

(19)



(11)

**EP 2 056 470 B9**

(12)

**CORRECTED EUROPEAN PATENT SPECIFICATION**

(15) Correction information:  
**Corrected version no 1 (W1 B1)**  
**Corrections, see**  
**Claims EN 1**

(51) Int Cl.:  
**H04L 27/26** <sup>(2006.01)</sup> **H04L 5/00** <sup>(2006.01)</sup>  
**H04L 1/00** <sup>(2006.01)</sup> **H03M 13/27** <sup>(2006.01)</sup>

(48) Corrigendum issued on:  
**01.08.2012 Bulletin 2012/31**

(45) Date of publication and mention  
of the grant of the patent:  
**21.03.2012 Bulletin 2012/12**

(21) Application number: **08253375.3**

(22) Date of filing: **17.10.2008**

**(54) Data processing apparatus and method**

Vorrichtung und Verfahren zur Datenverarbeitung

Appareil et procédé de traitement de données

(84) Designated Contracting States:  
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR**  
**HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT**  
**RO SE SI SK TR**

(30) Priority: **30.10.2007 GB 0721271**  
**30.10.2007 GB 0721269**  
**19.11.2007 GB 0722645**  
**20.11.2007 GB 0722728**

(43) Date of publication of application:  
**06.05.2009 Bulletin 2009/19**

(60) Divisional application:  
**11189835.9 / 2 421 166**

(73) Proprietor: **Sony Corporation**  
**Tokyo (JP)**

(72) Inventors:  
• **Taylor, Matthew Paul Athol**  
**Ringwood, Hampshire BH24 1JQ (GB)**  
• **Asanbeng Atungsiri, Samuel**  
**Basingstoke, Hampshire RG21 4BS (GB)**  
• **Wilson, John Nicholas**  
**Hook, Hampshire RG27 9SG (GB)**

(74) Representative: **DeVile, Jonathan Mark**  
**D Young & Co LLP**  
**120 Holborn**  
**London EC1N 2DY (GB)**

(56) References cited:  
**EP-A- 1 463 255 EP-A- 1 463 256**  
**US-A1- 2005 008 084 US-B1- 6 353 900**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

**EP 2 056 470 B9**

**Description****Field of Invention**

5 **[0001]** The present invention relates to data processing apparatus operable to map data symbols received from a predetermined number of sub-carrier signals of an Orthogonal Frequency Division Multiplexed (OFDM) symbol into an output symbol stream.

**[0002]** The present invention also relates to an address generator for use in writing symbols to/reading symbols from an interleaver memory.

10 **[0003]** Embodiments of the present invention can provide an OFDM receiver.

**Background of the Invention**

15 **[0004]** The Digital Video Broadcasting-Terrestrial standard (DVB-T) utilises Orthogonal Frequency Division Multiplexing (OFDM) to communicate data representing video images and sound to receivers via a broadcast radio communications signal. There are known to be two modes for the DVB-T standard which are known as the 2k and the 8k mode. The 2k mode provides 2048 sub-carriers whereas the 8k mode provides 8192 sub-carriers. Similarly for the Digital Video Broadcasting-Handheld standard (DVB-H) a 4k mode has been provided, in which the number of sub-carriers is 4096.

20 **[0005]** In order to improve the integrity of data communicated using DVB-T or DVB-H a symbol interleaver is provided in order to interleave input data symbols as these symbols are mapped onto the sub-carrier signals of an OFDM symbol. Such a symbol interleaver comprises an interleaver memory in combination with an address generator. The address generator generates an address for each of the input symbols, each address indicating one of the sub-carrier signals of the OFDM symbol onto which the data symbol is to be mapped. For the 2k mode and the 8k mode an arrangement has been disclosed in the DVB-T standard for generating the addresses for the mapping. Likewise for the 4k mode of  
25 DVB-H standard, an arrangement for generating addresses for the mapping has been provided and an address generator for implementing this mapping is disclosed in European Patent application 04251667.4. The address generator comprises a linear feed back shift register which is operable to generate a pseudo random bit sequence and a permutation circuit. The permutation circuit permutes the order of the content of the linear feed back shift register in order to generate an address. The address provides an indication of the location in memory in which a data symbol received from one of the  
30 OFDM sub-carriers should be stored in the interleaver memory, in order to map the symbols received from the sub-carrier signals of the OFDM symbol into an output data stream.

**[0006]** In accordance with a further development of the Digital Video Broadcasting-Terrestrial broadcasting standard, known as DVB-T2 there has been proposed that further modes for communicating data be provided.

35 **[0007]** EP1463256 discloses a data processing apparatus which is arranged to map input data symbols to be communication onto a predetermined number of sub-carrier signals of an OFDM symbol. The data processing apparatus includes an interleaver which uses an interleaver memory and an address generator to interleave the input data symbols onto the sub-carriers of the OFDM symbol. The address generator includes a linear feedback shift register and a permutation circuit, which are adapted to generate addresses which can be used to interleave the input data symbols onto the sub-carriers of the OFDM symbols when the number of addresses is approximately 4k. The data processing  
40 apparatus is therefore suitable for use in a transmitter of the 4k mode of DVB-T2.

**[0008]** In US 6,353,900 there is disclosed an interleaver for use with interleaving coded data symbols for transmission and for reception. The interleaver includes an interleaver memory and an address generator. The address generator includes a linear feedback shift register which generates a Pseudo-random Number (PN) sequence from which the addresses can be generated. The taps of the linear feedback shift register are selected to produce a PN sequence of  
45 optimal length.

**Summary of Invention**

50 **[0009]** According to an aspect of the present invention there is provided a data processing apparatus operable to map input symbols to be communicated onto a predetermined number of sub-carrier signals of an Orthogonal Frequency Division Multiplexed (OFDM) symbol. The data processing apparatus comprises an interleaver operable to read-into a memory the predetermined number of data symbols for mapping onto the OFDM sub-carrier signals, and to read-out of the memory the data symbols for the OFDM sub-carriers to effect the mapping. The read-out is in a different order than the read-in, the order being determined from a set of addresses, with the effect that the data symbols are interleaved  
55 on the sub-carrier signals. The set of addresses is determined by an address generator, an address being generated for each of the input symbols to indicate one of the sub-carrier signals onto which the data symbol is to be mapped.

**[0010]** The address generator comprises a linear feedback shift register including a predetermined number of register stages and is operable to generate a pseudo-random bit sequence in accordance with a generator polynomial, and a

permutation circuit and a control unit. The permutation circuit is operable to receive the content of the shift register stages and to permute the bits present in the register stages in accordance with a permutation order to form an address of one of the OFDM sub-carriers.

**[0011]** The control unit is operable in combination with an address check circuit to regenerate an address when a generated address exceeds a predetermined maximum valid address. The data processing apparatus is characterised in that the predetermined maximum valid address is approximately sixteen thousand, the linear feedback shift register has thirteen register stages with a generator polynomial for the linear feedback shift register of

$$R_i'[12] = R_{i-1}'[0] \oplus R_{i-1}'[1] \oplus R_{i-1}'[4] \oplus R_{i-1}'[5] \oplus R_{i-1}'[9] \oplus R_{i-1}'[11], \text{ and the permutation order forms, with an addi-}$$

tional bit, a fourteen bit address  $R_i[n]$  for the i-th data symbol from the bit present in the n-th register stage  $R_i'[n]$  in accordance with the table:

$R_i'$ bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
$R_i$ bit positions	8	4	3	2	0	11	1	5	12	10	6	7	9

**[0012]** Although it is known within the DVB-T standard to provide the 2k mode and the 8k mode, and the DVB-H standard provides a 4k mode, there has been proposed to provide a 16k mode for DVB-T2. Whilst the 8k mode provides an arrangement for establishing a single frequency network with sufficient guard periods to accommodate larger propagation delays between DVB transmitters, the 2k mode is known to provide an advantage in mobile applications. This is because the 2k symbol period is only one quarter of the 8k symbol period, allowing the channel estimation to be more frequently updated allowing the receiver to track the time variation of the channel due to doppler and other effects more accurately. The 2k mode is therefore advantageous for mobile applications.

**[0013]** In order to provide an even sparser deployment of DVB transmitters within a single frequency network, it has been proposed to provide the 16k mode. To implement the 16k mode, a symbol interleaver must be provided for mapping the input data symbols onto the sub-carrier signals of the OFDM symbol.

**[0014]** Embodiments of the present invention can provide a data processing apparatus operable as a symbol interleaver for mapping data symbols to be communicated on an OFDM symbol, having substantially sixteen thousand sub-carrier signals. In one embodiment the number of sub-carrier signals maybe a value substantially between twelve thousand and sixteen thousand three hundred and eighty four. Furthermore, the OFDM symbol may include pilot sub-carriers, which are arranged to carry known symbols, and the predetermined maximum valid address depends on a number of the pilot sub-carrier symbols present in the OFDM symbol. As such the 16k mode can be provided for example for a DVB standard, such as DVB-T2, DVB-T or DVB-H.

**[0015]** Mapping data symbols to be transmitted onto the sub-carrier signals of an OFDM symbol, where the number of sub-carrier signals is approximately sixteen thousand, represents a technical problem requiring simulation analysis and testing to establish an appropriate generator polynomial for the linear feedback shift register and the permutation order. This is because the mapping requires that the symbols are interleaved onto the sub-carrier signals with the effect that successive symbols from the input data stream are separated in frequency by a greatest possible amount in order to optimise the performance of error correction coding schemes.

**[0016]** Error correction coding schemes such as LDPC/BCH coding, which has been proposed for DVB-T2 perform better when noise and degradation of the symbol values resulting from communication is un-correlated. Terrestrial broadcast channels may suffer from correlated fading in both the time and the frequency domains. As such by separating encoded symbols on to different sub-carrier signals of the OFDM symbol by as much as possible, the performance of error correction coding schemes can be increased.

**[0017]** As will be explained, it has been discovered from simulation performance analysis that the generator polynomial for the linear feed back shift register in combination with the permutation circuit order indicated above, provides a good performance. Furthermore, by providing an arrangement which can implement address generating for each of the 2k mode, the 4k mode and the 8k mode by changing the taps of the generator polynomial for the linear feed back shift register and the permutation order, a cost effective implementation of the symbol interleaver for the 16k mode can be provided. Furthermore, a transmitter and a receiver can be changed between the 2k mode, 4k mode, 8k mode and the 16k mode by changing the generator polynomial and the permutation orders. This can be effected in software (or by the embedded signalling) whereby a flexible implementation is provided.

**[0018]** The additional bit, which is used to form the address from the content of the linear feedback shift register, may be produced by a toggle circuit, which changes from 1 to 0 for each address, so as to reduce a likelihood that if an address exceeds the predetermined maximum valid address, then the next address will be a valid address. In one example the additional bit is the most significant bit.

[0019] In one example the above permutation code is used to generate the addresses for performing the interleaving for successive OFDM symbols. In other examples, the above permutation code is one of a plurality of permutation codes which are changed so as to reduce a possibility that successive or data bits which are close in order in an input data stream are mapped onto the same sub-carrier of an OFDM symbol. In one example, a different permutation code is used between performing the interleaving for successive OFDM symbols. The use of different permutation codes for successive OFDM symbols can provide an advantage where the data processing apparatus is operable to interleave the data symbols received from the sub-carrier signals of each of the OFDM symbols only by reading in the data symbols into the memory in an order determined in accordance with the set of addresses generated by the address generator and reading out the data symbols from the memory in a sequential order.

[0020] Various aspects and features of the present invention are defined in the appended claims. Further aspects of the present invention include a method of mapping symbols received from a predetermined number of sub-carrier signals of an Orthogonal Frequency Division Multiplexed (OFDM) symbol into an output symbol stream, as well as a receiver.

### **Brief Description of Drawings**

[0021] Embodiments of the present invention will now be described by way of example only with reference to the accompanying drawings, wherein like parts are provided with corresponding reference numerals, and in which:

Figure 1 is a schematic block diagram of an OFDM transmitter which may be used, for example, with the DVB-T2 standard;

Figure 2 is a schematic block diagram of parts of the transmitter shown in Figure 1 in which a symbol mapper and a frame builder illustrate the operation of an interleaver;

Figure 3 is a schematic block diagram of the symbol interleaver shown in Figure 2;

Figure 4 is a schematic block diagram of an interleaver memory shown in Figure 3 and the corresponding symbol de-interleaver in the receiver;

Figure 5 is a schematic block diagram of an address generator shown in Figure 3 for the 16k mode;

Figure 6(a) is a diagram illustrating results for an interleaver using the address generator shown in Figure 5 for even OFDM symbols and Figure 6(b) is a diagram illustrating design simulation results for odd OFDM symbols, whereas Figure 6(c) is a diagram illustrating comparative results for an address generator using a different permutation code for even and Figure 6(d) is a corresponding diagram for odd OFDM symbols;

Figure 7 is a schematic block diagram of an OFDM receiver which may be used, for example, with the DVB-T2 standard;

Figure 8 is a schematic block diagram of a symbol de-interleaver which appears in Figure 7;

Figure 9(a) is diagram illustrating results for an interleaver using the address generator shown in Figure 5 for even OFDM symbols and Figure 9(b) is a diagram illustrating results for odd OFDM symbols. Figures 9(a) and 9(b) show plots of the distance at the interleaver output of sub-carriers that were adjacent at the interleaver input;

Figure 10 provides a schematic block diagram of the symbol interleaver shown in Figure 3, illustrating an operating mode in which interleaving is performed in accordance with an odd interleaving mode only; and

Figure 11 provides a schematic block diagram of the symbol de-interleaver shown in Figure 8, illustrating the operating mode in which interleaving is performed in accordance with the odd interleaving mode only.

### **Description of Preferred Embodiments**

[0022] It has been proposed that the number of modes, which are available within the DVB-T2 standard should be extended to include a 1k mode, a 16k mode and a 32k mode. The following description is provided to illustrate the operation of a symbol interleaver in accordance with the present technique, although it will be appreciated that the symbol interleaver can be used with other modes and other DVB standards.

[0023] Figure 1 provides an example block diagram of a Coded OFDM transmitter which may be used for example to transmit video images and audio signals in accordance with the DVB-T2 standard. In Figure 1 a program source generates data to be transmitted by the COFDM transmitter. A video coder 2, and audio coder 4 and a data coder 6 generate video, audio and other data to be transmitted which are fed to a program multiplexer 10. The output of the program multiplexer 10 forms a multiplexed stream with other information required to communicate the video, audio and other data. The multiplexer 10 provides a stream on a connecting channel 12. There may be many such multiplexed streams which are fed into different branches A, B etc. For simplicity, only branch A will be described.

