the shape of the interleaver to effect a constant delay by adapting interleaver depth parameter (M) to be proportional with interleaver line rate. The interleaver input speed is adapted based on a change in the M parameter and interleaver output speed, line rate and deinterleaver speed remain constant until transformation is completed.

In another embodiment of the present invention, a communications system, comprises: an encoder adapted to receive and encode a signal and generate an encoded signal; an interleaver adapted to receive and to interleave the encoded signal to thereby generate an interleaved encoded signal, the interleaver defining a dummy insertion pattern in the received signal to transform the shape of the interleaver to effect a constant delay by adapting interleaver depth parameter (M) to be proportional with interleaver line rate; a modulator that modulates the interleaved encoded signal to generate an encoded, interleaved, modulated signal; a demodulator receiving and demodulating the modulated interleaved encoded signal and generating a demodulated interleaved encoded signal; a deinterleaver adapted to receive and deinterleave the demodulated interleaved encoded signal and generate a deinterleaved demodulated encoded signal; and a decoder adapted to decode the deinterleaved demodulated
encoded signal and generate an output signal essentially equivalent to the received signal. The interleaver input speed is adapted based on a change in the M parameter and interleaver output speed, line rate and deinterleaver speed remain constant until transformation is completed.

[0019] In yet another embodiment of the present invention, a communications system having an adaptive interleaver scheme comprises: an interleaver means for receiving data and interleaving the received data to thereby generate interleaved data and for inserting dummy data in the received data to transform the shape of the interleaver to effect a constant delay by adapting interleaver depth parameter (M) to be proportional with interleaver line rate; and a de-interleaver means for receiving the interleaved data and deinterleaving the received interleaved data to thereby generate data essentially equivalent to the received data. The interleaver input speed is adapted based on a change in the M parameter and interleaver output speed, line rate and deinterleaver speed remain constant until transformation is completed.

[0020] In another embodiment of the present invention, a communications system, comprises: an encoder adapted to receive and encode a signal and generate an encoded signal; an interleaver means for receiving data and interleaving the received data to thereby generate interleaved data and for inserting dummy data in the received data to transform the shape of the interleaver to effect a constant delay by adapting interleaver depth parameter (M) to be proportional with interleaver line rate; a modulator that modulates the interleaved encoded signal to generate an encoded, interleaved, modulated signal; a demodulator receiving and demodulating the modulated interleaved encoded signal and generating a demodulated interleaved
encoded signal; a de-interleaver means for receiving and deinterleaving the
demodulated interleaved encoded signal and generate a deinterleaved demodulated
encoded signal; and a decoder adapted to decode the deinterleaved demodulated
encoded signal and generate an output signal essentially equivalent to the received
signal. The interleaver input speed is adapted based on a change in the M parameter
and interleaver output speed, line rate and deinterleaver speed remain constant until
transformation is completed.

[0021] In another embodiment the present invention provides a method of
performing an adaptive interleaver scheme comprising: receiving data at an interleaver
and interleaving the received data to thereby generate interleaved data; receiving at a
deinterleaver the interleaved data and deinterleaving the received interleaved data to
thereby generate data essentially equivalent to the received data; inserting at the
interleaver a dummy insertion pattern in the received data to transform the shape of
the interleaver to effect a constant delay by adapting interleaver depth parameter (M)
to be proportional with interleaver line rate; and adapting interleaver input speed
based on a change in the M parameter whereby interleaver output speed, line rate and
deinterleaver speed remain constant until transformation is completed.

[0022] One inventive aspect provides a method and apparatus for a new
interleaver adaptation scheme that achieves "error free" and zero delay (interleaving -
triangular) or near zero delay variation (interleaving - GCI), and with easier
implementation but no additional memory required.

[0023] The invention is disclosed or use in both directions, decrease in rate M
and increase in rate M. Although the invention may be described in the particular
embodiments of decrease in rate M4 to M3 and increase in rate M3 to M4, it should be understood that those skilled in the art would appreciate and know how to adopt the formulas and schemes herein disclosed to effect other rate increases/decreases. The invention relates to transceivers and applies in data transmitted in a bi-directional manner.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0024] FIG. 1 is a data stream representation of an exemplary triangular implementation of GCI for use in the present invention.

[0025] FIG. 2 is a data stream representation of an interleaving data stream for implementation with the first and second embodiments of the present invention for output dummy insertion.

[0026] FIG. 3 is a data stream representation of an exemplary output dummy insertion pattern for “Decrease of D” for implementation with the first and second embodiments of the present invention.

[0027] FIG. 4 is a data stream representation of a dummy insertion pattern for “Increase of D” for implementation with the first and second embodiments of the present invention.

[0028] FIG. 5 is a data stream representation of an interleaving data stream for implementation with the third embodiment of the present invention for input dummy insertion.

[0029] FIG. 6 is a data stream representation of a dummy insertion pattern for “Decrease of D” for implementation with the third embodiment of the present invention.
[0030] FIG. 7 is a data stream representation of a dummy insertion pattern for “Increase of D” for implementation with the third embodiment of the present invention.

[0031] FIG. 8 is a data stream representation of an exemplary negative dummy byte insertion scenario for implementation with the first and second embodiments of the present invention.

[0032] FIG. 9 is a data stream representation of an exemplary negative dummy byte insertion scenario for implementation with the first and second embodiments of the present invention.

[0033] FIG. 10 is a data stream representation of an exemplary negative dummy byte insertion scenario for implementation with the third embodiment of the present invention.

[0034] FIG. 11 is a data stream representation of an exemplary negative dummy byte insertion scenario for implementation with the third embodiment of the present invention.

[0035] FIG. 12 is a data stream representation of an interleaver output stream: no overlapping and long gap.

[0036] FIG. 13 is a data stream representation of an interleaver output stream: overlapping and shorter gap.

[0037] FIG. 14 is a data stream representation of an interleaver output stream: complete overlapping and no gap (seamless).

[0038] FIG. 15 is a data stream representation of an insert dummy bytes for the decreasing depth transition.
[0039] FIG. 16 is a data stream representation of an insert dummy bytes for the increasing depth transition.

[0040] FIG. 17 is a data stream representation of changing the shape of the interleaver buffer.

[0041] FIG. 18 is a data stream representation of different dummy insertion patterns (Decrease of D).

[0042] FIG. 19 is a data stream representation of condensed spreading affected by dummy bytes insertion.

[0043] FIG. 20 is a data stream representation of ideal condensed spreading during transition.

[0044] FIG. 21 is a data stream representation of condensed spreading due to dummy insertion.

[0045] FIG. 22 is a data stream representation of a first embodiment of the present invention dynamic interleaver configuration scheme.

[0046] FIG. 23 is a block diagram illustrating a communications system embodiment incorporating the interleaving and de-interleaving schemes of the present invention.

[0047] FIG. 24 is a data stream representation of a second embodiment of the present invention dynamic interleaver configuration scheme (triangular, decrease of M from 4 to 3).

[0048] FIG. 25 is a data stream representation related to the second embodiment dynamic interleaver configuration scheme (triangular, increase of M from 3 to 4).
FIG. 26 is a data stream representation related to the second embodiment dynamic interleaver configuration scheme (GCI, decrease of D from 19 to 13).

FIG. 27 is a data stream representation related to the second embodiment dynamic interleaver configuration scheme (GCI, increase of D from 13 to 19).

FIG. 28 is a data stream representation of a third embodiment of the present invention dynamic interleaver configuration scheme (triangular, decrease of M from 3 to 4).

FIG. 29 is a data stream representation related to the third embodiment dynamic interleaver configuration scheme (triangular, increase of M from 3 to 4).

FIG. 30 is a data stream representation related to the third embodiment dynamic interleaver configuration scheme (GCI, decrease of D from 19 to 13).

FIG. 31 is a data stream representation related to the third embodiment dynamic interleaver configuration scheme (GCI, increase of D from 13 to 19).

**DETAILED DESCRIPTION OF INVENTION**

The present invention provides a dynamic interleaver configuration to effect improved performance and aspects of the invention provide one or more of: INP preservation throughout transition; zero delay variation; reduced memory requirements; error-free operation; versatility between triangular and GCI interleaver; and versatility in granularity in interleaving depth.
First, we address the design of a dynamic interleaver for seamless depth transition and in particular with respect to a triangular interleaver and the triangular interleaver implementation of a general convolutional interleaver (GCI).