[0024] As shown in Figure 1 a COFDM transmitter 20 receives the stream at a multiplexer adaptation and energy dispersal block 22. The multiplexer adaptation and energy dispersal block 22 randomises the data and feeds the appropriate data to a forward error correction encoder 24 which performs error correction encoding of the stream. A bit interleaver 26 is provided to interleave the encoded data bits which for the example of DVB-T2 is the LDPC/BCH encoder

output. The output from the bit interleaver 26 is fed to a bit into constellation mapper 28, which maps groups of bits onto a constellation point, which is to be used for conveying the encoded data bits. The outputs from the bit into constellation mapper 28 are constellation point labels that represent real and imaginary components. The constellation point labels represent data symbols formed from two or more bits depending on the modulation scheme used. These will be referred to as data cells. These data cells are passed through a time-interleaver 30 whose effect is to interleave data cells resulting from multiple LDPC code words.

**[0025]** The data cells are received by a frame builder 32, with data cells produced by branch B etc in Figure 1, via other channels 31. The frame builder 32 then forms many data cells into sequences to be conveyed on COFDM symbols, where a COFDM symbol comprises a number of data cells, each data cell being mapped onto one of the sub-carriers. The number of sub-carriers will depend on the mode of operation of the system, which may include one of 1k, 2k, 4k, 8k, 16k or 32k, each of which provides a different number of sub-carriers according, for example to the following table:

Number of Sub-carriers Adapted from DVB-T/H	
Mode	Sub-carriers
1K	756
2K	1512
4K	3024
8K	6048
16K	12096
32K	24192

**[0026]** Thus in one example, the number of sub-carriers for the 16k mode is twelve thousand and ninety six. For the DVB-T2 system, the number of sub-carriers per OFDM symbol can vary depending upon the number of pilot and other reserved carriers. Thus, in DVB-T2, unlike in DVB-T, the number of sub-carriers for carrying data is not fixed. Broadcasters can select one of the operating modes from 1k, 2k, 4k, 8k, 16k, 32k each providing a range of sub-carriers for data per OFDM symbol, the maximum available for each of these modes being 1024, 2048, 4096, 8192, 16384, 32768 respectively. In DVB-T2 a physical layer frame is composed of many OFDM symbols. Typically the frame starts with one or more preamble or P2 OFDM symbols, which are then followed by a number payload carrying OFDM symbols. The end of the physical layer frame is marked by a frame closing symbols. For each operating mode, the number of sub-carriers may be different for each type of symbol. Furthermore, this may vary for each according to whether bandwidth extension is selected, whether tone reservation is enabled and according to which pilot sub-carrier pattern has been selected. As such a generalisation to a specific number of sub-carriers per OFDM symbol is difficult. However, the frequency interleaver for each mode can interleave any symbol whose number of sub-carriers is smaller than or the same as the maximum available number of sub-carriers for the given mode. For example, in the 1k mode, the interleaver would work for symbols with the number of sub-carriers being less than or equal to 1024 and for 16k mode, with the number of sub-carriers being less than or equal to 16384.

**[0027]** The sequence of data cells to be carried in each COFDM symbol is then passed to the symbol interleaver 33. The COFDM symbol is then generated by a COFDM symbol builder block 37 which introduces pilot and synchronising signals fed from a pilot and embedded signal former 36. An OFDM modulator 38 then forms the OFDM symbol in the time domain which is fed to a guard insertion processor 40 for generating a guard interval between symbols, and then to a digital to analogue convertor 42 and finally to an RF amplifier within an RF frontend 44 for eventual broadcast by the COFDM transmitter from an antenna 46.

### **Providing a 16k Mode**

**[0028]** To create a new 16K mode, several elements are to be defined, one of which is the 16K symbol interleaver 33. The bit to constellation mapper 28, symbol interleaver 33 and the frame builder 32 are shown in more detail in Figure 2.

**[0029]** As explained above, the present invention provides a facility for providing a quasi-optimal mapping of the data symbols onto the OFDM sub-carrier signals. According to the example technique the symbol interleaver is provided to effect the optimal mapping of input data symbols onto COFDM sub-carrier signals in accordance with a permutation code and generator polynomial, which has been verified by simulation analysis.

**[0030]** As shown in Figure 2 a more detailed example illustration of the bit to symbol constellation mapper 28 and the frame builder 32 is provided to illustrate an example embodiment of the present technique. Data bits received from the bit interleaver 26 via a channel 62 are grouped into sets of bits to be mapped onto a data cell, in accordance with a

number of bits per symbol provided by the modulation scheme. The groups of bits, which forms a data word, are fed in parallel via data channels 64 to a mapping processor 66. The mapping processor 66 then selects one of the data symbols, in accordance with a pre-assigned mapping. The constellation point, is represented by a real and an imaginary component that is provided to the output channel 29 as one of a set of inputs to the frame builder 32.

**[0031]** The frame builder 32 receives the data cells from the bit to constellation mapper 28 through channel 29, together with data cells from the other channels 31. After building a frame of many COFDM cell sequences, the cells of each COFDM symbol are then written into an interleaver memory 100 and read out of the interleaver memory 100 in accordance with write addresses and read addresses generated by an address generator 102. According to the write-in and read-out order, interleaving of the data cells is achieved, by generating appropriate addresses. The operation of the address generator 102 and the interleaver memory 100 will be described in more detail shortly with reference to Figures 3, 4 and 5. The interleaved data cells are then combined with pilot and synchronisation symbols received from the pilot and embedded signalling former 36 into an OFDM symbol builder 37, to form the COFDM symbol, which is fed to the OFDM modulator 38 as explained above.

## **Interleaver**

**[0032]** Figure 3 provides an example of parts of the symbol interleaver 33, which illustrates the present technique for interleaving symbols. In Figure 3 the input data cells from the frame builder 32 are written into the interleaver memory 100. The data cells are written into the interleaver memory 100 according to a write address fed from the address generator 102 on channel 104, and read out from the interleaver memory 100 according to a read address fed from the address generator 102 on a channel 106. The address generator 102 generates the write address and the read address as explained below, depending on whether the COFDM symbol is odd or even, which is identified from a signal fed from a channel 108, and depending on a selected mode, which is identified from a signal fed from a channel 110. As explained, the mode can be one of a 1k mode, 2k mode, 4k mode, 8k mode, 16k mode or a 32k mode. As explained below, the write address and the read address are generated differently for odd and even symbols as explained with reference to Figure 4, which provides an example implementation of the interleaver memory 100.

**[0033]** In the example shown in Figure 4, the interleaver memory is shown to comprise an upper part 100 illustrating the operation of the interleaver memory in the transmitter and a lower part 340, which illustrates the operation of the de-interleaver memory in the receiver. The interleaver 100 and the de-interleaver 340 are shown together in Figure 4 in order to facilitate understanding of their operation. As shown in Figure 4 a representation of the communication between the interleaver 100 and the de-interleaver 340 via other devices and via a transmission channel has been simplified and represented as a section 140 between the interleaver 100 and the de-interleaver 340. The operation of the interleaver 100 is described in the following paragraphs:

**[0034]** Although Figure 4 provides an illustration of only four input data cells onto an example of four sub-carrier signals of a COFDM symbol, it will be appreciated that the technique illustrated in Figure 4 can be extended to a larger number of sub-carriers such as 756 for the 1k mode 1512 for the 2k mode, 3024 for the 4k mode and 6048 for the 8k mode, 12096 for the 16k mode and 24192 for the 32k mode.

**[0035]** The input and output addressing of the interleaver memory 100 shown in Figure 4 is shown for odd and even symbols. For an even COFDM symbol the data cells are taken from the input channel 77 and written into the interleaver memory 124.1 in accordance with a sequence of addresses 120 generated for each COFDM symbol by the address generator 102. The write addresses are applied for the even symbol so that as illustrated interleaving is effected by the shuffling of the write-in addresses. Therefore, for each interleaved symbol  $y(h(q)) = y'(q)$ .

**[0036]** For odd symbols the same interleaver memory 124.2 is used. However, as shown in Figure 4 for the odd symbol the write-in order 132 is in the same address sequence used to read out the previous even symbol 126. This feature allows the odd and even symbol interleaver implementations to only use one interleaver memory 100 provided the read-out operation for a given address is performed before the write-in operation. The data cells written into the interleaver memory 124 during odd symbols are then read out in a sequence 134 generated by the address generator 102 for the next even COFDM symbol and so on. Thus only one address is generated per symbol, with the read-in and write-out for the odd/even COFDM symbol being performed contemporaneously.

**[0037]** In summary, as represented in Figure 4, once the set of addresses  $H(q)$  has been calculated for all active sub-carriers, the input vector  $Y' = (y_0', y_1', y_2', \dots, y_{N_{\max}-1}')$  is processed to produce the interleaved vector  $Y = (y_0, y_1, y_2, \dots, y_{N_{\max}-1})$  defined by:

$$y_{H(q)} = y'_q \text{ for even symbols for } q = 0, \dots, N_{\max} - 1$$

$$y_q = y'_{H(q)} \text{ for odd symbols for } q = 0, \dots, N_{\max} - 1$$

**[0038]** In other words, for even OFDM symbols the input words are written in a permuted way into a memory and read back in a sequential way, whereas for odd symbols, they are written sequentially and read back permuted. In the above case, the permutation  $H(q)$  is defined by the following table:

Table 1: permutation for simple case where Nmax =4

q	0	1	2	3
H (q)	1	3	0	2

**[0039]** As shown in Figure 4, the de-interleaver 340 operates to reverse the interleaving applied by the interleaver 100, by applying the same set of addresses as generated by an equivalent address generator, but applying the write-in and read-out addresses in reverse. As such, for even symbols, the write-in addresses 342 are in sequential order, whereas the read out address 344 are provided by the address generator. Correspondingly, for the odd symbols, the write-in order 346 is determined from the set of addresses generated by the address generator, whereas read out 348 is in sequential order.

#### Address Generation for the 16k Mode

**[0040]** A schematic block diagram of the algorithm used to generate the permutation function H(q) is represented in Figure 5 for the 16K mode.

**[0041]** An implementation of the address generator 102 for the 16k mode is shown in Figure 5. In Figure 5 a linear feed back shift register is formed by thirteen register stages 200 and a xor-gate 202 which is connected to the stages of the shift register 200 in accordance with a generator polynomial. Therefore, in accordance with the content of the shift register 200 a next bit of the shift register is provided from the output of the xor-gate 202 by xoring the content of shift registers R[0], R[1], R[4], R[5], R[9], R[11] according to the generator polynomial:

$$R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$$

**[0042]** According to the generator polynomial a pseudo random bit sequence is generated from the content of the shift register 200. However, in order to generate an address for the 16k mode as illustrated, a permutation circuit 210 is provided which effectively permutes the order of the bits within the shift register 200.1 from an order  $R'_i[n]$  to an order  $R_i[n]$  at the output of the permutation circuit 210. Thirteen bits from the output of the permutation circuit 210 are then fed on a connecting channel 212 to which is added a most significant bit via a channel 214 which is provided by a toggle circuit 218. A fourteen bit address is therefore generated on channel 212. However, in order to ensure the authenticity of an address, an address check circuit 216 analyses the generated address to determine whether it exceeds a predetermined maximum value. The predetermined maximum value may correspond to the maximum number of sub-carrier signals, which are available for data symbols within the COFDM symbol, available for the mode which is being used. However, the interleaver for the 16k mode may also be used for other modes, so that the address generator 102 may also be used for the 2k mode, 4k mode, 8k mode, 16k mode and the 32k mode, by adjusting accordingly the number of the maximum valid address.

**[0043]** If the generated address exceeds the predetermined maximum value then a control signal is generated by the address check unit 216 and fed via a connecting channel 220 to a control unit 224. If the generated address exceeds the predetermined maximum value then this address is rejected and a new address regenerated for the particular symbol.

**[0044]** For the 16k mode, an  $(N_r - 1)$  bit word  $R'_i$  is defined, with  $N_r = \log_2 M_{\max}$ , where  $M_{\max} = 16384$  using a LFSR (Linear Feedback Shift Register).

**[0045]** The polynomials used to generate this sequence is:

$$\text{16K mode: } R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$$

where i varies from 0 to  $M_{\max} - 1$

**[0046]** Once one  $R'_i$ , word has been generated, the  $R'_i$ , word goes through a permutation to produce another  $(N_r - 1)$  bit word called  $R_i$ .  $R_i$  is derived from  $R'_i$  by the bit permutations given as follows:

Bit permutation for the 16K mode

$R'_i$ bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
$R_i$ bit positions	8	4	3	2	0	11	1	5	12	10	6	7	9

[0047] As an example, this means that for the mode 16K, the bit number 12 of  $R_i$  is sent in bit position number 8 of  $R_i$ .  
 [0048] The address  $H(q)$  is then derived from  $R_i$  through the following equation:

$$H(q) = (i \bmod 2) \cdot 2^{N_r-1} + \sum_{j=0}^{N_r-2} R_i(j) \cdot 2^j$$

[0049] The  $(i \bmod 2) \cdot 2^{N_r-1}$  part of the above equation is represented in Figure 5 by the toggle block T 218.

[0050] An address check is then performed on  $H(q)$  to verify that the generated address is within the range of acceptable addresses: if  $(H(q) < N_{\max})$ , where  $N_{\max} = 12096$  for example in the 16K mode, then the address is valid. If the address is not valid, the control unit is informed and it will try to generate a new  $H(q)$  by incrementing the index  $i$ .

[0051] The role of the toggle block is to make sure that we do not generate an address exceeding  $N_{\max}$  twice in a row. In effect, if an exceeding value was generated, this means that the MSB (i.e. the toggle bit) of the address  $H(q)$  was one. So the next value generated will have a MSB set to zero, insuring to produce a valid address.

[0052] The following equations sum up the overall behaviour and help to understand the loop structure of this algorithm:

```

q = 0;
for (i = 0; i < M_max; i = i + 1)
{
    H(q) = (i mod 2) · 2Nr-1 + ∑j=0Nr-2 Ri(j) · 2j;
    if (H(q) < N_max) q = q + 1; }
    
```

[0053] As will be explained shortly, in one example of the address generator, the above mentioned permutation code is used for generating addresses for all OFDM symbols. In another example, the permutation codes may be changed between symbols, with the effect that a set of permutation codes are cycled through for successive OFDM symbols. To this end, the control lines 108, 110 provide an indication as to whether the OFDM symbol is odd or even are used to select the permutation code. This example mode in which a plurality of permutation codes are cycled through is particularly appropriate for the example in which the odd interleaver only is used, which will be explained later. A signal indicating that a different permutation code should be used is provided via a control channel 111. In one example the possible permutation codes are pre-stored in the permutation code circuit 210. In another example, the control unit 224 supplies the new permutation code to be used for an OFDM symbol.

#### Analysis Supporting the Address Generator for the 16k Mode

[0054] The selection of the polynomial generator and the permutation code explained above for the address generator 102 for the 16k mode has been identified following simulation analysis of the relative performance of the interleaver. The relative performance of the interleaver has been evaluated using a relative ability of the interleaver to separate successive symbols or an "interleaving quality". As mentioned above, effectively the interleaving must perform for both odd and even symbols, in order to use a single interleaver memory. The relative measure of the interleaver quality is determined by defining a distance  $D$  (in number of sub-carriers). A criterion  $C$  is chosen to identify a number of sub-carriers that are at distance  $\leq D$  at the output of the interleaver that were at distance  $\leq D$  at the input of the interleaver, the number of sub-carriers for each distance  $D$  then being weighted with respect to the relative distance. The criterion  $C$  is evaluated for both odd and even COFDM symbols. Minimising  $C$  produces a superior quality interleaver.

$$C = \sum_{d=1}^{d=D} N_{\text{even}}(d) / d + \sum_{d=1}^{d=D} N_{\text{odd}}(d) / d$$

where:  $N_{\text{even}}(d)$  and  $N_{\text{odd}}(d)$  are number of sub-carriers in an even and odd symbol respectively at the output of the interleaver that remain within  $d$  sub-carrier spacing of each other.

[0055] Analysis of the interleaver identified above for the 16k mode for a value of  $D = 5$  is shown in Figure 6(a) for



the even COFDM symbols and in Figure 6(b) for the odd COFDM symbol. According to the above analysis, the value of C for the permutation code identified above for the 16k mode produced a value of  $C = 22.43$ , that the weighted number of sub-carriers with symbols which are separated by five or less in the output according to the above equation was 22.43.

[0056] A corresponding analysis is provided for an alternative permutation code for even COFDM symbols in Figure 6(c) for odd COFDM symbols in Figure 6(d). As can be seen in comparison to the results illustrated in Figures 6(a) and 6(b), there are more components present which represent symbols separated by small distances such as  $D = 1$ , and  $D = 2$ , when compared with the results shown in Figure 6(a) and 6(b), illustrating that the permutation code identified above for the 16k mode symbol interleaver produces a superior quality interleaver.

#### Alternative Permutation Codes

[0057] The following nine alternative possible codes ( $[n]R_i$  bit positions, where  $n = 1$  to 9) have been found to provide a symbol interleaver with a good quality as determined by the criterion C identified above.

$R_i$ bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
[1] $R_i$ bit positions	7	12	5	8	9	1	2	3	4	10	6	11	0
[2] $R_i$ bit positions	8	5	4	9	2	3	0	1	6	11	7	12	10
[3] $R_i$ bit positions	7	5	6	9	11	2	3	0	8	4	1	12	10
[4] $R_i$ bit positions	11	5	10	4	2	1	0	7	12	8	9	6	3
[5] $R_i$ bit positions	3	9	4	10	0	6	1	5	8	11	7	2	12
[6] $R_i$ bit positions	4	6	3	2	0	7	1	5	8	10	12	9	11
[7] $R_i$ bit positions	10	4	3	2	1	8	0	6	7	9	11	5	12
[8] $R_i$ bit positions	10	4	11	3	7	1	5	0	2	12	8	6	9
[9] $R_i$ bit positions	2	4	11	9	0	10	1	7	8	6	12	3	5
Bit permutation for the 16K mode													

#### Receiver

[0058] Figure 7 provides an example illustration of a receiver which may be used with the present technique. As shown in Figure 7, a COFDM signal is received by an antenna 300 and detected by a tuner 302 and converted into a digital form by an analogue-to-digital converter 304. A guard interval removal processor 306 removes the guard interval from a received COFDM symbol, before the data is recovered from the COFDM symbol using a Fast Fourier Transform (FFT) processor 308 in combination with a channel estimator and correction 310 in co-operation with an embedded-signalling decoding unit 311, in accordance with known techniques. The demodulated data is recovered from a mapper 312 and fed to a symbol de-interleaver 314, which operates to effect the reverse mapping of the received data symbol to regenerate an output data stream with the data de-interleaved.

[0059] The symbol de-interleaver 314 is formed from a data processing apparatus as shown in Figure 7 with an interleaver memory 540 and an address generator 542. The interleaver memory is as shown in Figure 4 and operates as already explained above to effect de-interleaving by utilising sets of addresses generated by the address generator 542. The address generator 542 is formed as shown in Figure 8 and is arranged to generate corresponding addresses to map the data symbols recovered from each COFDM sub-carrier signals into an output data stream.

[0060] The remaining parts of the COFDM receiver shown in Figure 7 are provided to effect error correction decoding 318 to correct errors and recover an estimate of the source data.

[0061] One advantage provided by the present technique for both the receiver and the transmitter is that a symbol interleaver and a symbol de-interleaver operating in the receivers and transmitters can be switched between the 1k, 2k, 4k, 8k, 16k and the 32k mode by changing the generator polynomials and the permutation order. Hence the address generator 542 shown in Figure 8 includes an input 544, providing an indication of the mode as well as an input 546 indicating whether there are odd/even COFDM symbols. A flexible implementation is thereby provided because a symbol interleaver and de-interleaver can be formed as shown in Figures 3 and 8, with an address generator as illustrated in either of Figures 5. The address generator can therefore be adapted to the different modes by changing to the generator polynomials and the permutation orders indicated for each of the modes. For example, this can be effected using a software change. Alternatively, in other embodiments, an embedded signal indicating the mode of the DVB-T2 trans-

mission can be detected in the receiver in the embedded-signalling processing unit 311 and used to configure automatically the symbol de-interleaver in accordance with the detected mode.

### **Optimal Use of Odd Interleavers**

**[0062]** As shown in Figure 4, two symbol interleaving processes, one for even COFDM symbols and one for odd COFDM symbols allows the amount of memory used during interleaving to be reduced. In the example shown in Figure 4, the write in order for the odd symbol is the same as the read out order for the even symbol therefore, while an odd symbol is being read from the memory, an even symbol can be written to the location just read from; subsequently, when that even symbol is read from the memory, the following odd symbol can be written to the location just read from.

**[0063]** As mentioned above, during an experimental analysis of the performance of the interleavers (using criterion C as defined above) and for example shown in Figure 9(a) and Figure 9(b) it has been discovered that the interleaving schemes designed for the 2k and 8k symbol interleavers for DVB-T and the 4k symbol interleaver for DVB-H work better for odd symbols than even symbols. Thus from performance evaluation results of the interleavers, for example, as illustrated by Figures 9(a) and 9(b) have revealed that the odd interleavers work better than the even interleavers. This can be seen by comparing Figure 9(a) which shows results for an interleaver for even symbols and Figure 6(b) illustrating results for odd symbols: it can be seen that the average distance at the interleaver output of sub-carriers that were adjacent at the interleaver input is greater for an interleaver for odd symbols than an interleaver for even symbols.

**[0064]** As will be understood, the amount of interleaver memory required to implement a symbol interleaver is dependent on the number of data symbols to be mapped onto the COFDM carrier symbols. Thus a 16k mode symbol interleaver requires half the memory required to implement a 32k mode symbol interleaver and similarly, the amount of memory required to implement an 8k symbol interleaver is half that required to implement a 16k interleaver. Therefore a transmitter or receiver which is arranged to implement a symbol interleaver of a mode, which sets the maximum number of data symbols which can be carried per OFDM symbol, then that receiver or transmitter will include sufficient memory to implement two odd interleaving processes for any other mode, which provides half or smaller than half the number of sub-carriers per OFDM symbol in that given maximum mode. For example a receiver or transmitter including a 32K interleaver will have enough memory to accommodate two 16K odd interleaving processes each with their own 16K memory.

**[0065]** Therefore, in order to exploit the better performance of the odd interleaving processes, a symbol interleaver capable of accommodating multiple modulation modes can be arranged so that only an odd symbol interleaving process is used if in a mode which comprises half or less than half of the number of sub-carriers in a maximum mode, which represents the maximum number of sub-carriers per OFDM symbol. This maximum mode therefore sets the maximum memory size. For example, in a transmitter/receiver capable of the 32K mode, when operating in a mode with fewer carriers (i.e. 16K, 8K, 4K or 1K) then rather than employing separate odd and even symbol interleaving processes, two odd interleavers would be used.

**[0066]** An illustration of an adaptation of the symbol interleaver 33 which is shown in Figure 3 when interleaving input data symbols onto the sub-carriers of OFDM symbols in the odd interleaving mode only is shown in Figure 10. The symbol interleaver 33.1 corresponds exactly to the symbol interleaver 33 as shown in Figure 3, except that the address generator 102.1 is adapted to perform the odd interleaving process only. For the example shown in Figure 10, the symbol interleaver 33.1 is operating in a mode where the number of data symbols which can be carried per OFDM symbol is less than half of the maximum number which can be carried by an OFDM symbol in an operating mode with the largest number of sub-carriers per OFDM symbol. As such, the symbol interleaver 33.1 has been arranged to partition the interleaver memory 100. For the present illustration shown in Figure 10 the interleaver memory then 100 is divided into two parts 401, 402. As an illustration of the symbol interleaver 33.1 operating in a mode in which data symbols are mapped onto the OFDM symbols using the odd interleaving process, Figure 10 provides an expanded view of each half of the interleaver memory 401, 402. The expanded provides an illustration of the odd interleaving mode as represented for the transmitter side for four symbols A, B, C, D reproduced from Figure 4. Thus as shown in Figure 10, for successive sets of first and second data symbols, the data symbols are written into the interleaver memory 401, 402 in a sequential order and read out in accordance with addresses generated by the address generator 102 in a permuted order in accordance with the addresses generated by the address generator as previously explained. Thus as illustrated in Figure 10, since an odd interleaving process is being performed for successive sets of first and second sets of data symbols, the interleaver memory must be partitioned into two parts. Symbols from a first set of data symbols are written into a first half of the interleaver memory 401, and symbols from a second set of data symbols are written into a second part of the interleaver memory 402, because the symbol interleaver is no longer able to reuse the same parts of the symbol interleaver memory as can be accommodated when operating in an odd and even mode of interleaving.

**[0067]** A corresponding example of the interleaver in the receiver, which appears in Figure 8 but adapted to operate with an odd interleaving process only is shown in Figure 11. As shown in Figure 11 the interleaver memory 540 is divided into two halves 410, 412 and the address generator 542 is adapted to write data symbols into the interleaver memory

and read data symbols from the interleaver memory into respective parts of the memory 410, 402 for successive sets of data symbols to implement an odd interleaving process only. Therefore, in correspondence with representation shown in Figure 10, Figure 11 shows the mapping of the interleaving process which is performed at the receiver and illustrated in Figure 4 as an expanded view operating for both the first and second halves of the interleaving memory 410, 412.

Thus a first set of data symbols are written into a first part of the interleaver memory 410 in a permuted order defined in accordance with the addresses generated by the address generator 542 as illustrated by the order of writing in the data symbols which provides a write sequence of 1, 3, 0, 2. As illustrated the data symbols are then read out of the first part of the interleaver memory 410 in a sequential order thus recovering the original sequence A, B, C, D.

**[0068]** Correspondingly, a second subsequent set of data symbols which are recovered from a successive OFDM symbol are written into the second half of the interleaver memory 412 in accordance with the addresses generated by the address generator 542 in a permuted order and read out into the output data stream in a sequential order.

**[0069]** In one example the addresses generated for a first set of data symbols to write into the first half of the interleaver memory 410 can be reused to write a second subsequent set of data symbols into the interleaver memory 412. Correspondingly, the transmitter may also reuse addresses generated for one half of the interleaver for a first set of data symbols for reading out a second set of data symbols which have been written into the second half of the memory in sequential order.

### **Odd Interleaver with Offset**

**[0070]** The performance of an interleaver, which uses two odd interleavers could be further improved by using a sequence of odd only interleavers rather than a single odd only interleaver, so that any bit of data input to the interleaver does not always modulate the same carrier in the OFDM symbol.

**[0071]** A sequence of odd only interleavers could be realised by either:

- adding an offset to the interleaver address modulo the number of data carriers, or
- using a sequence of permutations in the interleaver

### **Adding an Offset**

**[0072]** Adding an offset to the interleaver address modulo the number of data carriers effectively shifts and wraps-round the OFDM symbol so that any bit of data input to the interleaver does not always modulate the same carrier in the OFDM symbol. Thus the address generator, could optionally include an offset generator, which generates an offset in an address generated by the address generator on the output channel H(q).

**[0073]** The offset would change each symbol. For example, this offset could provide be a cyclic sequence. This cyclic sequence could be, for example, of length 4 and could consist of, for example, prime numbers. For example, such a sequence could be:

0,41,97,157

**[0074]** Furthermore, the offset may be a random sequence, which may be generated by another address generator from a similar OFDM symbol interleaver or may be generated by some other means.

### **Using a Sequence of Permutations**

**[0075]** As shown in Figures 5, a control line 111 extends from the control unit of the address generator to the permutation circuit. As mentioned above, in one example the address generator can apply a different permutation code from a set of permutation codes for successive OFDM symbols. Using a sequence of permutations in the interleaver address generator reduces a likelihood that any bit of data input to the interleaver does not always modulate the same sub-carrier in the OFDM symbol.

**[0076]** For example, this could be a cyclic sequence, so that a different permutation code in a set of permutation codes in a sequence is used for successive OFDM symbols and then repeated. This cyclic sequence could be, for example, of length two or four. For the example of the 16K symbol interleaver a sequence of two permutation codes which are cycled through per OFDM symbol could be for example:

8 4 3 2 0 1 1 1 5 1 2 1 0 6 7 9  
7 9 5 3 1 1 1 4 0 2 1 2 1 0 8 6

whereas a sequence of four permutation codes could be:

## EP 2 056 470 B9

8432011151210679  
7953111402121086  
6117523011081294  
5129031024678111

**[0077]** The switching of one permutation code to another could be effected in response to a change in the Odd/Even signal indicated on the control channel 108. In response the control unit 224 changes the permutation code in the permutation code circuit 210 via the control line 111.