There are different ways to implement a GCI. In the example discussed here for illustrative purposes and not by way of limitation, the triangular implementation of GCI 100, as shown in Figure 1, defines how dummy bytes are inserted based on the dynamic interleaver schemes proposed hereinbelow. The notations used in this description include: j - delay line index; L - length of each delay line; M - interleaving depth parameter; and r - reordering index. Figure 1 provides definitions for the notations j, I_j, M_j, and r_j. Unless otherwise provided herein the notations and definitions thus described will be used throughout this specification. In accordance with this exemplary interleaver, during the transition the rate is changed from Rate_1 to Rate_2 and the interleaving depth is changed from D_1 to D_2. Based on the definition of M in Figure 1, we define M_j for the interleaver with larger D = Max(D_1,D_2), define M'_j for the interleaver with smaller D = Min(D_1,D_2), and define \Delta M_j = M_j - M'_j. The term “fast data” represents the data bytes with faster rate, whereas “slow data” represents the data bytes with slower rate.

Now with reference to Figure 3, for example in data stream (1) 302 the data bytes 304 are “fast data” and the data bytes 306 are “slow data.” While in Figure 4, the data bytes 404 are “slow data” and the data bytes 406 are “fast data” in data stream (1) 402.
[0059] The term “transition data” represents the data bytes transmitted during transition period at the interleaver output, whereas the data bytes transmitted during non-transition period are “non-transition data.” For example, in data stream (2) 308 or (3) 310 of Figure 3, the data bytes 312 and 314, respectively, enclosed within the rectangle (between “transition begins” and “transition ends”) are “transition data,” the other data bytes, i.e., those outside the rectangle, are “non-transition data.”

[0060] Figures 2 through 4 illustrate one exemplary set of dummy patterns illustrating a first embodiment of the present output dummy insertion method and system. In the exemplary embodiment of Figure 3, the following rules define the dummy insertion pattern for “Decrease of D” transition of HH-085. The dummy insertion pattern is defined in the data stream after the triangular delay lines but before the reordering as shown in (2) 308 of Figure 3. In data stream (2) 308 of Figure 3, from right to left, the rectangle 312 of the “transition data” can be divided into (I-I) blocks and referenced as the 1st block with $M_1^I$ bytes ($M_1$ columns), … , the jth block with $M_j^I$ bytes ($M_j$ columns), … , until the (I-I)th block. Within each block 316, represented as $M_1$, $M_2$, $M_3$, and $M_4$, the data is composed of “slow data” 318 at the top and “fast data” 320 at the bottom. For the jth block, insert $\Delta M_j^I$ dummy bytes. These dummy bytes, represented as “x”, are in $\Delta M_j$ columns, 322, each with j bytes. They are all inserted at the beginning (right-most) of the “slow data” 318 section within each block 316.

[0061] In the exemplary embodiment of Figure 4, the following rules define the dummy insertion pattern for “Increase of D” transition of HH-085. The dummy pattern is defined in the data stream after the triangular delay lines but before the
reordering, as shown in (2) 408 of Figure 4. In data stream (2) 408 of Figure 4, from right to left, the rectangle 412 of the "transition data" can be divided into (I-1) blocks and referenced as the 1st block with \( M_1 \times I \) bytes (\( M_1 \) columns), \( \ldots \), the \( j \)th block with \( M_j \times I \) bytes (\( M_j \) columns), \( \ldots \), until the (I-1)th block. Within each block 416, the data is composed of "fast data" 418 at the top and "slow data" 420 at the bottom. For the \( j \)th block, insert \( \Delta M_j \times (I-j) \) dummy bytes. These dummy bytes are in \( \Delta M_j \) columns, each with (I-j) bytes. They are all inserted at the end (left-most) of the "slow data" 420 section within each block 416.

[0062] Now with reference to Figure 6, for example in data stream (1) 602 the data bytes 604 are "fast data" and the data bytes 606 are "slow data." While in Figure 7, the data bytes 704 are "slow data" and the data bytes 706 are "fast data" in data stream (1) 702.

[0063] Figures 5 through 7 illustrate one exemplary set of dummy patterns illustrating an embodiment of the present input dummy insertion method and system. In the exemplary embodiment of Figure 6, the following rules define the dummy insertion pattern for the "Decrease of D" transition of HH-086. The dummy pattern is defined in the input data stream after dummy insertion (not going through the triangular delay lines yet), as shown in (2) 608 of Figure 6. In data stream (2) 608 of Figure 6, from left to right, starting from the transition end point, the rectangle 612 of the "slow data" can be divided into (I-1) blocks as referenced as the 1st block with \( M'_1 \times I \) bytes (\( M'_1 \) columns), \( \ldots \), the \( j \)th block with \( M'_j \times I \) bytes (\( M'_j \) columns), \( \ldots \), until the (I-1)th block. Within each block 616, represented as \( M_1, M_2, M_3 \), and \( M_4 \), the data is composed of "transition data" 618 at the top and "non-transition data" 620
at the bottom. For the $j^{th}$ block, insert $\Delta M_j \times j$ dummy bytes. These dummy bytes $x$ are in $\Delta M_j$ columns, each with $j$ bytes. The dummy bytes $x$ are all inserted at the beginning (right-most) of the “transition data” section within each block 616.

[0064] In the exemplary embodiment of Figure 7, the following rules define the dummy insertion pattern for “Increase of D” transition of HH-086. The dummy pattern is defined in the input data stream after dummy insertion (not going through the triangular delay lines yet), as shown in (2) 708 of Figure 7. In data stream (2) 708 of Figure 7, from left to right, starting from the transition beginning point, the rectangle 712 of the “slow data” can be divided into (I-1) blocks – the 1$^{st}$ block with $M'_1 \times I$ bytes ($M'_1$ columns), $\ldots$, the $j^{th}$ block with $M'_j \times I$ bytes ($M'_j$ columns), $\ldots$, until the (I-1)$^{th}$ block. Within each block, represented as $M_1$, $M_2$, $M_3$, and $M_4$, the data is composed of “non-transition data” 718 at the top and “transition data” 720 at the bottom. For the $j^{th}$ block, insert $\Delta M_j \times (I-j)$ dummy bytes. These dummy bytes are in $\Delta M_j$ columns, each with (I-j) bytes, and are all inserted at the end (left-most) of the “transition data” section within each block 716.

[0065] The interleaver design so far discussed can handle extreme cases, e.g., when $\Delta M_j < 0$. Above, in the case of the triangular implementation of GCI, we defined $M_j$ for the interleaver with larger $D = \text{Max}(D_1,D_2)$, defined $M'_j$ for the interleaver with smaller $D = \text{Min}(D_1,D_2)$, and defined $\Delta M_j = M_j - M'_j$. From the rules described above with respect to the inventive dummy insertion methods, the number of dummy bytes $x$ inserted is based on $\Delta M_j$. Initially it may appear that the rules require that $\Delta M_j$ be greater than or equal to zero, because “negative” dummy bytes cannot be inserted into the data stream. But with some special handling, the rules can
be extended to handle "negative" $\Delta M_j$ cases. For instance, Table 1 shows an example of how the interleaver design may handle a "negative" $\Delta M_j$.

<table>
<thead>
<tr>
<th>$j$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_j (D_1=18, I=5)$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$M'_j (D_2=17, I=5)$</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>$\Delta M_j$</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1

The following describes conditions under which a "negative" $\Delta M_j$ may happen. First, if $D=M*+I\times x$, then $M_j$ will be either $M$ or $M+1$. Based on the definition of $M_j$ the following results:

$$L_j = \left\lfloor \frac{j \cdot D}{I} \right\rfloor = \left\lfloor \frac{j \cdot (M \cdot I + x)}{I} \right\rfloor = j \cdot M + \left\lfloor \frac{j \cdot x}{I} \right\rfloor; \quad (1)$$

$$L_{j-1} = \left\lfloor \frac{(j-1) \cdot D}{I} \right\rfloor = \left\lfloor \frac{(j-1) \cdot (M \cdot I + x)}{I} \right\rfloor = (j-1) \cdot M + \left\lfloor \frac{(j-1) \cdot x}{I} \right\rfloor; \quad (2)$$

$$M_j = L_j - L_{j-1} = M + \left\lfloor \frac{j \cdot x}{I} \right\rfloor - \left\lfloor \frac{(j-1) \cdot x}{I} \right\rfloor. \quad (3)$$

Since $x < I$, the following results:

$$0 \leq \left\lfloor \frac{j \cdot x}{I} \right\rfloor - \left\lfloor \frac{(j-1) \cdot x}{I} \right\rfloor \leq 1. \quad (4)$$