For the example of a 1k symbol interleaver, two permutation codes could be:

432105678  
325014786

whereas four permutation codes could be:

432105678  
325014786  
753826140  
168253407

**[0078]** Other combinations of sequences may be possible for 2k, 4k and 8k carrier modes or indeed 0.5k carrier mode. For example, the following permutation codes for each of the 0.5k, 2k, 4k and 8k provide good de-correlation of symbols and can be used cyclically to generate the offset to the address generated by an address generator for each of the respective modes:

2k Mode:

0751826934\*  
4832901567  
8390215746  
7048369152

4k Mode:

710581249036\*\*  
627108034195  
954231010687  
141039726508

8k Mode:

51130108692417\*  
10854291067311  
11698472101053  
83117915640210

**[0079]** For the permutation codes indicated above, the first two could be used in a two sequence cycle, whereas all four could be used for a four sequence cycle. In addition, some further sequences of four permutation codes, which are cycled through to provide the offset in an address generator to produce a good de-correlation in the interleaved symbols (some are common to the above) are provided below:

0.5k Mode:

37461205

(continued)

4 2 5 7 3 0 1 6  
 5 3 6 0 4 1 2 7  
 6 1 0 5 2 7 4 3

2k Mode:

0 7 5 1 8 2 6 9 3 4 \*  
 3 2 7 0 1 5 8 4 9 6  
 4 8 3 2 9 0 1 5 6 7  
 7 0 3 9 5 2 1 0 6 4 8

4k Mode:

7 1 0 5 8 1 2 4 9 0 3 6 \*\*  
 6 2 7 1 0 8 0 3 4 1 9 5  
 1 0 3 4 1 2 7 0 6 8 5 9  
 0 8 9 5 1 0 4 6 3 2 1 7

8k Mode:

5 1 1 3 0 1 0 8 6 9 2 4 1 7 \*  
 8 1 0 7 6 0 5 2 1 3 9 4 1 1  
 1 1 3 6 9 2 7 4 1 0 5 1 0 8  
 1 0 8 1 7 5 6 0 1 1 4 2 9 3

\*these are the permutations in the DVB-T standard

\*\*these are the permutations in the DVB-H standard

**[0080]** Examples of address generators, and corresponding interleavers, for the 2k, 4k and 8k modes are disclosed in European patent application number 04251667.4. An address generator for the 0.5k mode are disclosed in our co-pending UK patent application number 0722553.5.

**[0081]** Various modifications may be made to the embodiments described above without departing from the scope of the present invention.

**[0082]** As will be appreciated the transmitter and receiver shown in Figures 1 and 7 respectively are provided as illustrations only and are not intended to be limiting. For example, it will be appreciated that the position of the symbol interleaver and the de-interleaver with respect, for example to the bit interleaver and the mapper can be changed. As will be appreciated the effect of the interleaver and de-interleaver is unchanged by its relative position, although the interleaver may be interleaving I/Q symbols instead of v-bit vectors. A corresponding change may be made in the receiver. Accordingly the interleaver and de-interleaver may be operating on different data types, and may be positioned differently to the position described in the example embodiments.

**[0083]** According to an implementation of an example transmitter there is provided a data processing apparatus operable to map input symbols to be communicated onto a predetermined number of sub-carrier signals of an Orthogonal Frequency Division Multiplexed (OFDM) symbol. The data processing apparatus comprises an interleaver operable to read-into a memory the predetermined number of data symbols for mapping onto the OFDM sub-carrier signals, and to read-out of the memory the data symbols for the OFDM sub-carriers to effect the mapping. The read-out is in a different order than the read-in, the order being determined from a set of addresses, with the effect that the data symbols are interleaved on the sub-carrier signals. The set of addresses is determined by an address generator, an address being generated for each of the input symbols to indicate one of the sub-carrier signals onto which the data symbol is to be mapped.

**[0084]** The address generator comprises a linear feedback shift register including a predetermined number of register stages and is operable to generate a pseudo-random bit sequence in accordance with a generator polynomial, and a permutation circuit and a control unit. The permutation circuit is operable to receive the content of the shift register stages and to permute the bits present in the register stages in accordance with a permutation order to form an address of one

of the OFDM sub-carriers.

[0085] The control unit is operable in combination with an address check circuit to regenerate an address when a generated address exceeds a predetermined maximum valid address. The predetermined maximum valid address is approximately sixteen thousand, the linear feedback shift register has thirteen register stages with a generator polynomial

for the linear feedback shift register of  $R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$ , and the permutation order forms, with an additional bit, a fourteen bit address  $R_i[n]$  for the i-th data symbol from the bit present

in the n-th register stage  $R'_i[n]$  in accordance with a code defined by the table:

R <sub>i</sub> ' bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> bit positions	8	4	3	2	0	11	1	5	12	10	6	7	9

[0086] As explained above the permutation codes and generator polynomial of the interleaver, which has been described with reference to an implementation of a particular mode, can equally be applied to other modes, by changing the predetermined maximum allowed address in accordance with the number of sub-carriers for that mode.

[0087] As mentioned above, embodiments of the present invention find application with DVB standards such as DVB-T, DVB-T2 and DVB-H, which are incorporated herein by reference. For example embodiments of the present invention may be used in a transmitter or receiver operating in accordance with the DVB-H standard, in handheld mobile terminals. The mobile terminals may be integrated with mobile telephones (whether second, third or higher generation) or Personal Digital Assistants or Tablet PCs for example. Such mobile terminals may be capable of receiving DVB-H or DVB-T compatible signals inside buildings or on the move in for example cars or trains, even at high speeds. The mobile terminals may be, for example, powered by batteries, mains electricity or low voltage DC supply or powered from a car battery. Services that may be provided by DVB-H may include voice, messaging, internet browsing, radio, still and/or moving video images, television services, interactive services, video or near-video on demand and option. The services might operate in combination with one another. In other examples embodiments of the present invention finds application with the DVB-T2 standard as specified in accordance with ETSI standard EN 302 755. In other examples embodiments of the present invention find application with the cable transmission standard known as DVB-C2. However, it will be appreciated that the present invention is not limited to application with DVB and may be extended to other standards for transmission or reception, both fixed and mobile.

## Claims

1. A data processing apparatus operable to map symbols received from a predetermined number of sub-carrier signals of an Orthogonal Frequency Division Multiplexed (OFDM) symbol into an output symbol stream, the data processing apparatus comprising  
a de-interleaver (314) operable to read-into a memory (540) the predetermined number of data symbols from the OFDM sub-carrier signals, and to read-out of the memory (540) the data symbols into the output symbol stream to effect the mapping, the read-out being in a different order than the read-in, the order being determined from a set of addresses, with the effect that the data symbols are de-interleaved from the OFDM sub-carrier signals,  
an address generator (542) operable to generate the set of addresses, an address being generated for each of the received data symbols to indicate the OFDM sub-carrier signal from which the received data symbol is to be mapped into the output symbol stream, the address generator (542) comprising  
a linear feedback shift register (200) including a predetermined number of register stages and being operable to generate a pseudo-random bit sequence in accordance with a generator polynomial,  
a permutation circuit (210) operable to receive the content of the shift register stages and to permute the bits present in the register stages in accordance with a permutation order to form an address of one of the OFDM sub-carriers, and  
a control unit (224) operable in combination with an address check circuit to re-generate an address when a generated address exceeds a predetermined maximum valid address, **characterised in that**  
the predetermined maximum valid address is approximately sixteen thousand, the linear feedback shift register (200) has thirteen register stages with a generator polynomial for the linear feedback shift register of

$$R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11],$$

and the permutation order forms, with an additional bit, a fourteen bit address  $R_i[n]$  for the i-th data symbol from the bit present in the n-th register stage  $R_i[n]$  in accordance with a code defined by the table:

R <sub>i</sub> ' bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> bit positions	8	4	3	2	0	11	1	5	12	10	6	7	9

2. A data processing apparatus as claimed in Claim 1, wherein the predetermined maximum valid address is a value substantially between twelve thousand and sixteen thousand three hundred and eighty four.
3. A data processing apparatus as claimed in Claim 1, wherein the OFDM symbol includes pilot sub-carriers, which are arranged to carry known symbols, and the predetermined maximum valid address depends on a number of the pilot sub-carrier symbols present in the OFDM symbol.
4. A data processing apparatus as claimed in any of Claims 1 to 3, wherein the de-interleaver memory (540) is arranged to effect the mapping of the received data symbols from the sub-carrier signals onto the output data stream for even OFDM symbols by reading in the data symbols according to a sequential order and reading out the data symbols from memory according to the set of addresses generated by the address generator (542), and for odd OFDM symbols by reading in the symbols into the memory (540) in accordance with the set of addresses generated by the address generator and reading out the data symbols from the memory (540) in accordance with a sequential order.
5. A data processing apparatus as claimed any preceding Claim, wherein the permutation circuit (210) is operable to change the permutation code, which permutes the order of the bits of the register stages to form the addresses from one OFDM symbol to another.
6. A data processing apparatus as claimed in Claim 5, wherein the permutation circuit (210) is operable to cycle through a sequence of different permutation codes for successive OFDM symbols.
7. A data processing apparatus as claimed in Claim 6, wherein the sequence of permutation codes comprises two permutation codes, which are

R <sub>i</sub> ' bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> bit positions	8	4	3	2	0	11	1	5	12	10	6	7	9

and

R <sub>i</sub> ' bit positions	12	11	10	9	8	7	6	11	4	3	2	1	0
R <sub>i</sub> bit positions	7	9	5	3	11	1	4	0	2	12	10	8	6

8. A data processing apparatus as claimed in Claim 5, 6 or 7, wherein for both odd OFDM symbols and even OFDM symbols the interleaver (314) is operable to read-into the memory (540) the predetermined number of data symbols from the OFDM sub-carrier signals in accordance with the addresses generated by the address generator (542), and to read-out from the memory (540) the data symbols into the output symbol stream to effect the mapping in a sequential order.
9. A receiver for receiving data from Orthogonal Frequency Division Multiplexing (OFDM) modulated signal, the receiver including a data processing apparatus according to any of Claims 1 to 8.
10. A receiver as claimed in Claim 9, wherein the receiver is operable to receive data which has been modulated in accordance with a Digital Video Broadcasting standard such as the Digital Video Broadcasting-Terrestrial, Digital Video Broadcasting-Handheld or the Digital Video Broadcasting-Terrestrial2 standard.

11. A method of mapping symbols received from a predetermined number of sub-carrier signals of an Orthogonal Frequency Division Multiplexed (OFDM) symbol into an output symbol stream, the method comprising reading-into a memory (540) the predetermined number of data symbols from the OFDM sub-carrier signals, reading-out of the memory (540) the data symbols into the output symbol stream to effect the mapping, the read-out being in a different order than the read-in, the order being determined from a set of addresses, with the effect that the data symbols are de-interleaved from the OFDM sub-carrier signals, generating the set of addresses, an address being generated for each of the received symbols to indicate the OFDM sub-carrier signal from which the received data symbol is to be mapped into the output symbol stream, the generating the set of addresses comprising using a linear feedback shift register (200) including a predetermined number of register stages to generate a pseudo-random bit sequence in accordance with a generator polynomial, using a permutation circuit (210) to receive the content of the shift register stages and to permute the bits present in the register stages in accordance with a permutation order to form an address, and re-generating an address when a generated address exceeds a predetermined maximum valid address, **characterised in that** the predetermined maximum valid address is approximately sixteen thousand, the linear feedback shift register (200) has thirteen register stages with a generator polynomial for the linear feedback shift register of  $R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$ , and the permutation order forms, with an additional bit, a fourteen bit address  $R_i[n]$  for the i-th data symbol from the bit present in the n-th register stage

$R'_i[n]$  in accordance with a code defined by the table:

R' <sub>i</sub> bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> bit positions	8	4	3	2	0	11	1	5	12	10	6	7	9

12. A method as claimed in Claim 11, wherein the predetermined maximum valid address is a value substantially between twelve thousand and sixteen thousand three hundred and eighty four.
13. A method as claimed in Claim 11, wherein the OFDM symbol includes pilot sub-carriers, which are arranged to carry known symbols, and the predetermined maximum valid address depends on a number of the pilot sub-carrier symbols present in the OFDM symbol.
14. A method as claimed any of Claim 11 to 13, wherein the using a permutation circuit (210) to receive the content of the shift register stages and permuting the bits present in the register stages in accordance with the permutation code to form an address, includes changing the permutation code, which permutes the order of the bits of the register stages to form the addresses, from one OFDM symbol to another.
15. A method as claimed in Claim 14, wherein the changing the permutation code, which permutes the order of the bits of the register stages to form the addresses, from one OFDM symbol to another includes cycling through a sequence of different permutation codes for successive OFDM symbols.
16. A method as claimed in Claim 15, wherein the sequence of permutation codes comprises two permutation codes, which are

R' <sub>i</sub> bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> bit positions	8	4	3	2	0	11	1	5	12	10	6	7	9

and

R' <sub>i</sub> bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
-------------------------------	----	----	----	---	---	---	---	---	---	---	---	---	---



(continued)

R <sub>i</sub> bit positions	7	9	5	3	11	1	4	0	2	12	10	8	6
------------------------------	---	---	---	---	----	---	---	---	---	----	----	---	---

17. A method as claimed in Claim 14, 15 or 16, wherein the reading-into the memory (540) the predetermined number of data symbols from the OFDM sub-carrier signals, comprises for both odd OFDM symbols and even OFDM symbols reading in the data symbols into memory in accordance with the addresses generated by the address generator, and the reading-out of the memory (540) the data symbols into the output symbol stream to effect the mapping, comprises for both odd OFDM symbols and even OFDM symbols reading-out the data symbols from the memory (540) in a sequential order.

18. A method of receiving data from Orthogonal Frequency Division Multiplexing OFDM modulated symbols, the method including receiving a predetermined number of data symbols from a predetermined number of sub-carrier signals from the OFDM symbols for forming an output data stream, and mapping the data symbols received from the sub-carriers of the OFDM symbols into the output data stream according to the method claimed in any of claims 11 to 17.

19. An address generator (542) for use with reception of data symbols interleaved onto sub-carriers of an Orthogonal Frequency Division Multiplexed symbol, the address generator being operable to generate a set of addresses, each address being generated for each of the data symbols to indicate one of the sub-carrier signals onto which the data symbol is to be mapped, the address generator (542) comprising a linear feedback shift register (200) including a predetermined number of register stages and being operable to generate a pseudo-random bit sequence in accordance with a generator polynomial, a permutation circuit (210) operable to receive the content of the shift register stages and to permute the bits present in the register stages in accordance with a permutation order to form an address, and a control unit (224) operable in combination with an address check circuit to re-generate an address when a generated address exceeds a predetermined maximum valid address, **characterised in that** the predetermined maximum valid address is approximately sixteen thousand, the linear feedback shift register (200) has thirteen register stages with a generator polynomial for the linear feedback shift register of

$R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$ , and the permutation order forms, with

an additional bit, a fourteen bit address  $R_i[n]$  for the i-th data symbol from the bit present in the n-th register stage

$R'_i[n]$  in accordance with the table:

R' <sub>i</sub> bit positions	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> bit positions	8	4	3	2	0	11	1	5	12	10	6	7	9

## Patentansprüche

1. Datenverarbeitungsvorrichtung, welche betriebsfähig ist, Symbole, welche von einer vorher bestimmten Anzahl von Unterträgersignalen eines orthogonalen Frequenzmultiplexsymbols (OFDM-Symbol) empfangen werden, auf einen Ausgangssymbolstrom abzubilden, wobei die Datenverarbeitungsvorrichtung umfasst:

ein Entschachtelungsorgan (314), welches betriebsfähig ist, in einen Speicher (540) die vorher bestimmte Anzahl von Datensymbolen von OFDM-Unterträgersignalen zu lesen, und aus dem Speicher die Datensymbole in den Ausgangssymbolstrom zu lesen, um das Abbilden zu bewirken, wobei das Auslesen in einer anderen Reihenfolge als das Einlesen erfolgt, wobei die Reihenfolge von einem Satz an Adressen bestimmt wird, mit der Wirkung, dass die Datensymbole von den OFDM-Unterträgersignalen entschachtelt sind, einen Adressgenerator (542), der betriebsfähig ist, den Satz an Adressen zu erzeugen, wobei eine Adresse für jedes der empfangenen Datensymbole erzeugt wird, um das OFDM-Unterträgersignal zu zeigen, von wel-

chem das empfangene Datensymbol auf einen Ausgangssymbolstrom abzubilden ist, wobei der Adressgenerator (542) umfasst:

ein lineares Rückführschieberegister (200), welches eine vorher bestimmte Anzahl an Registerstufen aufweist und betriebsfähig ist, eine Pseudozufalls-Bitsequenz gemäß einem Generator-Polynom zu erzeugen, eine Permutationsschaltung (210), die betriebsfähig ist, den Inhalt der Schieberegisterstufen zu empfangen und die Bits, welche in den Registerstufen vorhanden sind, gemäß einer Permutationsreihenfolge zu permutieren, um eine Adresse von einem der OFDM-Hilfsträger zu bilden, und  
eine Steuereinheit (224), welche in Kombination mit einer Adressenprüfschaltung betriebsfähig ist, eine Adresse wieder zu erzeugen, wenn eine erzeugte Adresse eine vorher bestimmte maximale gültige Adresse übersteigt, **dadurch gekennzeichnet, dass**  
die vorher bestimmte maximale gültige Adresse ungefähr sechszehntausend ist,  
das linear rückgekoppelte Schieberegister (200) dreizehn Registerstufen mit einem Generator-Polynom für das lineare Rückführschieberegister von  $R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$  hat, und die Permutationsreihenfolge mit einem zusätzlichen Bit eine Vierzehn-Bit-Adresse  $R_i[n]$  für das i-te Datensymbol von dem Bit bildet, welches in der n-ten Registerstufe  $R'_i$  vorhanden ist, gemäß einem Code, der durch die Tabelle definiert ist:

$R'_i$ - Bitpositionen	12	11	10	9	8	7	6	5	4	3	2	1	0
$R_i$ - Bitpositionen	8	4	3	2	0	11	1	5	12	10	6	7	9

2. Datenverarbeitungsvorrichtung nach Anspruch 1, wobei die vorher bestimmte maximale gültige Adresse ein Wert ist, der im Wesentlichen zwischen zwölftausend und sechszehntausenddreihundertvierundachtzig ist.
3. Datenverarbeitungsvorrichtung nach Anspruch 1, wobei das OFDM-Symbol Pilot-Unterträger aufweist, die eingerichtet sind, bekannte Symbole zu tragen, und die vorher bestimmte maximale gültige Adresse in Abhängigkeit von einer Anzahl der Pilot-Unterträgersymbole, welche im OFDM-Symbol vorhanden sind, abhängt.
4. Datenverarbeitungsvorrichtung nach Anspruch 1 bis 3, wobei der Entschachtelungsspeicher (540) eingerichtet ist, das Abbilden der empfangenen Datensymbole von den Unterträgersignalen auf den Ausgangsdatenstrom für geradzahlige OFDM-Symbole zu bewirken, indem die Datensymbole gemäß einer sequentiellen Reihenfolge eingelesen werden und die Datensymbole vom Speicher gemäß dem Satz an Adressen, welche durch den Adressgenerator (542) erzeugt werden, gelesen werden, und für ungeradzahlige OFDM-Symbole, indem die Symbole in den Speicher (100) gemäß einem Satz von Adressen, welche durch den Adressgenerator erzeugt werden, eingelesen werden und die Datensymbole vom Speicher gemäß einer sequentiellen Reihenfolge ausgelesen werden.
5. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, wobei die Permutationsschaltung (210) betriebsfähig ist, den Permutationscode zu ändern, der die Reihenfolge der Bits der Registerstufen permutiert, um die Adressen von einem OFDM-Symbol zum anderen zu bilden.
6. Datenverarbeitungsvorrichtung nach Anspruch 5, wobei die Permutationsschaltung (210) betriebsfähig ist, eine Folge verschiedener Permutationscodes für aufeinanderfolgende OFDM-Symbole periodisch zu durchlaufen.
7. Datenverarbeitungsvorrichtung nach Anspruch 6, wobei die Folge der Permutationscodes zwei Permutationscodes umfasst, welche sind:

$R'_i$ - Bitpositionen	12	11	10	9	8	7	6	5	4	3	2	1	0
$R_i$ - Bitpositionen	8	4	3	2	0	11	1	5	12	10	6	7	9

und

$R'_i$ - Bitpositionen	12	11	10	9	8	7	6	5	4	3	2	1	0
------------------------	----	----	----	---	---	---	---	---	---	---	---	---	---

(fortgesetzt)

R <sub>i</sub> - Bitpositionen	7	9	5	3	11	1	4	0	2	12	10	8	6
--------------------------------	---	---	---	---	----	---	---	---	---	----	----	---	---

8. Datenverarbeitungsvorrichtung nach Anspruch 5, 6 oder 7, wobei für sowohl ungeradzahlige OFDM-Symbole als auch für geradzahlige OFDM-Symbole das Entschaltungsorgan (314) betriebsfähig ist, in den Speicher (540) die vorher bestimmte Anzahl von Datensymbolen von den OFDM-Unterträgersignalen gemäß den Adressen einzulesen, welche durch den Adressgenerator (542) erzeugt werden, und aus dem Speicher (540) die Datensymbole in den Ausgangssymbolstrom auszulesen, um das Abbilden in einer sequentiellen Reihenfolge zu bewirken.
9. Empfänger zum Empfangen von Daten von einem orthogonalen Frequenzmultiplex(OFDM)-Modulationssignal, wobei der Empfänger eine Datenverarbeitungsvorrichtung nach einem der Ansprüche 1 bis 8 aufweist.
10. Empfänger nach Anspruch 9, wobei der Empfänger betriebsfähig ist, Daten, welche gemäß einem digitalen Videorundfunkstandard moduliert wurden, beispielsweise dem terrestrischen digitalen Videorundfunk, dem Standard für digitalen Videorundfunk für Handgeräte (DVB-H) oder dem terrestrischen digitalen Videorundfunkstandard 2, zu empfangen.
11. Verfahren zum Abbilden von Symbolen, welche von einer vorher bestimmten Anzahl an Unterträgersignalen eines orthogonalen Frequenzmultiplexsymbols (OFDM) empfangen werden, auf einen Ausgangssymbolstrom, wobei das Verfahren umfasst:

Einlesen - in einen Speicher (540) - der vorher bestimmten Anzahl von Datensymbolen von den OFDM-Unterträgersignalen,

Auslesen - aus dem Speicher (540) - der Datensymbole in den Ausgangssymbolstrom, um das Abbilden zu bewirken, wobei das Auslesen in einer anderen Reihenfolge als das Einlesen ist, wobei die Reihenfolge von einem Satz an Adressen bestimmt wird, mit der Wirkung, dass die Datensymbole von den OFDM-Unterträgersignalen entschachtelt werden,

Erzeugen des Satzes an Adressen, wobei eine Adresse für jedes der empfangenen Symbole erzeugt wird, um das OFDM-Unterträgersignal anzuzeigen, von dem das empfangene Datensymbol auf den Ausgangssymbolstrom abzubilden ist, wobei das Erzeugen des Satzes von Adressen umfasst:

Verwenden eines linear rückgekoppelten Schieberegisters (200), welches eine vorher bestimmte Anzahl an Registerstufen aufweist, um eine Pseudozufalls-Bitsequenz zu erzeugen, gemäß einem Generator-Polynom,

Verwenden einer Permutationsschaltung (210), welche betriebsfähig ist, den Inhalt der Schieberegisterstufen zu empfangen, um die Bits, die in den Registerstufen vorhanden sind, gemäß einer vorher bestimmten Reihenfolge zu permutieren, um eine Adresse zu bilden, und

Wieder-Erzeugen einer Adresse, wenn eine erzeugte Adresse eine vorher bestimmte maximale gültige Adresse übersteigt, **dadurch gekennzeichnet, dass**

die vorher bestimmte maximale gültige Adresse ungefähr sechszehntausend ist, das linear rückgekoppelte Schieberegister (200) dreizehn Registerstufen mit einem Generator-Polynom für das lineare Rückführschieberegister von  $R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$  hat, und die Permutationsreihenfolge mit einem zusätzlichen Bit eine Vierzehn-Bit-Adresse  $R_i[n]$  für das  $i$ -te Datensymbol von dem Bit bildet, welches in der  $n$ -ten Registerstufe  $R'_i$  vorhanden ist, gemäß der Tabelle:

R' <sub>i</sub> - Bitpositionen	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> - Bitpositionen	8	4	3	2	0	11	1	5	12	10	6	7	9

12. Verfahren nach Anspruch 11, wobei die vorher bestimmte maximale gültige Adresse ein Wert ist, der im Wesentlichen zwischen zwölftausend und sechszehntausenddreihundertvierundachtzig ist.
13. Verfahren nach Anspruch 11, wobei das OFDM-Symbol Pilot-Unterträger aufweist, welche eingerichtet sind, bekannte Symbole zu tragen, und die vorher bestimmte maximale gültige Adresse von einer Anzahl der Pilot-UnterträgerSymbole abhängt, welche im OFDM-Symbol vorhanden sind.

14. Verfahren nach einem der Ansprüche 11 bis 13, wobei das Verwenden einer Permutationsschaltung (210), um den Inhalt der Schieberegisterstufen zu empfangen und das Permutieren der Bits, welche in den Registerstufen vorhanden sind, gemäß dem Permutationscode, um eine Adresse zu bilden, das Ändern der Permutationscodes umfasst, der die Reihenfolge der Bits der Registerstufen, um die Adressen zu bilden, von einem OFDM-Symbol zum anderen permutiert.

15. Verfahren nach Anspruch 14, wobei das Ändern des Permutationscodes, der die Reihenfolge der Bits der Registerstufen permutiert, um die Adressen zu bilden, von einem OFDM-Symbol zum anderen das periodische Durchlaufen einer Sequenz verschiedener Permutationscodes für aufeinanderfolgende OFDM-Symbole umfasst.

16. Verfahren nach Anspruch 15, wobei die Sequenz der Permutationscodes zwei Permutationscodes umfasst, welche sind:

R' <sub>i</sub> - Bitpositionen	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> - Bitpositionen	8	4	3	2	0	11	1	5	12	10	6	7	9

und

R' <sub>i</sub> - Bitpositionen	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> - Bitpositionen	7	9	5	3	11	1	4	0	2	12	10	8	6

17. Verfahren nach Anspruch 14, 15 oder 16, wobei das Einlesen in den Speicher (540) der vorher bestimmten Anzahl von Datensymbolen von den OFDM-Unterträgersignalen für sowohl ungeradzahlige OFDM-Symbole als auch für geradzahlige OFDM-Symbole das Einlesen der Daten in den Speicher gemäß den Adressen, welche durch den Adressgenerator erzeugt werden, umfasst, und das Auslesen aus dem Speicher (540) der Datensymbole in den Ausgangssymbolstrom, um das Abbilden zu bewirken, für sowohl ungeradzahlige OFDM-Symbole als auch für geradzahlige OFDM-Symbole das Auslesen der Datensymbole aus dem Speicher (540) in einer sequentiellen Reihenfolge umfasst.

18. Verfahren zu Empfangen von Daten von orthogonalen Frequenzmultiplex-OFDM-Modulationssymbolen, wobei das Verfahren umfasst Empfangen einer vorher bestimmten Anzahl von Datensymbolen von einer vorher bestimmten Anzahl von Unterträgersignalen von den OFDM-Symbolen zum Bilden eines Ausgangsdatenstroms, und Abbilden der Datensymbole, welche von den Unterträgern der OFDM-Symbole empfangen werden, in den Ausgangsdatenstrom gemäß dem Verfahren nach einem der Ansprüche 11 bis 17.