Therefore, based on Equations (3) and (4):

$$M_j = M \quad \text{or} \quad M + 1 \quad (5)$$

Next, if $D=M*+I+x$ and $D'=M'*+I+x'$ ($D>D'$), then only when $M=M'$ will the negative $\Delta M_j$ happen. Based on the conclusion (5), $M_j$ is either $M$ or $M+1$, $M'_j$ is either $M'$ or $M'+1$. If $M>M'$, then there is no way that $\Delta M_j (=M_j-M'_j)$ could be negative because $\min(M_j) = M \geq M'+1 = \max(M'_j)$. Only where $M=M'$ could $\Delta M_j$ be less than zero.
(when \(M_j=M\) and \(M'_j=M'+1\)). When "negative" \(\Delta M_j\) occurs, then \(M\) must be equal to \(M'\) and there are only three possible \(\Delta M_j\) values: 1, 0, or -1. The "negative" \(\Delta M_j\) may be handled by taking the "negative" dummy bytes as a credit, which may be used later in the process. That is, if \(\Delta M_j\) is negative at the \(j^{th}\) block, then we don't insert any dummy bytes at this block. \(\Delta M_j^{*}j\) dummy bytes are saved as a credit \((\Delta M_j^{*}j = -j, \text{ because } \Delta M_j = -1 \text{ always})\), and used to offset the positive dummy bytes in the following blocks. In other words, if later on in the \((j+i)^{th}\) block we need to insert \(\Delta M_{j+i}^{*}(j+i)\) positive dummy bytes \((\Delta M_{j+i}^{*}(j+i) = j+i, \text{ because a positive } \Delta M_{j+i} \text{ can only be } 1 \text{ if there is a negative } \Delta M_j\)), then it is not necessary to insert the full amount \((j+i)\) and instead only \((j+i)-j=i\) dummy bytes need by inserted. This rule only applies to "decrease of D" cases where in the \(j^{th}\) block, the number of dummy bytes to be inserted is \(\Delta M_j^{*}j\). In the "increase of D" cases where in the \(j^{th}\) block the number of dummy bytes is \(\Delta M_j^{*}(I-j)\), the above rule applies reversely from \((I-1)\) block to the \(1^{st}\) block.

[0067] Figures 8 through 11 illustrate examples of how the rule of negative \(\Delta M_j\) applies to different cases. In Figure 8, the four positive dummy bytes to be inserted at the \(4^{th}\) block are offset by the three negative dummy bytes in the \(3^{rd}\) block thereby leaving no dummy bytes inserted in the \(3^{rd}\) block and only one dummy byte inserted at the \(4^{th}\) block. In Figure 9, the three positive dummy bytes to be inserted at the \(2^{nd}\) block are offset by the two negative dummy bytes in the \(3^{rd}\) block, thereby leaving no dummy bytes at the \(3^{rd}\) clock and only one dummy byte inserted at the \(2^{nd}\) block. In Figure 10, the four positive dummy bytes to be inserted at the \(4^{th}\) block are offset by the three negative dummy bytes in the \(3^{rd}\) block, thereby leaving no dummy
bytes inserted at the 3rd block and only one dummy byte inserted at the 4th block. In Figure 11, the three dummy bytes to be inserted at the 2nd block are offset by the two negative dummy bytes in the 3rd block, thereby leaving no dummy bytes inserted in the 3rd block and only one dummy byte inserted at the 2nd block.

[0068] Figures 12 through 21 illustrate a dynamic interleaver design for seamless depth transition and for implementing the dummy byte insertion patterns for dynamic interleaver configuration for dummy insertion. To change the interleaver depth at run-time, a normal interleaver must stop the input data stream for a time equal to at least the interleaving delay so that the residue data in the interleaver buffer can be fully flushed. After that, the interleaver is changed to the new depth and the input stream is resumed. This “interleaver flush” process is illustrated in Figure 12. The goal is to have the dynamic interleaver overlap the tail 1202 of the old interleaver output 1204 and the head 1206 of the new interleaver output 1208, so that the gap 1210 at the interleaver input stream can be reduced to zero resulting in a seamless transition, as shown in Figure 14.

[0069] During the transition period, as shown in Figures 15 and 16, dummy bytes are inserted to the lower-rate data stream 1502 and 1602, respectively, to accommodate the (de)interleaver input/output speed difference.

[0070] Now with regard to three functions of dummy insertion, i.e., 1) changing the shape of the interleaver buffer to change the interleaving depth; 2) accommodating the (de)interleaver input/output rates to minimize the delay variations; and 3) maintaining or preserving the INP. The following addresses the function of the
dummy insertion in changing the shape of the interleaver buffer to change the interleaver depth on-the-fly. From the implementation point of view, changing the interleaver depth is equivalent to changing the shape of the interleaver buffer. This change is accomplished by inserting the dummy bytes. There are many acceptable dummy insertion patterns. Figure 17 shows just a few. For each delay line of the interleaver buffer, the total number of dummy bytes may be determined by the relative delay line length between the old and the new interleavers. This is common for all the dummy insertion patterns of Figure 17. The difference is in how to insert the dummy bytes, i.e., in one chunk, distributed, or evenly distributed. The decision to choose one pattern over another greatly affects the implementation complexity and how we meet the delay variation and INP requirements.

[0071] The following addresses the function of the dummy insertion in minimizing the delay variations. A perfect “seamless” transition should have zero delay variation and should be transparent to the upper layers, however this is not always necessary for real world applications. Certain dummy insertion patterns can minimize the delay variations. Figure 18 illustrates six different dummy insertion patterns for decreasing depth transition. The dummy bytes x are all inserted into the data pool with lower input speed (the upper portion or triangle), referred to as “slow data.” The data bytes with higher input speed (the lower portion or triangle) referred to as “fast data” are not affected by the dummy insertion so the “fast data” arrive at the deinterleaver as expected.

[0072] Patterns 1(a), 2(a), 3(a) of Figure 18 send out dummy bytes first, and then send out the data bytes. Due to the dummy byte insertion, at the interleaver
output the “slow data” are all delayed more than they should be. For example, the first “Q” byte should be sent out at time $T_0$, but is actually delayed to time $T_1$ ($0 < T_1 - T_0 < \text{InterleaveDelay}$). When these delayed data arrive at the deinterleaver, due to the dummy byte removal, they are brought forward, which compensates the previously added delay. Overall, the dummy byte-affected “slow data” still exit from the deinterleaver on time so there is no delay variation.

Patterns 1(b), 2(b), and 3(b) of Figure 18 send out data bytes first and then send out the dummy bytes. In these patterns, the “slow data” are all sent out earlier than expected, i.e., $T_1 - T_0 < 0$. For some of the “slow data” (e.g., the data in the first row), this looks impossible because they are experiencing a negative delay (being sent out earlier than they are available). In real world application, a negative delay can be compensated by the existing positive delay (buffering) or it may become a positive delay when idle bytes are sent out at the time slots of the unavailable data. Either way, these dummy byte insertion patterns cannot achieve zero delay variation.

Based on the above discussion, patterns 1(a), 2(a), and 3(a) can achieve zero delay variation. Of the three patterns, 1(a) requires the most interleave memory while being the least complex, while 3(a) requires the least interleave memory while being the most complex.

With reference to Figures 19-21, the following addresses the function of the dummy byte insertion in maintaining the INP. During the transition, the positions of the “slow data” are altered by the dummy byte insertion. This inevitably changes the spreading of the interleave codeword. The challenge to maintain the INP is to preserve the spreading of the dummy-affected “slow data” during the transition.
period. First, consider how to insert the dummy so as to preserve the spreading of the interchange codeword.

[0076] Figure 19 illustrates four dummy byte insertion patterns 1-4 and how they affect the spreading of the “Q” codeword. From Figure 19, it is clear that patterns #1 and #2, respectively 1902 and 1904, cause shortened distance of adjacent “Q” bytes, while #3 and #4, respectively 1906 and 1908, preserve the spreading of “Q.” This is more clearly shown in the interleaver input stream (assuming the dummy bytes are inserted at the input) – if Cj is the column number of the jth “Q” byte at the interleaver input stream, then as long as Cj ≥ Cj-1, the spreading of the “Q” codeword is at least preserved or better.

[0077] Based on this guideline and with reference back to the dummy patterns in Figure 18, the ideal codeword spreading is illustrated in Figure 20 as a reference. The codeword spreading of the six patterns of Figure 18 are illustrated in Figure 21, which shows that only patterns 1(b), 2(b) and 3(b) can maintain the INP during the transition, while 1(a), 2(a) and 3(a) cannot.