19. Adressgenerator (542) zur Verwendung beim Empfang von Datensymbolen, welche auf Unterträgern eines orthogonalen Frequenzmultiplexsymbols verschachtelt sind, wobei der Adressgenerator betriebsfähig ist, einen Satz von Adressen zu erzeugen, wobei jede Adresse für jedes der Datensymbole erzeugt wird, um eines von den Unterträgersignalen anzuzeigen, auf welches das Datensymbol abzubilden ist, wobei der Adressgenerator (542) umfasst:

ein linear rückgekoppeltes Schieberegister (200), welches eine vorher bestimmte Anzahl an Registerstufen aufweist und betriebsfähig ist, eine Pseudozufalls-Bitsequenz gemäß einem Generator-Polynom zu erzeugen, eine Permutationsschaltung (210), die betriebsfähig ist, den Inhalt der Schieberegisterstufen zu empfangen und die Bits, welche in den Registerstufen vorhanden sind, gemäß einer Permutationsreihenfolge zu permutieren, um eine Adresse von einem der OFDM-Unterträger zu bilden, und eine Steuereinheit (224), welche in Kombination mit einer Adressenprüfschaltung betriebsfähig ist, eine Adresse wieder zu erzeugen, wenn eine erzeugte Adresse eine vorher bestimmte maximale gültige Adresse übersteigt, **dadurch gekennzeichnet, dass** die vorher bestimmte maximale gültige Adresse ungefähr sechszehntausend ist, das linear rückgekoppelte Schieberegister (200) dreizehn Registerstufen mit einem Generator-Polynom für das lineare Rückführschieberegister von  $R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$  hat, und die Permutationsreihenfolge mit einem zusätzlichen Bit eine Vierzehn-Bit-Adresse  $R_i[n]$  für das i-te Datensymbol von dem Bit bildet, welches in der n-ten Registerstufe

$R'_i$  vorhanden ist, gemäß einem Code, der durch die Tabelle definiert ist:

$R'_i$ - Bitpositionen	12	11	10	9	8	7	6	5	4	3	2	1	0
$R_i$ - Bitpositionen	8	4	3	2	0	11	1	5	12	10	6	7	9

## Revendications

- Appareil de traitement de données utilisable pour mapper des symboles reçus à partir d'un nombre prédéterminé de signaux de sous-porteuse d'un symbole multiplexé par division de fréquence orthogonale (OFDM) en un flux de symboles de sortie, l'appareil de traitement de données comprenant un désentrelaceur (314) utilisable pour entrer en lecture dans une mémoire (540) le nombre prédéterminé de symboles de données à partir des signaux de sous-porteuse OFDM, et pour sortir en lecture de la mémoire (540) les symboles de données en le flux de symboles de sortie pour effectuer le mappage, la sortie en lecture étant dans un ordre différent de l'entrée en lecture, l'ordre étant déterminé à partir d'un ensemble d'adresses, avec l'effet que les symboles de données sont désentrelacés des signaux de sous-porteuse OFDM, un générateur d'adresse (542) utilisable pour générer l'ensemble d'adresses, une adresse étant générée pour chacun des symboles de données reçus pour indiquer le signal de sous-porteuse OFDM à partir duquel le symbole de données reçu est à mapper en le flux de symboles de sortie, le générateur d'adresse (542) comprenant un registre à décalage de contre-réaction linéaire (200) comprenant un nombre prédéterminé d'étages de registre et étant utilisable pour générer une séquence binaire pseudo-aléatoire selon un polynôme de générateur, un circuit de permutation (210) utilisable pour recevoir le contenu des étages de registre à décalage et pour permuter les bits présents dans les étages de registre selon un ordre de permutation pour former une adresse d'une des sous-porteuses OFDM, et une unité de commande (224) utilisable en combinaison avec un circuit de contrôle d'adresse pour régénérer une adresse lorsqu'une adresse générée dépasse une adresse valide maximum prédéterminée, **caractérisé en ce que** l'adresse valide maximum prédéterminée est approximativement seize mille, le registre à décalage de contre-réaction linéaire (200) a treize étages de registre avec un polynôme de générateur pour le registre à décalage de contre-réaction linéaire de  $R'_i[12]=R'_{i-1}[0]\oplus R'_{i-1}[1]\oplus R'_{i-1}[4]\oplus R'_{i-1}[5]\oplus R'_{i-1}[9]\oplus R'_{i-1}[11]$ , et l'ordre de permutation forme, avec un bit additionnel, une adresse de quatorze bits  $R_i[n]$  pour le  $i^{\text{ème}}$  symbole de données à partir du bit présent dans le  $n^{\text{ème}}$  étage de registre  $R'_i[n]$  selon un code défini par la table :

$R'_i$ positions de bit	12	11	10	9	8	7	6	5	4	3	2	1	0
$R_i$ positions de bit	8	4	3	2	0	11	1	5	12	10	6	7	9

- Appareil de traitement de données selon la revendication 1, dans lequel l'adresse valide maximum prédéterminée est une valeur sensiblement entre douze mille et seize mille trois cent quatre-vingt-quatre.
- Appareil de traitement de données selon la revendication 1, dans lequel le symbole OFDM comprend des sous-porteuses pilotes, qui sont agencées pour acheminer des symboles connus, et l'adresse valide maximum prédéterminée dépend d'un nombre des symboles de sous-porteuse pilote présents dans le symbole OFDM.
- Appareil de traitement de données selon l'une quelconque des revendications 1 à 3, dans lequel la mémoire de désentrelaceur (540) est agencée pour effectuer le mappage des symboles de données reçus à partir des signaux de sous-porteuse sur le flux de données de sortie pour des symboles OFDM pairs par l'entrée en lecture des symboles de données selon un ordre séquentiel et la sortie en lecture des symboles de données à partir de la mémoire selon l'ensemble d'adresses généré par le générateur d'adresse (542), et pour des symboles OFDM impairs par l'entrée en lecture des symboles dans la mémoire (540) selon l'ensemble d'adresses généré par le générateur d'adresse et la sortie en lecture des symboles de données à partir de la mémoire (540) selon un ordre séquentiel.
- Appareil de traitement de données selon l'une quelconque des revendications précédentes, dans lequel le circuit de permutation (210) est utilisable pour changer le code de permutation, qui permute l'ordre des bits des étages

de registre pour former les adresses d'un symbole OFDM à un autre.

6. Appareil de traitement de données selon la revendication 5, dans lequel le circuit de permutation (210) est utilisable pour passer par le cycle complet d'une séquence de codes de permutation différents pour des symboles OFDM successifs.

7. Appareil de traitement de données selon la revendication 6, dans lequel la séquence de codes de permutation comprend deux codes de permutation, qui sont

R' <sub>i</sub> positions de bit	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> positions de bit	8	4	3	2	0	11	1	5	12	10	6	7	9
R' <sub>i</sub> positions de bit	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> positions de bit	7	9	5	3	11	1	4	0	2	12	10	8	6

8. Appareil de traitement de données selon la revendication 5, 6 ou 7, dans lequel pour à la fois les symboles OFDM impairs et les symboles OFDM pairs, l'entrelaceur (314) est utilisable pour entrer en lecture dans la mémoire (540) le nombre prédéterminé de symboles de données à partir des signaux de sous-porteuse OFDM selon les adresses générées par le générateur d'adresse (542), et pour sortir en lecture à partir de la mémoire (540) les symboles de données en le flux de symboles de sortie pour effectuer le mappage dans un ordre séquentiel.

9. Récepteur pour recevoir des données à partir d'un signal modulé de multiplexage par division de fréquence orthogonale (OFDM), le récepteur comprenant un appareil de traitement de données selon l'une quelconque des revendications 1 à 8.

10. Récepteur selon la revendication 9, dans lequel le récepteur est utilisable pour recevoir des données qui ont été modulées selon une norme de diffusion vidéo numérique comme la norme de diffusion vidéo numérique terrestre, la norme de diffusion vidéo numérique portable ou la norme de diffusion vidéo numérique terrestre 2.

11. Procédé de mappage de symboles reçus à partir d'un nombre prédéterminé de signaux de sous-porteuse d'un symbole multiplexé par division de fréquence orthogonale (OFDM) en un flux de symboles de sortie, le procédé comprenant

l'entrée en lecture dans une mémoire (540) du nombre prédéterminé de symboles de données à partir des signaux de sous-porteuse OFDM,

la sortie en lecture de la mémoire (540) des symboles de données en le flux de symboles de sortie pour effectuer le mappage, la sortie en lecture étant dans un ordre différent de l'entrée en lecture, l'ordre étant déterminé à partir d'un ensemble d'adresses, avec l'effet que les symboles de données sont désentrelacés des signaux de sous-porteuse OFDM,

la génération de l'ensemble d'adresses, une adresse étant générée pour chacun des symboles reçus pour indiquer le signal de sous-porteuse OFDM à partir duquel le symbole de données reçu est à mapper en le flux de symboles de sortie, la génération de l'ensemble d'adresses comprenant

l'utilisation d'un registre à décalage de contre-réaction linéaire (200) comprenant un nombre prédéterminé d'étages de registre pour générer une séquence binaire pseudo-aléatoire selon un polynôme de générateur,

l'utilisation d'un circuit de permutation (210) pour recevoir le contenu des étages de registre à décalage et pour permuter les bits présents dans les étages de registre selon un ordre de permutation pour former une adresse, et la régénération d'une adresse lorsqu'une adresse générée dépasse une adresse valide maximum prédéterminée,

**caractérisé en ce que**

l'adresse valide maximum prédéterminée est approximativement seize mille,

le registre à décalage de contre-réaction linéaire (200) a treize étages de registre avec un polynôme de générateur pour le registre à décalage de contre-réaction linéaire de  $R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$ , et l'ordre de permutation forme, avec un bit additionnel, une adresse de quatorze bits  $R_i[n]$  pour le  $i^{\text{ème}}$  symbole de données à partir du bit présent dans le  $n^{\text{ème}}$  étage de registre  $R'_i[n]$  selon un code défini par la table :

R' <sub>i</sub> positions de bit	12	11	10	9	8	7	6	5	4	3	2	1	0
----------------------------------	----	----	----	---	---	---	---	---	---	---	---	---	---

(suite)

R <sub>i</sub> positions de bit	8	4	3	2	0	11	1	5	12	10	6	7	9
---------------------------------	---	---	---	---	---	----	---	---	----	----	---	---	---

5 12. Procédé selon la revendication 11, dans lequel l'adresse valide maximum prédéterminée est une valeur sensiblement entre douze mille et seize mille trois cent quatre-vingt-quatre.

10 13. Procédé selon la revendication 11, dans lequel le symbole OFDM comprend des sous-porteuses pilotes, qui sont agencées pour acheminer des symboles connus, et l'adresse valide maximum prédéterminée dépend d'un nombre des symboles de sous-porteuse pilote présents dans le symbole OFDM.

15 14. Procédé selon l'une quelconque des revendications 11 à 13, dans lequel l'utilisation d'un circuit de permutation (210) pour recevoir le contenu des étages de registre à décalage et permuter les bits présents dans les étages de registre selon le code de permutation pour former une adresse, comprend le changement du code de permutation, qui permute l'ordre des bits des étages de registre pour former les adresses, d'un symbole OFDM à un autre.

20 15. Procédé selon la revendication 14, dans lequel le changement du code de permutation, qui permute l'ordre des bits des étages de registre pour former les adresses, d'un symbole OFDM à un autre comprend le passage par le cycle complet d'une séquence de codes de permutation différents pour des symboles OFDM successifs.

25 16. Procédé selon la revendication 15, dans lequel la séquence de codes de permutation comprend deux codes de permutation, qui sont

R' <sub>i</sub> positions de bit	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> positions de bit	8	4	3	2	0	11	1	5	12	10	6	7	9

et

R' <sub>i</sub> positions de bit	12	11	10	9	8	7	6	5	4	3	2	1	0
R <sub>i</sub> positions de bit	7	9	5	3	11	1	4	0	2	12	10	8	6

35 17. Procédé selon la revendication 14, 15 ou 16, dans lequel l'entrée en lecture dans la mémoire (540) du nombre prédéterminé de symboles de données à partir des signaux de sous-porteuse OFDM, comprend pour à la fois les symboles OFDM impairs et les symboles OFDM pairs, l'entrée en lecture des symboles de données dans la mémoire selon les adresses générées par le générateur d'adresse, et  
40 la sortie en lecture de la mémoire (540) des symboles de données en le flux de symboles de sortie pour effectuer le mappage, comprend pour à la fois les symboles OFDM impairs et les symboles OFDM pairs la sortie en lecture des symboles de données à partir de la mémoire (540) dans un ordre séquentiel.

45 18. Procédé de réception de données à partir de symboles modulés de multiplexage par division de fréquence orthogonale (OFDM), le procédé comprenant la réception d'un nombre prédéterminé de symboles de données à partir d'un nombre prédéterminé de signaux de sous-porteuse des symboles OFDM pour former un flux de données de sortie, et le mappage des symboles de données reçus à partir des sous-porteuses des symboles OFDM en le flux de données de sortie selon le procédé selon l'une quelconque des revendications 11 à 17.

50 19. Générateur d'adresse (542) pour l'utilisation avec la réception de symboles de données entrelacés sur des sous-porteuses d'un symbole multiplexé par division de fréquence orthogonale, le générateur d'adresse étant utilisable pour générer un ensemble d'adresses, chaque adresse étant générée pour chacun des symboles de données pour indiquer un des signaux de sous-porteuse sur lequel le symbole de données est à mapper, le générateur d'adresse (542) comprenant  
55 un registre à décalage de contre-réaction linéaire (200) comprenant un nombre prédéterminé d'étages de registre et étant utilisable pour générer une séquence binaire pseudo-aléatoire selon un polynôme de générateur, un circuit de permutation (210) utilisable pour recevoir le contenu des étages de registre à décalage et pour permuter

les bits présents dans les étages de registre selon un ordre de permutation pour former une adresse, et une unité de commande (224) utilisable en combinaison avec un circuit de contrôle d'adresse pour régénérer une adresse lorsqu'une adresse générée dépasse une adresse valide maximum prédéterminée, **caractérisé en ce que** l'adresse valide maximum prédéterminée est approximativement seize mille,

le registre à décalage de contre-réaction linéaire (200) a treize étages de registre avec un polynôme de générateur pour le registre à décalage de contre-réaction linéaire de  $R'_i[12] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[5] \oplus R'_{i-1}[9] \oplus R'_{i-1}[11]$ , et l'ordre de permutation forme, avec un bit additionnel, une adresse de quatorze bits  $R_i[n]$  pour le  $i^{\text{ème}}$  symbole de données à partir du bit présent dans le  $n^{\text{ème}}$  étage de registre  $R'_i[n]$  selon la table :

$R'_i$ positions de bit	12	11	10	9	8	7	6	5	4	3	2	1	0
$R_i$ positions de bit	8	4	3	2	0	11	1	5	12	10	6	7	9