[0078] In looking at the requirements from a standpoint of maintaining Quality of Service (QoS) during rate adaptation, based on the delay and INP equations of the GCI (Equations 6 and 7), both the delay and INP are inverse-proportional to the line rate. As the line rate changes, the quality of the data stream may be maintained by changing the interleaving depth in proportion to the line rate.

\[
\text{Delay (ms)} = \frac{8 \cdot (D-1) \cdot (I-1)}{\text{LineRate (kb/s)}},
\]

\[
\text{INP (ms)} = \frac{t \cdot 8 \cdot D}{q \cdot \text{LineRate (kb/s)}};
\]
\[ \frac{t}{q} \frac{8 \cdot (D-1)}{\text{LineRate} \text{ (kb/s)}} = \text{INP}_{LB} \] (4)

[0079] In more detail, during rate adaptation the system should both 1) keep the FEC parameters \( N \) and \( R \) unchanged, thus the Reed-Solomon coding gain would not change, and 2) keep the interleaving block size \( I \) unchanged. As the line rate changes, the interleaving depth \((D-1)\) should change in proportion to the line rate. This makes the delay remain constant. Since \( N, R \) and \( I \) are unchanged, the parameters \( t \) and \( q \) \((=N/I)\) are unchanged. To keep \( \text{INP} \) no less than \( \text{INP}_{\text{min}} \) for all possible rates, when selecting the initial FEC/interleaver parameters at the training stage, a degree of over-provisioning is needed to make \( \text{INP}_{LB} \geq \text{INP}_{\text{min}} \) (Eq. 3). Because \((D-1)\) is proportional to the line rate, \( \text{INP}_{LB} \) is constant as the line rate changes. This ensures the real \( \text{INP} \) (Eq. 2) will be larger than \( \text{INP}_{\text{min}} \) at any time.


[0081] SI-090 purports to disclose an “error free” interleaver adaptation method for SRA (Seamless on-line Rate Adaptation) and DRR (Dynamic Rate Repartitioning). The scheme is simple and does not require changes to the current
interleavers. However, it does not ensure "seamless" adaptation of the interleaver in the sense of maintaining constant delay. According to SI-090 and D.130, true seamless on-line rate adaptation would mean no errors, no interruption in data, and no changes in delay and impulse noise protection (INP), both in the steady states and in the transient state. To achieve these on an interleaved latency path, a dynamic interleaver reconfiguration scheme is required to change the interleaving depth on the fly. In one aspect of the invention, a new dynamic interleaver reconfiguration scheme allows seamless on-line rate adaptation with finer control of the rate change and provides near-zero delay variation and preserves INP during the transition. SI-090 purports to provide an error-free rate adaptation method that requires that the transmitter pause the transmission. D.131 describes a dynamic triangular interleaver scheme that allows constant delay on average, but requires large additional memory during "decrease of M", and requires large delay variation during "increase of M".

[0082] The method of SI-090 describes pausing the data stream during the interleaver reconfiguration so as to allow the interleaver buffer to be fully flushed. However, interruption of the data stream is particularly undesirable for data services that are sensitive to "delay variations", e.g., constant bit rate (CBR) audio/video (A/V) streams. SI-090 purports to describe a transition process wherein the line rate is changed to 3/4 of the original line rate and the interleaving depth is decreased accordingly. The data streams are observed at the interleaver input (transition state 1), interleaver output (transition state 2), and at the deinterleaver output (transition state 3) and are purported to result in the interleaving delay remaining the same after the transition. However, the method of SI-090 results in an interruption of the incoming
data during the transition, for a time of at least the interleaving delay, to flush the interleaver buffer. This interruption in data avoids the overlapping between the tail of the old interleaver and the head of the new interleaver and thus ensures the error-free transition to the new configuration.

**[0083]** A purported advantage of the SI-090 method is that it is very simple and does not require any changes to the interleaver. But the interruption in the data is very undesirable for audio/video streaming services. For streaming services, the pause in the data stream can be considered as a positive delay variation, and in the case of SI-090, this variation is as large as the interleaving delay. Because every rate adaptation causes an interruption, as more and more rate adaptation occurs (e.g., video channels switched on and off), the end-to-end delay could accumulate until it can not be recovered by the service application.

**[0084]** In the DRR case, when a user has two video channels subscribed then typically one video channel is always on (the “always-on channel”) while the other video channel is switched on/off very often (the “intermittent channel”). If the DRR technique allows or causes an interruption of service then there is essentially no way to avoid the disturbance to the always-on channel. The reason for this is that every interleaver transformation contributes positive delay variation to the always-on channel. As more and more “on/off” events (therefore more and more DRR events) occur on the intermittent channel, more and more buffering is required until eventually virtually unlimited buffering for the always-on channel is required to compensate for the interruption of the stream. SI-090 is purportedly error free, but it does not provide “seamless” transition in that it does not provide a zero delay...
variation. Accordingly, for "seamless" adaptation of CBR traffic, zero delay variation is desired.

[0085] Turning attention now to D.131, which purports to describe a dynamic interleaver scheme to achieve zero delay variation during DRR. However, in the method of D.131 the adaptation scheme requires extra memory during interleaver transformation. To eliminate interruption during interleaver transformation, the tail of the old interleaver and the head of the new interleaver should be overlapped. During the transition, the interleaver output speed (equal to the line rate) should be kept at the higher speed of the two rates (before and after transition). Dummy bytes are inserted to compensate for the speed difference and to shape the interleaver and the deinterleaver. D.131 illustrates the data streams at the following successive stages: interleaver input (1), after dummy insertion (2), followed by interleaver output (3), and finally deinterleaver output (4). As described, the line rate is changed to 3/4 of the original line rate and the triangular interleaver is changed from $M_1=4$ to $M_2=3$, with interleaving block length I equal to 5.

[0086] According to D.131, it takes $\text{Max}(M_1,M_2)I(I-1)$ bytes to finish the interleaver transformation. During the whole transformation process, $\Delta M*I(I-1)/2$ dummy bytes ($\Delta M=M_1-M_2$) are inserted at the interleaver, and the same amount of dummy bytes, or "dummy bytes," are removed from the deinterleaver. The transition process is synchronized between the interleaver and the deinterleaver. The dummy bytes are inserted and removed in a pre-defined pattern so that the interleaver and the deinterleaver can be transformed to the new shape as expected. When $\Delta M>1$, the whole transition process can be performed as multiple $\Delta M=1$ transitions.
In considering D.131 one must address certain drawbacks, for instance during dummy insertion impulse noise protection (INP) is impaired. During the "decrease of M" transition additional memory is needed to buffer the incoming data at the interleaver. According to D.131, the dummy bytes may not be physically present in the interleaver memory and during the transition the interleaver output speed is higher than the input speed. At the beginning of the transition, most of the outputs are dummy bytes. According to this method, a relatively minor amount of real data is consumed inside the interleaver buffer, while the inputs keep coming. At this stage of the process the interleaver buffer grows, though in the long run the buffer size is decreased.

D.131 describes the process in terms of interleaver buffer draining speed vs. filling speed during the "decrease of M" transition. The filling speed (interleaver input speed) is constant during the whole transition period, which is \((M-1)/M\) of the output speed (in \(\Delta M=1\) case). The buffer draining speed (interleaver output speed minus the dummy insertion speed) is characterized by a "slow start" behavior and reaches full output speed until all the dummy bytes are transmitted. D.131 requires additional memory, which in the worst case is \(I^*(I-1)/2\) bytes.

According to D.131, during the "increase of M" transition a positive delay variation as large as \(I^*(I-1)\) bytes could occur at the deinterleaver output. As described, for a given interleaving codeword, during the transition the first \((I-1)\) bytes of the codeword arrive at the deinterleaver output on time. The last byte of the codeword is delayed by \(I^*(I-1)\) bytes. At the Reed-Solomon decoder output, the whole codeword is delayed by \(I^*(I-1)\) bytes because the decoding cannot output a
single byte until the whole codeword is collected. Further, the delayed byte of the codeword also blocks the following codeword. To compensate for this positive delay variation, “pre-buffering” of size I*(I-1) bytes is needed at the receiver side.

[0090] D.131 describes a dynamic triangular interleaver scheme in terms of an “increase of M from 3 to 4.” During the “decrease of M” transition, the dummy insertion causes impairment to the impulse noise protection. By definition of the convolutional interleaver: for adjacent bytes within an interleaving block (of size I), their distance should be D after interleaving. During the transition period of D.131, since the line rate is at the higher rate “Rate1”, in order to maintain the same INP the distance of adjacent bytes within an interleaving block should be at least D1=M1*I+1. However, due to the dummy insertion the distance is shortened to D2=M2*I+1, which means the INP is dropped to D2/D1 of the original. One disadvantage leading to INP impairment is the insertion of the dummy bytes at the same time and at the end of each delay line. D.131 is purports to be error free and to provide “seamless” transition in that it does not provide a zero delay variation. However, D.131 requires extra memory during transition, is complex to implement, and is applicable on triangular interleaver.

[0091] According to one aspect of the present invention, INP may be preserved throughout transition by spreading the dummy bytes more evenly and not always at the end of each delay line.

[0092] With reference to Figure 22, in one embodiment the present invention provides a dynamic interleaver adaptation scheme wherein a constant delay is
maintained by adapting the interleaving depth parameter M proportional with the rate. This approach is based on a triangular interleaver or general convolutional interleaver ("GCI") with x=1. Figure 22 shows the step-by-step behavior of the new interleaver scheme when M is changed from M₁=4 to M₂=3, with interleaving block length I equal to 5 using the triangular interleaver (or GCI with x=1). This approach can be expanded to GCI as well, e.g., for use in VDSL2.

[0093] Figure 22 illustrates an exemplary embodiment of the interleaver adaptation scheme for seamless DRR. The interleaver adaptation begins at Step (0) 2202 and ends at Step (80) 2212. At Step (0) 2202, no dummy bytes are needed to shape the interleaver. As described in this exemplary embodiment, it takes, for example, M₁*I*(I-1) steps to finish the transformation. Starting from Step (0), the interleaver input speed is switched to the new rate of M₂/M₁ (or simply 3/4) of the original. The interleaver output speed, the line rate, and the deinterleaver input/output speed are not changed until the transformation is completed at Step (80). Dummy bytes x are generated naturally due to the input/output speed difference. The dummy bytes x at interleaver and deinterleaver are effectively "removed" simultaneously at the end of the transformation (Step 80). Removing the dummy bytes does not necessarily mean that they are physically extricated from the delay line, but may also mean that the dummy bytes are simply ignored by skipping the read/write pointers at their locations. Accordingly, the "increase of M" case can be implemented as follows. Before the "increase of M" adaptation starts, the dummy bytes may be pre-allocated in the delay line with pointers skipping them over. After the adaptation the dummy "holes" are filled.
[0094] In accordance with this exemplary embodiment of the invention, no extra memory is required to shape the interleaver. Further, this scheme simplifies implementation for interleavers using circular buffer (RAM) based delay lines. Although this approach uses a non-linear increment of the read/write pointers, the pointer increment is in an orderly fashion, e.g., every $M_1$ bytes, skipping the last $\Delta M$ bytes.

[0095] This exemplary embodiment of the present invention is error free, provides "seamless" transition in that it does provide a zero delay variation, does not require extra memory during transition, is not complex to implement, and is applicable on either triangular or GCI interleavers.

[0096] Now referring to Figures 24 through 27 in describing a second embodiment of the present invention which provides a dynamic interleaver configuration that maintains constant delay during transition. The dynamic interleaver of the second embodiment of the invention provides an optimized dummy insertion pattern that resolves the drawbacks of D.131 and can be extended to the generalized convolutional interleaver (GCI).

[0097] Figures 24 and 25 respectively show how the "decrease of $M$" and "increase of $M$" interleaver transformation are performed using the invention. The dummy bytes $x$ are inserted as illustrated in the figures and as described as follows. Figure 24 illustrates the "decrease of $M$" transition in which for each delay line $\Delta M$ ($\Delta M=M_1-M_2$) dummy bytes are inserted at the beginning of every $M$ outgoing bytes. Figure 25 illustrates the "increase of $M$" transition in which for each delay line $\Delta M$
dummy bytes are inserted at the end of every M outgoing bytes. Since the dummy bytes are inserted more evenly, the issues of D.131 are resolved. For instance, during "decrease of M" transition the additional memory issue is resolved. Only at the first delay line is an additional buffer of M bytes needed. Also, during "increase of M" transition no positive delay variation occurs at the deinterleaver output.

[0098] Although this embodiment of the invention has been described in terms of a triangular interleaver, it may also be extended to a GCI. With finer granularity of interleaving depth, the GCI allows finer control of the rate adaptation. In this respect, it may be more effective to have the dynamic interleaver scheme extended to GCI. The implementation of GCI is similar to the triangular interleaver in that it can be implemented using a near triangular buffer structure with a reordered output. Figures 26 and 27 show how the dynamic interleaver scheme is applied to GCI when the interleaving depth D is changed between 19 and 13.

[0099] The whole transition takes Ceil((D-1)*((I-1)/I))*I bytes, where D is the larger of D1 and D2. The duration is not exactly equal to the interleaving delay. It is the interleaving delay (D-1)*((I-1)/I) rounded up to the next multiple of I. In this way, the beginning and the ending of the transition are still aligned with the interleaving codeword boundary.

[0100] The dummy insertion strategy is similar to that in the triangular case. If we consider the triangular interleaver as a stairway, then there are (I-1) steps and the width of each step is constant, i.e., always M. For each step, we need to insert ΔM dummy bytes to transform the triangular structure from M1 to M2. In the case of GCI,
the interleaver is a near triangular stairway that the width of the \( j \)th step is \( M_j \) \((j=1,2,...,I-1)\), which is not constant. To transform the GCI structure from \( M1_j \) to \( M2_j \), the number of dummy bytes for each step \( \Delta M_j \) is not constant either. At the beginning of the transition, the interleaver input speed is switched from “Rate1” to “Rate2”. During the transition, the interleaver output speed (or the line rate) should be kept at the higher speed of “Rate1” and “Rate2”, and the interleaving depth \( D \) is also kept as the larger of \( D1 \) and \( D2 \). In other words, in the “decrease of \( D \)” transition, the interleaver output speed (line rate) and the interleaving depth are changed at the end of the transition and, therefore, the end of the “decrease of \( D \)” transition should be aligned with the DMT symbol boundary. Further, in the “increase of \( D \)” transition, the interleaver output speed and the interleaving depth are changed at the beginning of the transition and, therefore, the beginning of the “increase of \( D \)” transition should be aligned with the DMT symbol boundary.

[0101] It is easy to see that the dynamic triangular scheme is a special case of the dynamic GCI scheme when \( D=M^*I+1 \). For non-triangular GCI (\( D=M^*I+x \) and \( x \neq 1 \)), because the duration of the transition is \((D-1)^*(I-1)\) rounded up to the next multiple of \( I \), which is not exactly equal to the interleaving delay, there might be a worst-case delay variation of \((I-1)\) bytes at the \( \alpha-\beta \) interface. This delay variation is relatively small when compared to the variations generated by the other parts of the network, so it can be easily compensated. We still consider it as “near zero” delay variation, as defined in D.130.

[0102] It should be well understood, and the invention fully contemplates, that the exemplary embodiments described herein are for illustration purposes only and are
not limiting and that different interleaver implementations may be used. One critical
difference between D.131 and the above described dynamic interleaver dummy byte
insertion configuration is the way that the dummy bytes are inserted during the
transition. During interleaver transformation, the dummy insertion has two functions.
First, to morph the interleaver to the desired shape, thus the interleaving depth D is
changed. To maintain constant delay, a sudden change of D is not feasible because it
causes an interruption of the data. In the other two algorithms, the dummy bytes are
inserted to change the length of each delay line. For each delay line, the total number
of inserted dummy bytes is same in both schemes but with different patterns. In
D.131, the dummy bytes are grouped together, while in the above described
embodiment of the present invention they are distributed more evenly. However, they
both fulfill the function of shaping the interleaver. The second function of dummy
byte insertion is to minimize the delay variation during the transition, because one
desirable goal is "seamlessness." This function requires that: (1) at the interleaver
output, the dummy bytes should be transmitted evenly among the outgoing data (it is
recognized that "perfectly" even is unrealistic and would require a very sophisticated
scheme); and (2) at the interleaver input, the incoming data should not be rejected or
delayed. From the foregoing analysis, we see that this function is not fulfilled by
D.131 at least because it causes a large delay variation and that the present invention
provides these two functions. Although the present dummy insertion methods and
systems of Figs. 2-27 distribute the dummy bytes more evenly than D.131, they do not
ensure that the INP is "perfectly" preserved during GCI transition. However, when
the interleaver is triangular then INP is preserved. In other words, the distance of the
adjacent bytes within an interleaving block (of size I) is not a constant D during the transition. Though the average adjacent distance is D, some of the distance could be shortened to D-2*I (when M is large) or I (when M is small) in the worst case. Accordingly, the worst-case impact to the INP is approximately 2*I/D in percentage drop when M is relatively big, or 1/(t+1) when M is small. If the requirement that transient period INP(t)\geq INP_{min} could be relaxed (the argument here is the transient time is short enough that the INP impairment during this period is negligible), then this scheme is acceptable and prevails over the other schemes on most of the aspects. The dummy insertion methods and systems embodiments of the invention provide an effective dummy byte insertion scheme for applications that require seamless on-line rate changes, e.g., SRA (seamless rate adaptation), DRR (dynamic rate repartitioning) and adaptive INP (impulse noise protection).