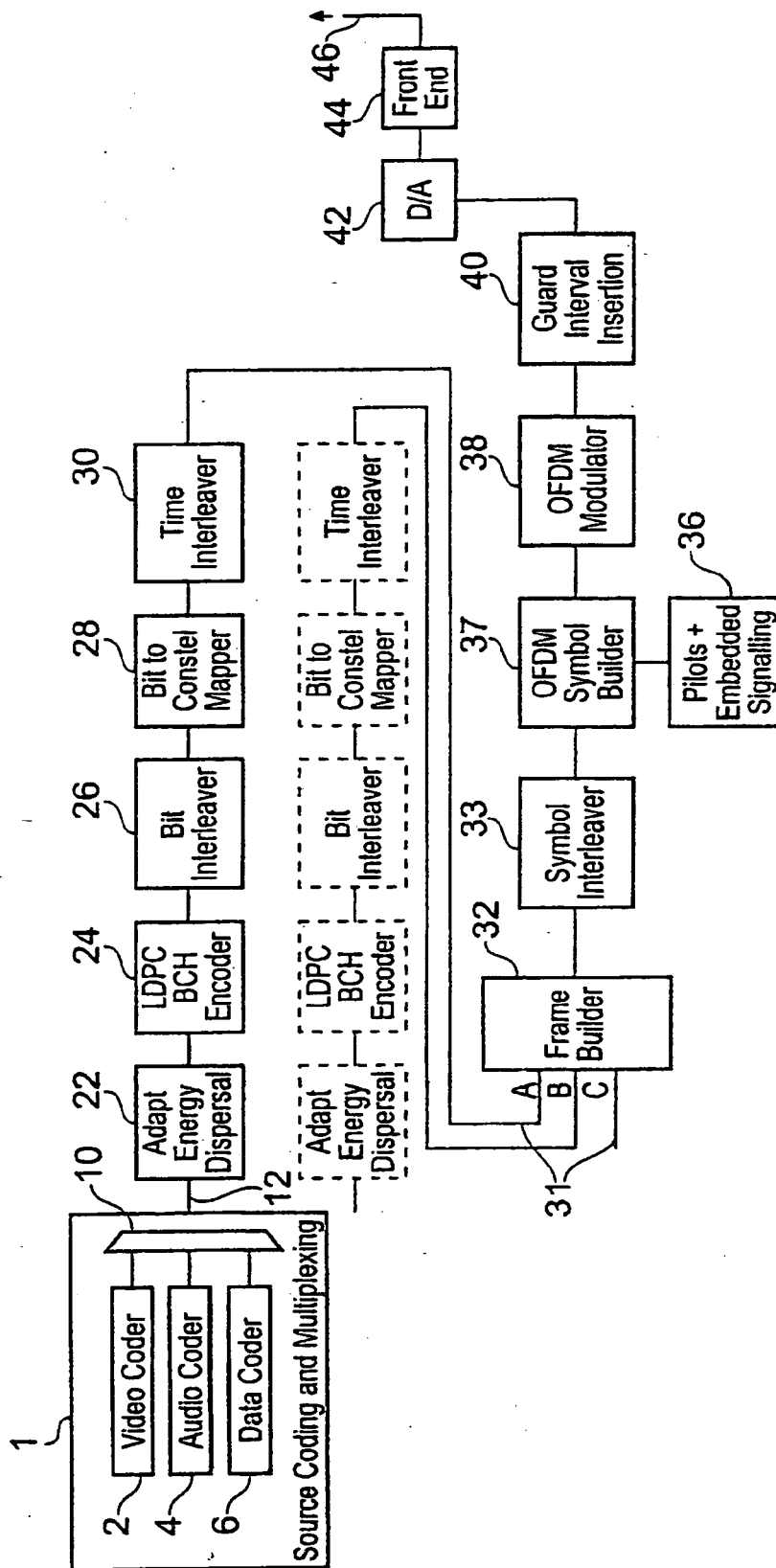


FIG. 1

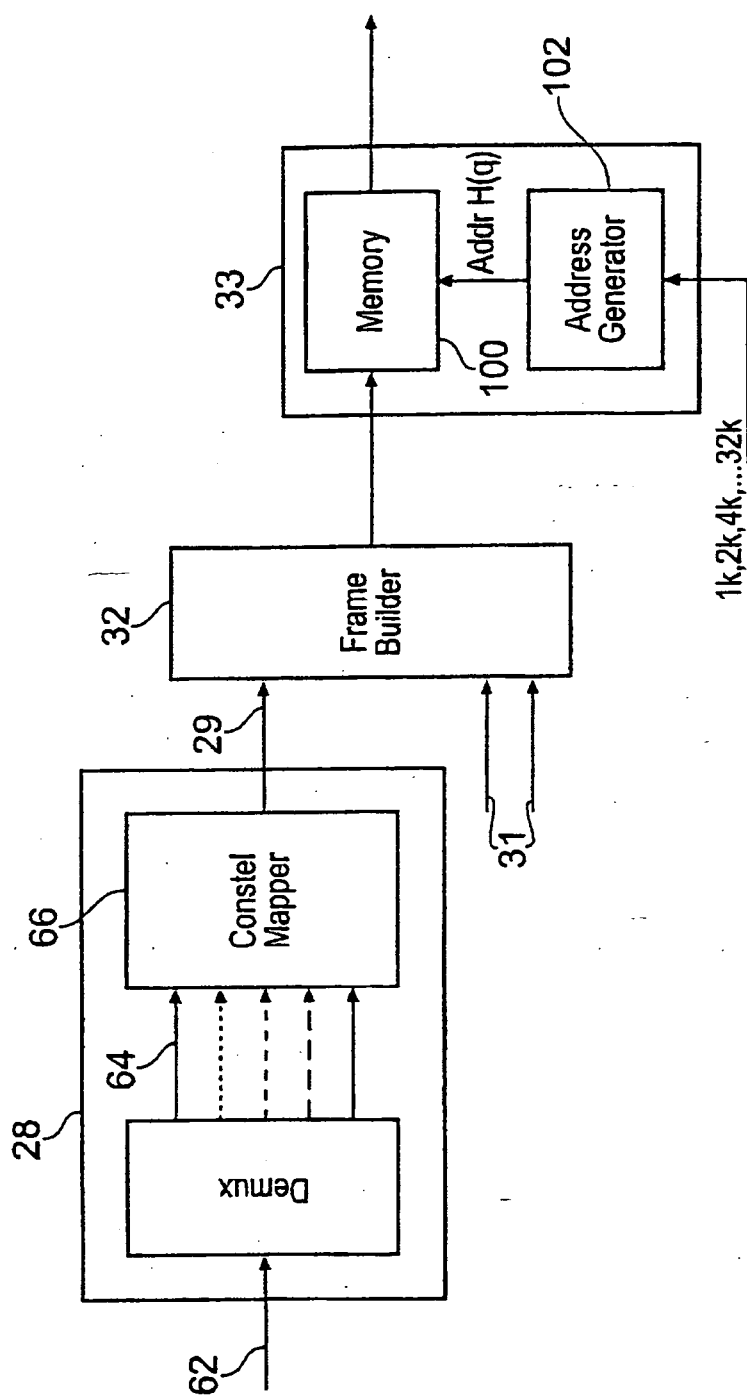


FIG. 2

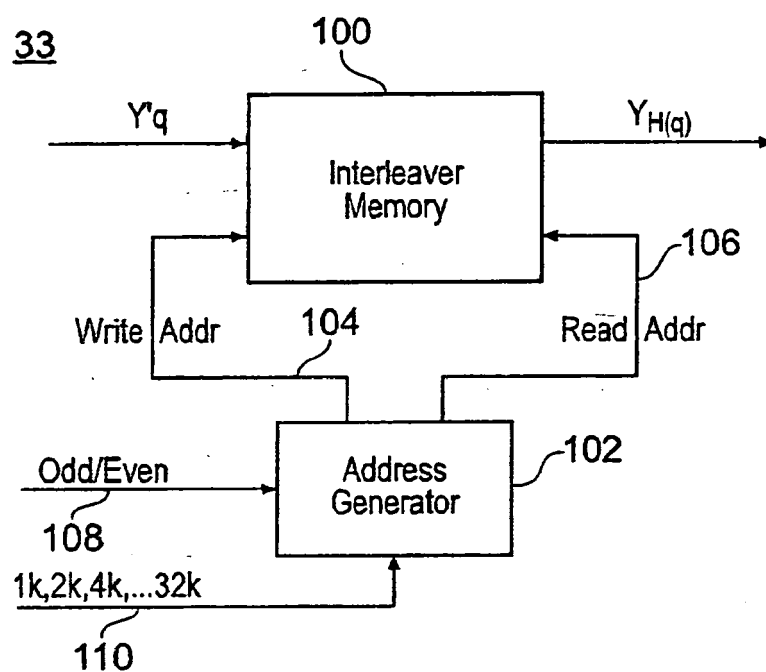


FIG. 3

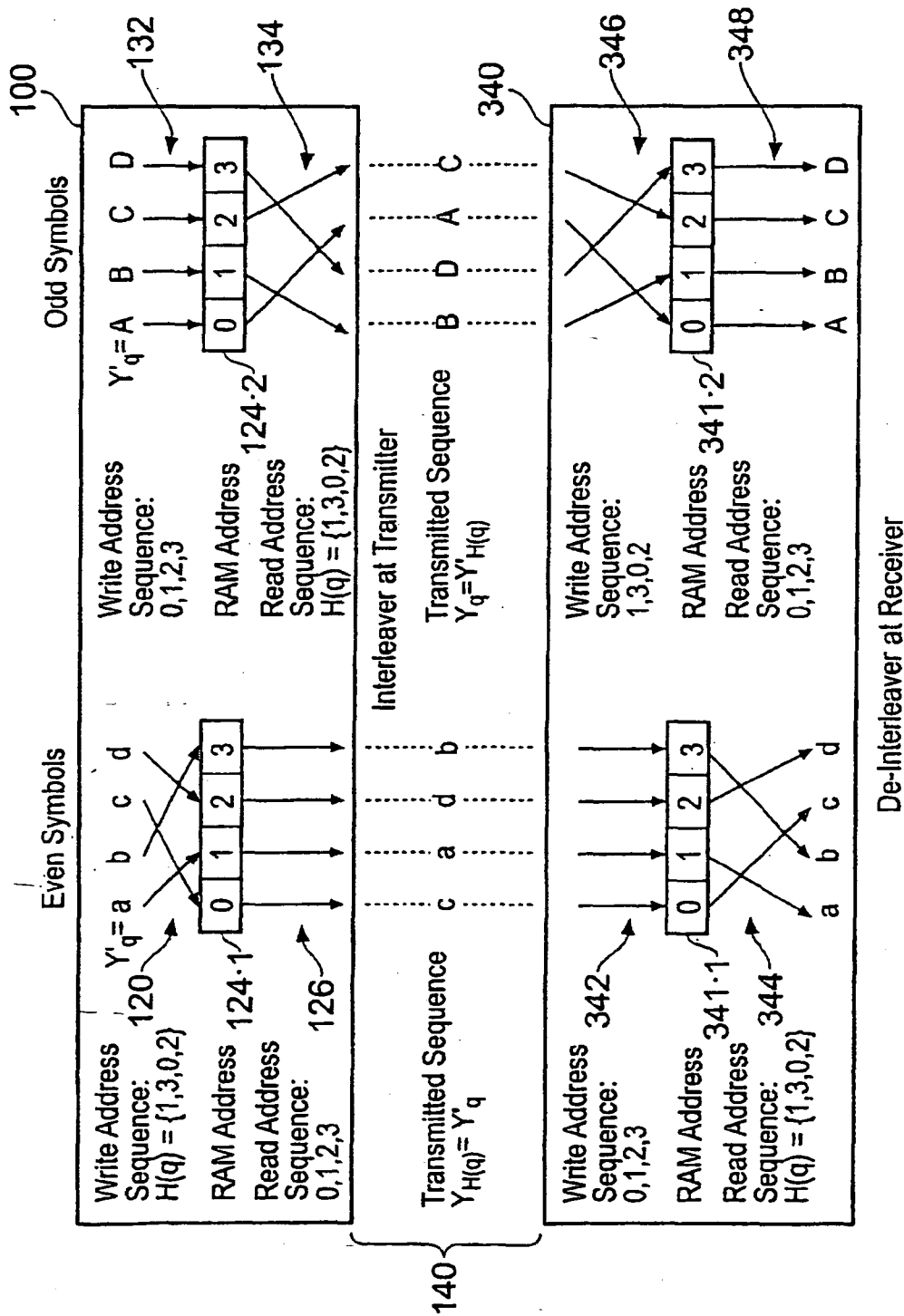


FIG. 4

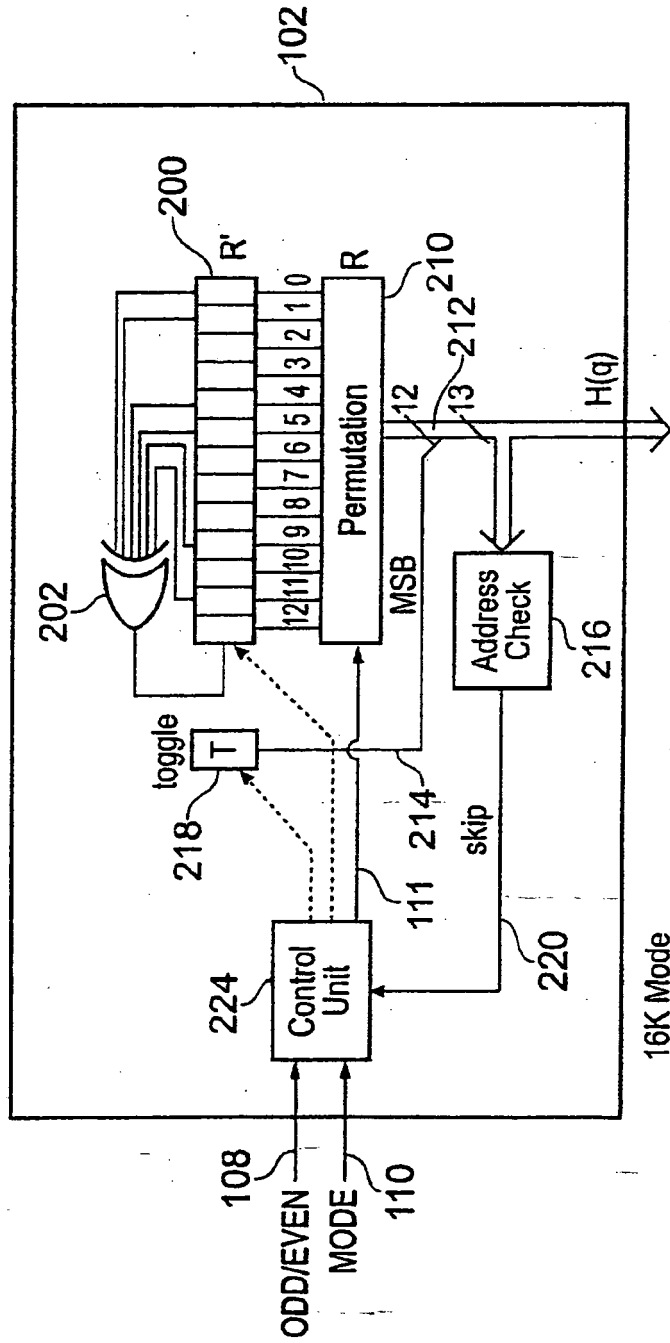


FIG. 5

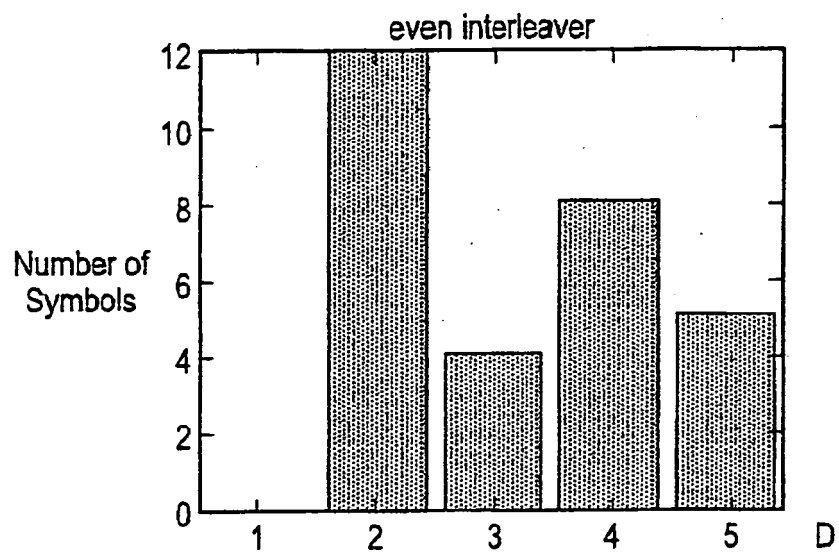


FIG. 6(a)