[0103] Now referring to Figures 28 through 31 in describing a third embodiment of the present invention which provides a dynamic interleaver configuration that maintains constant delay during transition. The dynamic interleaver of the third embodiment of the invention provides an optimized dummy insertion pattern that resolves the drawbacks of D.131 and can be extended to the generalized convolutional interleaver (GCI). The embodiment allows seamless on-line rate adaptation and dynamic rate repartitioning on an interleaved latency path, with near-zero delay variation and finer control of the rate change. This embodiment also preserves INP value during the transition phase with any GCI.

[0104] Figures 28 and 29 respectively show how the “decrease of M” (in this example from 4 to 3) and “increase of M” (in this example from 3 to 4) interleaver
transformation are performed using this embodiment of the invention. Figures 28 and 29 illustrate the use of this embodiment of the invention in a triangular interleaver whereas Figures 30 and 31 illustrate the use of this embodiment in a GCI. The dummy bytes x are inserted as illustrated in the figures and as described as follows. Figure 28 illustrates the "decrease of M" transition 2800 in which for each delay line ΔM (ΔM=M1-M2) dummy bytes are inserted at the beginning of every M outgoing bytes. Figure 29 illustrates the "increase of M" transition in which for each delay line ΔM dummy bytes are inserted at the end of every M outgoing bytes. Since the dummy bytes are inserted more evenly, the issues of D.131 are resolved. For instance, during the "decrease of M" transition of Fig. 28, the additional memory issue is resolved. Only at the first delay line is an additional buffer of M bytes needed. During the "increase of M" transition of Fig. 29, no positive delay variation occurs at the deinterleaver output. As can be seen in the highlighted transition 2802 of Fig. 28, INP is not impaired at all, i.e., the distance of adjacent "P" bytes remains 21. The dynamic interleaver configuration of Fig. 28 illustrates the example of a decrease of M from 4 to 3 and the configuration of Fig. 29 illustrates the example of an increase of M from 3 to 4.

[0105] Now referring to Figs. 30 and 31, with finer granularity of interleaving depth, the dynamic interleaver configurations extended to GCI allows finer control of rate adaptation and results in a more effective scheme. The implementation of GCI is similar to that of triangular interleaver in that it can be implemented using a near triangular buffer structure with a reordered output. Figures 30 and 31 illustrate in exemplary fashion how the dynamic interleaver scheme may be applied to GCI when
the interleaving depth $D$ is changed between 19 and 13. Figure 30 illustrates a decrease in $D$ from 19 to 13 while Figure 31 illustrates an increase in $D$ from 13 to 19. The whole transition takes $\text{Ceil}((D-1)\times(I-1)/I)\times I$ bytes, where $D$ is the larger of $D1$ and $D2$. The duration is not exactly equal to the interleaving delay, but is the interleaving delay $(D-1)\times(I-1)$ rounded up to the next multiple of $I$. In this manner, the beginning and end of the transition are aligned with the interleaving codeword boundary. The dummy insertion scheme as illustrated is similar to that in the triangular case. Considering the triangular interleaver as a stairway, then there are $(I-1)$ steps and the width of each step is constant — always $M$. For each step $\Delta M$ dummy bytes are inserted to transform the triangular structure from $M1$ to $M2$. In the case of GCI, the interleaver is a near triangular stairway in that the width of the $j_{th}$ step is $M_j$ ($j=1,2,\ldots,I-1$), which is not constant. To transform the GCI structure from $M1_j$ to $M2_j$, the number of dummy bytes for each step $\Delta M_j$ is not constant either. At the beginning of the transition, the interleaver input speed is switched from “Rate1” to “Rate2.” During the transition, the interleaver output speed (or the line rate) is preferably maintained at the higher speed of “Rate1” and “Rate2,” and the interleaving depth $D$ is preferably maintained as the larger of $D1$ and $D2$.

[0106] In the “decrease of $D$” transition, the interleaver output speed (line rate) and the interleaving depth are changed at the end of the transition. Accordingly, the end of the “decrease of $D$” transition should be aligned with the DMT symbol boundary. In the “increase of $D$” transition, the interleaver output speed and the interleaving depth are changed at the beginning of the transition. Accordingly, the beginning of the “increase of $D$” transition should be aligned with the DMT symbol.
boundary. From this it is clear that the dynamic interleaver triangular configuration is a special case of the dynamic interleaver GCI configuration, i.e., when \( D = M^*I + 1 \). With the dummy bytes inserted as described above, the INP is maintained during the transition, i.e., the distance of the adjacent bytes within an interleaving block is still the larger of \( D_1 \) and \( D_2 \). For non-triangular GCI, i.e., when \( D = M^*I + x \) and \( x \neq 1 \), because the duration of the transition is \((D-1)^*(I-1)\) rounded up to the next multiple of \( I \), which is not exactly equal to the interleaving delay, there might be a worst-case delay variation of \((I-1)\) bytes at the \( \alpha-\beta \) interface. This delay variation is relatively small, considered a "near zero" delay variation, when compared to the variations generated by other parts of the network and can be easily compensated.

[0107] Typically there will be design and implementation considerations that vary across applications and suppliers of equipment employing the present invention. The following is provided to further help in the context of implementation. One key difference between D.131 and the present invention is in the manner by which dummy bytes are inserted during transition. During interleaver transformation, the dummy insertion has the following two functions. First, to morph the interleaver to the desired shape, thus the interleaving depth \( D \) is changed. A sudden change of \( D \) causes an interruption of the data and does not allow maintaining a constant delay. In the other two schemes, the dummy bytes are inserted to change the length of each delay line. For each delay line, the total number of inserted dummy bytes is same in both schemes, but with different patterns. In D.131, the dummy bytes are grouped together, while in the present invention, they are distributed more evenly. They both fulfill the function of shaping the interleaver.
The second function of the dummy insertion is to minimize the delay variation during the transition to achieve the goal of "seamlessness." This requires that (1) at the interleaver output, the dummy bytes should be transmitted substantially evenly among the outgoing data ("perfectly" even may not be practical and would require a very sophisticated scheme); and (2) at the interleaver input, the incoming data should not be rejected or delayed. D.131 fails to satisfy this function because it causes a large delay variation. Preferably, in addition to these two constraints the dummy insertion does not impair INP.

As for dummy pattern generation, there are two common ways to generate dummy bytes. The easiest way is to generate the dummy bytes at the interleaver input and physically put the dummy bytes into the interleaver. This has the disadvantage of requiring more interleaving memory. A better although more complex alternative is to generate the dummy bytes at the interleaver output so that they are generated only when they are to be transmitted. In this manner, the dummy bytes do not physically occupy space in the interleaver.

Advantages of this third embodiment of the inventive dynamic interleaver configuration include: providing seamless on-line rate changes, e.g., SRA (seamless rate adaptation); near zero delay variation; DRR (dynamic rate repartitioning); and adaptive INP (impulse noise protection) preservation during transition. The invention is for use in VDSL2 applications.

Figure 23 illustrates an exemplary communications system for use with the present invention in which a transmitter 3007, or transceiver in transmit mode, is in communication with a receiver 3010, or transceiver in receive mode, via
communications network or connection 3008. Both the transmitter 3007 and the receiver 3010 utilize at least one, and likely both, of the dynamic interleaving/de-interleaving configurations of the present invention. Transmitter 3007 receives a data signal 3001 at its input which is delivered to an encoder 3002, which encodes the data and delivers the encoded data to interleaver 3004. The interleaver 3004 operates as described above using at least one of the dynamic interleaver configurations and delivers the encoded interleaved data signal to modulator 3006, which modulates the data signal and outputs the modulated signal from the transmitter along a communications path or channel 3008. The receiver 3010 receives the modulated data signal and inputs the signal into a demodulator 3012, which demodulates the signal. After demodulation, the signal is then processed using de-interleaver 3014 using the dynamic interleaving processes of the present invention as described above. The signal is further processed, decoded, at decoder 3016 and finally delivered to a destination device via output 3018. Memory elements, which may include registers and the like, are associated with the communications system to facilitate processing of the data. Some or all of the processing of the signal and data may be accomplished by processing devices or circuits such as commonly understood and appreciated by those skilled in the art.

[0112] Other embodiments, uses, and advantages of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. The specification and drawings should be considered exemplary only, and the scope of the invention is not intended to be limited to the exemplary embodiments.
Claims:

1. A communications system having an adaptive interleaver scheme comprising:
   an interleaver adapted to receive data and to interleave the received data to
   thereby generate interleaved data; and
   a de-interleaver adapted to receive the interleaved data and to de-interleave the
   received interleaved data to thereby generate data essentially equivalent to the
   received data;
   the interleaver defining a dummy insertion pattern in the received data prior to
   reordering of the received data to thereby form transition data to effect interleaver
   depth (D) and interleaver shape and to accommodate interleaver line rates to minimize
   delay variations and to maintain impulse noise protection (INP).

2. The communications system of claim 1, wherein the interleaver redefines the
   dummy insertion pattern to change the interleaver depth and shape to accommodate
   interleaver line rates to minimize delay variations and to maintain impulse noise
   protection (INP).

3. The communications system of claim 1, wherein the delay and INP are
   inverse-proportional to the interleaver line rate and wherein quality of data is
   maintained by changing the interleaver depth in proportion to the interleaver line rate.

4. The communications system of claim 3, wherein the delay and INP are
   respectively defined by Equations 6 and 7.

5. The communications system of claim 4, wherein during rate adaptation FEC
   parameters N and R are unchanged and interleaving block I is unchanged and
interleaving depth \( (D-1) \) changes in proportion to the interleaver line rate, whereby the delay remains constant.

6. The communications system of claim 5, wherein INP is maintained at no less than a minimum \( \text{INP}_{\text{min}} \) for all possible rates by selecting initial FEC/interleaver parameters at a training stage to achieve a degree of over-provisioning in accordance with Equation 3.

7. The communications system of claim 1, wherein the interleaving performed by the interleaver comprises convolutional interleaving.

8. The communications system of claim 1, wherein the de-interleaving performed by the de-interleaver comprises convolutional de-interleaving.

9. A communications system, comprising:

   - an encoder adapted to receive and encode a signal and generate an encoded signal;
   - an interleaver adapted to receive and to interleave the encoded signal to thereby generate an interleaved encoded signal, the interleaver defining a dummy insertion pattern in the received signal prior to reordering of the received signal to thereby form a transition signal to effect interleaver depth \( (D) \) and interleaver shape and to accommodate interleaver line rates to minimize delay variations and to maintain impulse noise protection (INP);
   - a modulator that modulates the interleaved encoded signal to generate an encoded, interleaved, modulated signal;
   - a demodulator receiving and demodulating the modulated interleaved encoded signal and generating a demodulated interleaved encoded signal;
a de-interleaver adapted to receive and deinterleave the demodulated interleaved encoded signal and generate a deinterleaved demodulated encoded signal; and

a decoder adapted to decode the deinterleaved demodulated encoded signal and generate an output signal essentially equivalent to the received signal.

10. The communications system of claim 9, wherein the interleaver redefines the dummy insertion pattern to change the interleaver depth and shape to accommodate interleaver line rates to minimize delay variations and to maintain impulse noise protection (INP).

11. The communications system of claim 10, wherein the delay and INP are inverse-proportional to the interleaver line rate and wherein quality of signal is maintained by changing the interleaver depth in proportion to the interleaver line rate.

12. The communications system of claim 11, wherein the delay and INP are respectively defined by Equations 6 and 7.

13. The communications system of claim 12, wherein during rate adaptation FEC parameters N and R are unchanged and interleaving block I is unchanged and interleaving depth (D-1) changes in proportion to the interleaver line rate, whereby the delay remains constant.

14. The communications system of claim 13, wherein INP is maintained at no less than a minimum INP_min for all possible rates by selecting initial FEC/interleaver parameters at a training stage to achieve a degree of over-provisioning in accordance with Equation 3.
15. A communications system having an adaptive interleaver scheme comprising:
an interleaver means for receiving data and interleaving the received data to thereby
generate interleaved data; and

a de-interleaver means for receiving the interleaved data and deinterleaving the
received interleaved data to thereby generate data essentially equivalent to the
received data;

the interleaver means defining a dummy insertion pattern in the received data
prior to reordering of the received data to thereby form transition data to effect
interleaver depth (D) and interleaver shape and to accommodate interleaver line rates
to minimize delay variations and to maintain impulse noise protection (INP);

16. The communications system of claim 15, wherein the interleaver means
redefines the dummy insertion pattern to change the interleaver depth and shape to
accommodate interleaver line rates to minimize delay variations and to maintain
impulse noise protection (INP).

17. The communications system of claim 15, wherein the delay and INP are
inverse-proportional to the interleaver line rate and wherein quality of data is
maintained by changing the interleaver depth in proportion to the interleaver line rate.

18. A communications system, comprising:

an encoder adapted to receive and encode a signal and generate an encoded
signal;

an interleaver means for receiving and interleaving the encoded signal to
thereby generate an interleaved encoded signal, the interleaver means defining a
dummy insertion pattern in the received signal prior to reordering of the received
signal to thereby form a transition signal to effect interleaver depth (D) and interleaver shape and to accommodate interleaver line rates to minimize delay variations and to maintain impulse noise protection (INP);

a modulator that modulates the interleaved encoded signal to generate an encoded, interleaved, modulated signal;

a demodulator receiving and demodulating the modulated interleaved encoded signal and generating a demodulated interleaved encoded signal;

a de-interleaver means for receiving and deinterleaving the demodulated interleaved encoded signal and generate a deinterleaved demodulated encoded signal; and

a decoder adapted to decode the deinterleaved demodulated encoded signal and generate an output signal essentially equivalent to the received signal.

19. The communications system of claim 18, wherein the interleaver means redefines the dummy insertion pattern to change the interleaver depth and shape to accommodate interleaver line rates to minimize delay variations and to maintain impulse noise protection (INP).

20. The communications system of claim 18, wherein the delay and INP are inverse-proportional to the interleaver line rate and wherein quality of signal is maintained by changing the interleaver depth in proportion to the interleaver line rate.

21. A method of performing an adaptive interleaver scheme comprising:

receiving data at an interleaver and interleaving the received data to thereby generate interleaved data;
receiving at a deinterleaver the interleaved data and deinterleaving the received
interleaved data to thereby generate data essentially equivalent to the received data;
and

defining at the interleaver a dummy insertion pattern in the received data prior
to reordering of the received data to thereby form transition data to effect interleaver
depth (D) and interleaver shape and to accommodate interleaver line rates to minimize
delay variations and to maintain impulse noise protection (INP).

22. The method of claim 21, further comprising redefining at the interleaver the
dummy insertion pattern to change the interleaver depth and shape to accommodate
interleaver line rates to minimize delay variations and to maintain impulse noise
protection (INP).

23. The communications system of claim 21, wherein the delay and INP are
inverse-proportional to the interleaver line rate and wherein quality of data is
maintained by changing the interleaver depth in proportion to the interleaver line rate.

24. The method of claim 23, wherein the delay and INP are respectively defined
by Equations 6 and 7.

25. The method of claim 24, wherein during rate adaptation FEC parameters N
and R are unchanged and interleaving block I is unchanged and interleaving depth (D-
1) changes in proportion to the interleaver line rate, whereby the delay remains
constant.

26. The method of claim 25, wherein INP is maintained at no less than a minimum
INP_min for all possible rates by selecting initial FEC/interleaver parameters at a
training stage to achieve a degree of over-provisioning in accordance with Equation 3.
27. A communications system having a dynamic interleaver adaptation scheme, the system comprising:

an interleaver adapted to receive data and to interleave the received data to thereby generate interleaved data;

a de-interleaver adapted to receive the interleaved data and to de-interleave the received interleaved data to thereby generate data essentially equivalent to the received data;

the interleaver inserting dummy data in the received data to transform the shape of the interleaver to effect a constant delay by adapting interleaver depth parameter (M) to be proportional with interleaver line rate; and

wherein interleaver input speed is adapted based on a change in the M parameter and interleaver output speed, line rate and deinterleaver speed remain constant until transformation is completed.

28. The communications system of claim 27, wherein the interleaver is a triangular interleaver.

29. The communications system of claim 27, wherein the interleaver is a general convolutional interleaver (GCI).

30. The communications system of claim 29, wherein the general convolutional interleaver (GCI) is implemented using a near triangular buffer structure.

31. The communications system of claim 30, wherein at the beginning of transition the interleaver input speed is switched from a first rate to a second rate, the interleaver output speed is maintained at the higher of the first and second rates and the interleaver depth is maintained at the corresponding larger depth.
32. The communications system of claim 31, wherein in a decrease of D transition the interleaver output speed and the interleaving depth are changed at the end of the transition and the end of the decrease of D transition is aligned with a DMT symbol boundary.

33. The communications system of claim 31, wherein in an increase of D transition the interleaver output speed and the interleaving depth are changed at the beginning of the transition and the beginning of the increase of D transition is aligned with a DMT symbol boundary.

34. The communications system of claim 27, wherein the dummy data are generated in accordance with interleaver input/output speed difference.

35. The communications system of claim 27, wherein the dummy data are effectively removed at the completion of the transformation.

36. The communications system of claim 27, wherein in an increase of M scenario for each delay line ΔM dummy data are inserted at the end of every M outgoing bytes.

37. The communications system of claim 27, wherein the delay variation is a near zero variation.

38. The communications system of claim 27, wherein at the interleaver output the dummy data are transmitted substantially evenly among the outgoing data and at the interleaver input incoming data is not rejected or delayed, and INP is preserved during transition.

39. The communications system of claim 27, wherein dummy data insertion provides seamless rate adaptation, dynamic rate repartitioning and adaptive INP.

40. A communications system, comprising:
an encoder adapted to receive and encode a signal and generate an encoded signal;

an interleaver adapted to receive and to interleave the encoded signal to thereby generate an interleaved encoded signal, the interleaver defining a dummy insertion pattern in the received signal to transform the shape of the interleaver to effect a constant delay by adapt&interleave depth parameter (M) to be proportional with interleaver line rate;

a modulator that modulates the interleaved encoded signal to generate an encoded, interleaved, modulated signal;

a demodulator receiving and demodulating the modulated interleaved encoded signal and generating a demodulated interleaved encoded signal;

a deinterleaver adapted to receive and deinterleave the demodulated interleaved encoded signal and generate a deinterleaved demodulated encoded signal;

wherein interleaver input speed is adapted based on a change in the M parameter and interleaver output speed, line rate and deinterleaver speed remain constant until transformation is completed; and

a decoder adapted to decode the deinterleaved demodulated encoded signal and generate an output signal essentially equivalent to the received signal.

41. The communications system of claim 40, wherein the interleaver comprises a triangular interleaver.

42. The communications system of claim 41, wherein the interleaver comprises a general convolutional interleaver (GCI).
43. The communications system of claim 42, wherein the general convolutional interleaver (GCI) is implemented using a near triangular buffer structure.

44. The communications system of claim 43, wherein at the beginning of transition the interleaver input speed is switched from a first rate to a second rate, the interleaver output speed is maintained at the higher of the first and second rates and the interleaver depth is maintained at the corresponding larger depth.

45. The communications system of claim 44, wherein in a decrease of D transition the interleaver output speed and the interleaving depth are changed at the end of the transition and the end of the decrease of D transition is aligned with a DMT symbol boundary.

46. The communications system of claim 45, wherein in an increase of D transition the interleaver output speed and the interleaving depth are changed at the beginning of the transition and the beginning of the increase of D transition is aligned with a DMT symbol boundary.

47. A communications system having an adaptive interleaver scheme comprising:

   an interleaver means for receiving data and interleaving the received data to thereby generate interleaved data and for inserting dummy data in the received data to transform the shape of the interleaver to effect a constant delay by adapting interleaver depth parameter (M) to be proportional with interleaver line rate; and

   a de-interleaver means for receiving the interleaved data and deinterleaving the received interleaved data to thereby generate data essentially equivalent to the received data;
wherein interleaver input speed is adapted based on a change in the M parameter and interleaver output speed, line rate and deinterleaver speed remain constant until transformation is completed.

48. The communications system of claim 47, wherein the interleaver means comprises a triangular interleaver.

49. The communications system of claim 48, wherein the interleaver means comprises a general convolutional interleaver (GCI).

50. The communications system of claim 49, wherein the general convolutional interleaver (GCI) is implemented using a near triangular buffer structure.

51. The communications system of claim 50, wherein at the beginning of transition the interleaver input speed is switched from a first rate to a second rate, the interleaver output speed is maintained at the higher of the first and second rates and the interleaver depth is maintained at the corresponding larger depth.

52. The communications system of claim 51, wherein in a decrease of D transition the interleaver output speed and the interleaving depth are changed at the end of the transition and the end of the decrease of D transition is aligned with a DMT symbol boundary.

53. The communications system of claim 51, wherein in an increase of D transition the interleaver output speed and the interleaving depth are changed at the beginning of the transition and the beginning of the increase of D transition is aligned with a DMT symbol boundary.

54. A communications system, comprising:
an encoder adapted to receive and encode a signal and generate an encoded signal;

an interleaver means for receiving data and interleaving the received data to thereby generate interleaved data and for inserting dummy data in the received data to transform the shape of the interleaver to effect a constant delay by adapting interleaver depth parameter (M) to be proportional with interleaver line rate;

a modulator that modulates the interleaved encoded signal to generate an encoded, interleaved, modulated signal;

a demodulator receiving and demodulating the modulated interleaved encoded signal and generating a demodulated interleaved encoded signal;

a de-interleaver means for receiving and deinterleaving the demodulated interleaved encoded signal and generate a deinterleaved demodulated encoded signal; and

a decoder adapted to decode the deinterleaved demodulated encoded signal and generate an output signal essentially equivalent to the received signal;

wherein interleaver input speed is adapted based on a change in the M parameter and interleaver output speed, line rate and deinterleaver speed remain constant until transformation is completed.

55. A method of performing an adaptive interleaver scheme comprising:

receiving data at an interleaver and interleaving the received data to thereby generate interleaved data;
receiving at a deinterleaver the interleaved data and deinterleaving the received interleaved data to thereby generate data essentially equivalent to the received data;

inserting at the interleaver a dummy insertion pattern in the received data to transform the shape of the interleaver to effect a constant delay by adapting interleaver depth parameter (M) to be proportional with interleaver line rate; and

adapting interleaver input speed based on a change in the M parameter whereby interleaver output speed, line rate and deinterleaver speed remain constant until transformation is completed.
FIG. 5

1. Input data stream to interleaver
2. After delay lines before reordering
3. Interleaver output data stream
4. Deinterleaver output data stream
5.
The 4 positive dummies to be inserted at the 4th block is offset by the 3 negative dummies in the 3rd block, leaving no dummies inserted at the 3rd block and only 1 dummy inserted at the 4th block.

D = 17, M = 3, x = 2
Delay = 64

Decrease of D (H) = 086

FIG. 8
The 4 positive dummies to be inserted at the 4th block is offset by the 3 negative dummies in the 3rd block, leaving no dummies inserted at the 3rd block and only 1 dummy inserted at the 4th block.

Decrease of D (HH-086)

FIG. 10
The 3 positive dummies to be inserted at the 2th block is offset by the 2 negative dummies inserted at the 3rd block. Only 1 dummy inserted at the 2th block.
### FIG. 15

<table>
<thead>
<tr>
<th>After Transition</th>
<th>During Transition</th>
<th>Before Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
</tr>
<tr>
<td></td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
</tr>
<tr>
<td></td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
</tr>
<tr>
<td></td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
</tr>
<tr>
<td></td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
<td>$BAZYXWVUTSRQPNMLKJIHXGFEDCBA$</td>
</tr>
</tbody>
</table>

**Interleaver input speed**

**Interleaver output speed**

**Deinterleaver input speed**

**Deinterleaver output speed**
### FIG. 16

<table>
<thead>
<tr>
<th>After Transition</th>
<th>During Transition</th>
<th>Before Transition</th>
<th>Interferer input speed</th>
<th>Interferer output speed</th>
<th>Interferer input speed</th>
<th>Interferer output speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>FEDCBZYXWVUTSRQP</td>
<td>Low</td>
<td>High</td>
<td>FEDCBZYXWVUTSRQP</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>High</td>
<td>FEDCBZYXWVUTSRQP</td>
<td>Low</td>
<td>High</td>
<td>FEDCBZYXWVUTSRQP</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>High</td>
<td>FEDCBZYXWVUTSRQP</td>
<td>Low</td>
<td>High</td>
<td>FEDCBZYXWVUTSRQP</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>High</td>
<td>FEDCBZYXWVUTSRQP</td>
<td>Low</td>
<td>High</td>
<td>FEDCBZYXWVUTSRQP</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

1602
FIG. 19

FIG. 20

Ideal Interleaver Output

WV U T S R G
DW V U T S R Q

P O N M L K J H F E D C B A

Interleaver Input

Interleaver Output

1 2 3 4 5 6 7 8 9 10

W V U T S R G

X W Y U T S R Q

P O N M L K J H F E D C B A

1 2 3 4 5 6 7 8 9 10
FIG. 22
FIG. 29
FIG. 30