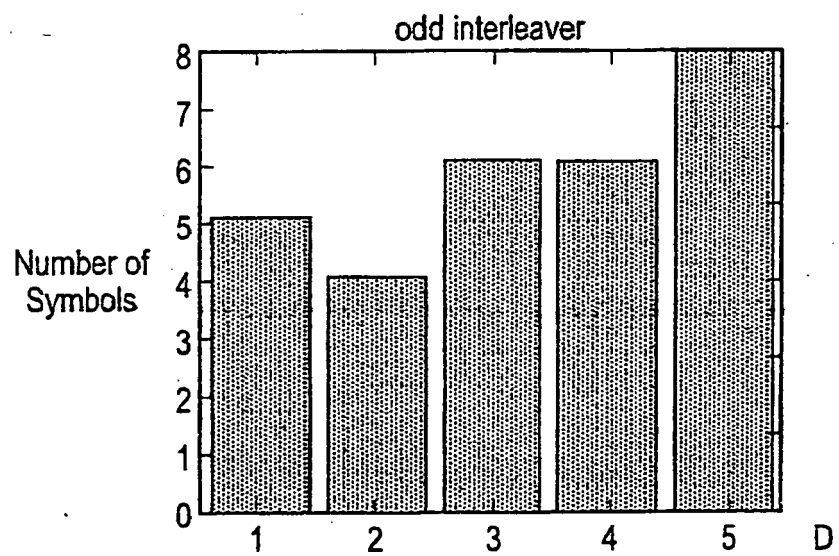


FIG. 6(b)

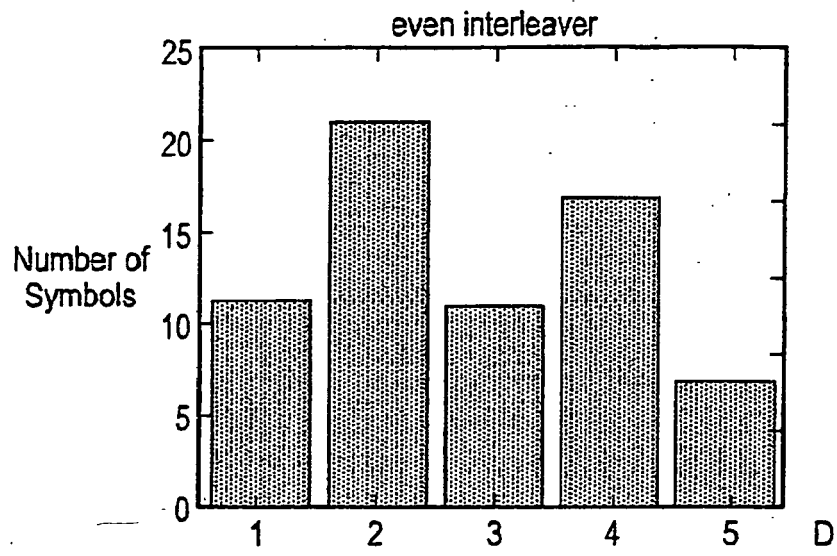


FIG. 6(c)

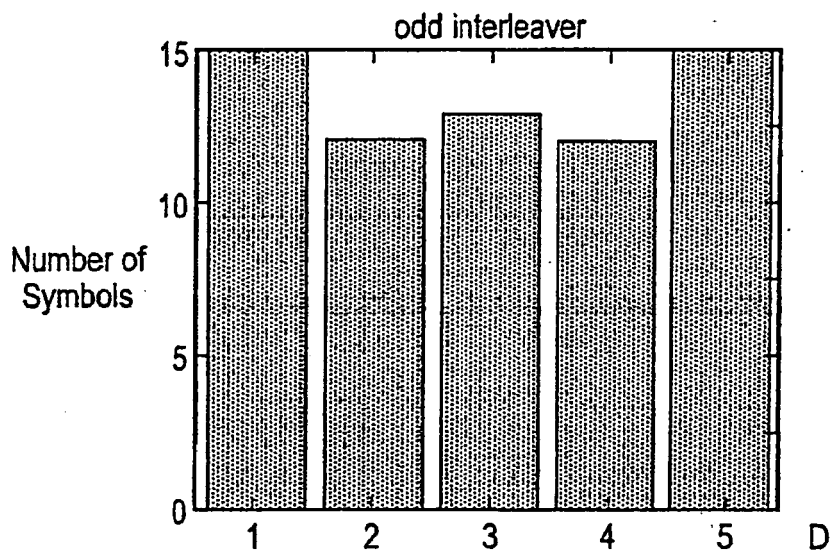


FIG. 6(d)

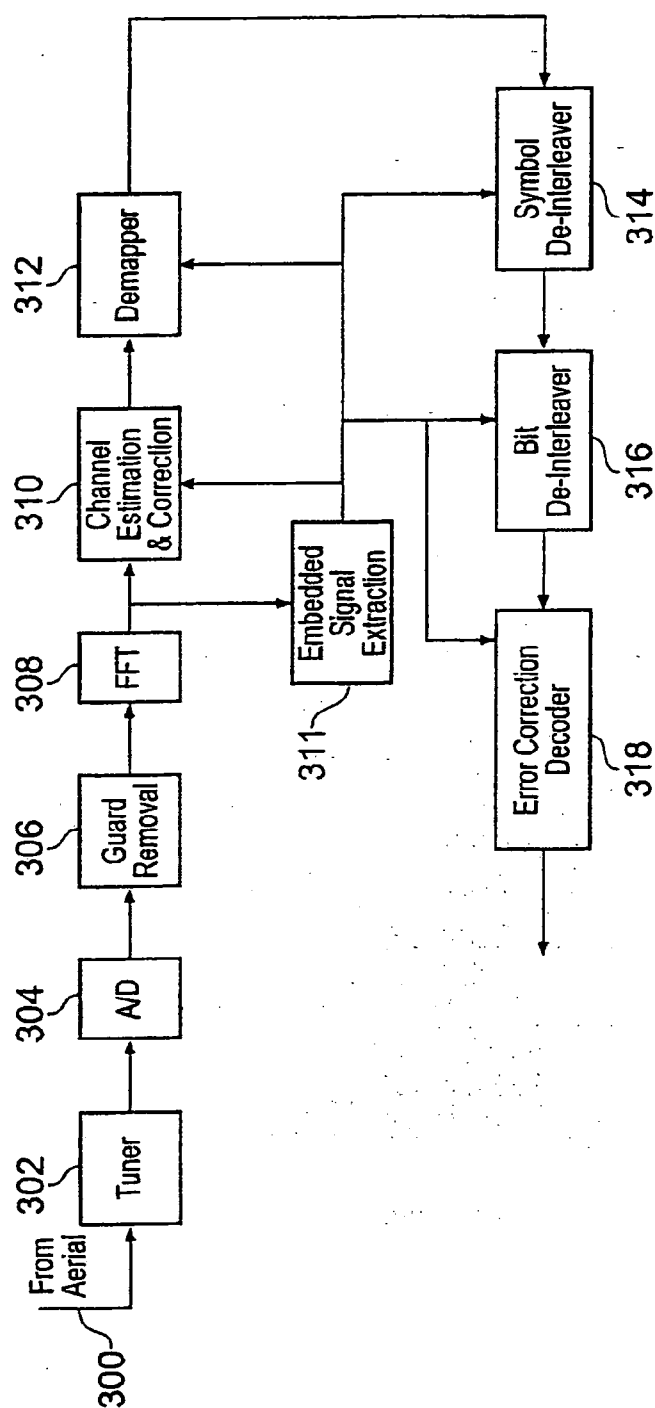


FIG. 7

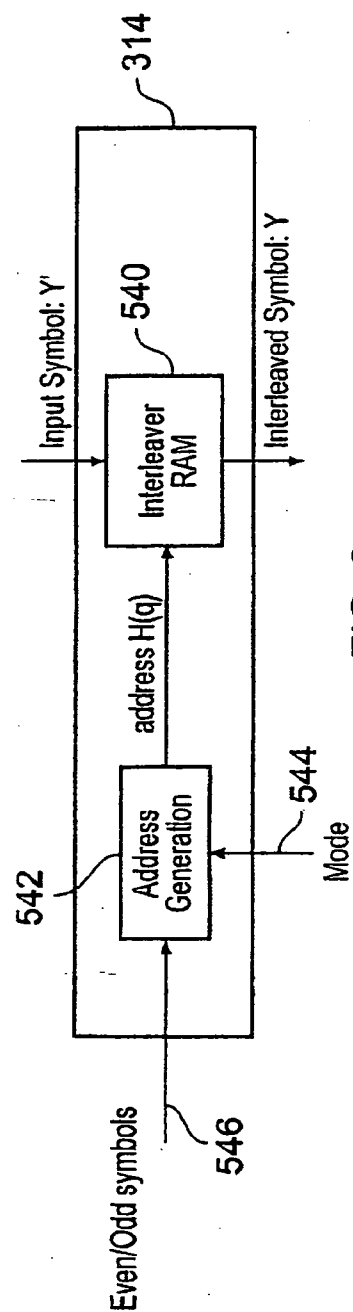


FIG. 8



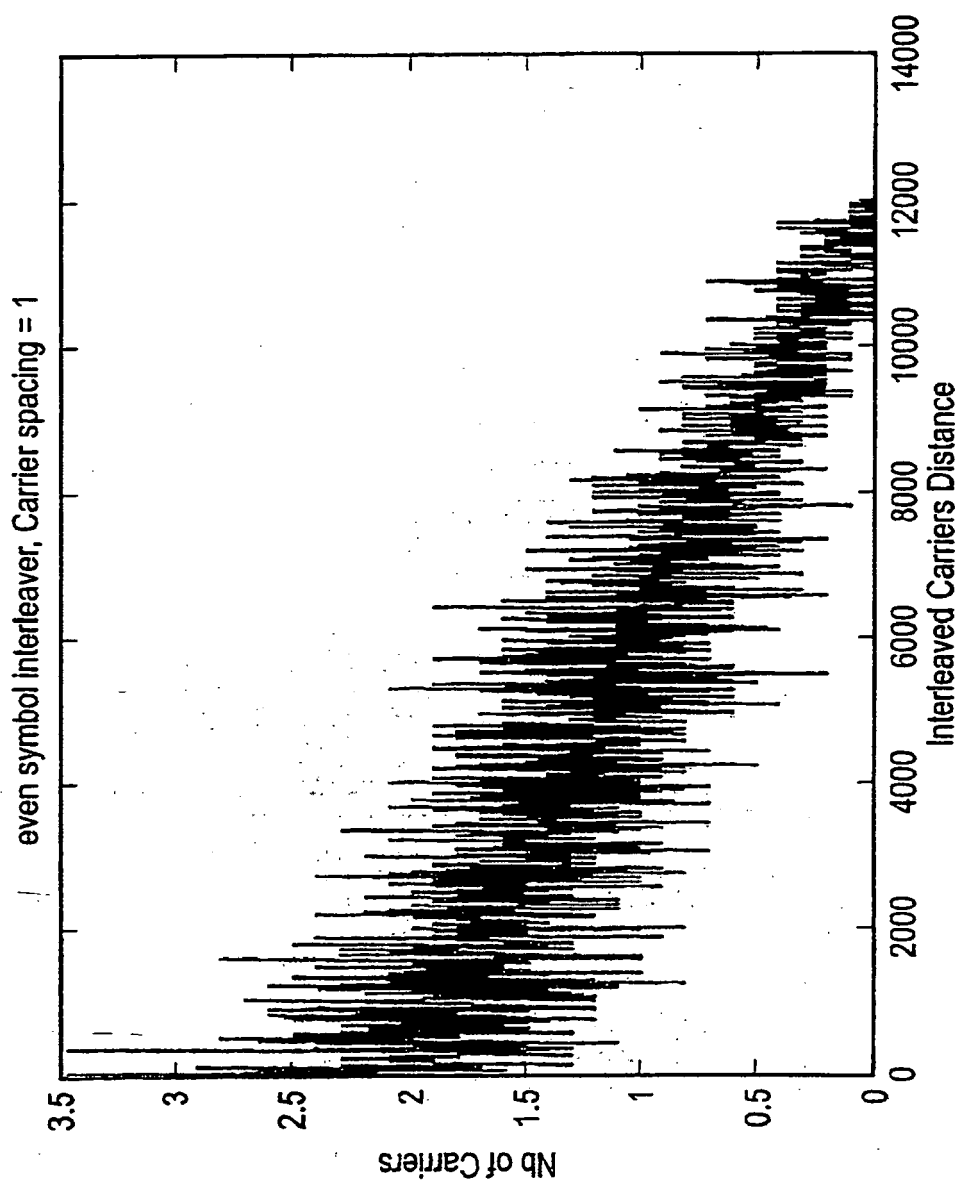


FIG. 9(a)

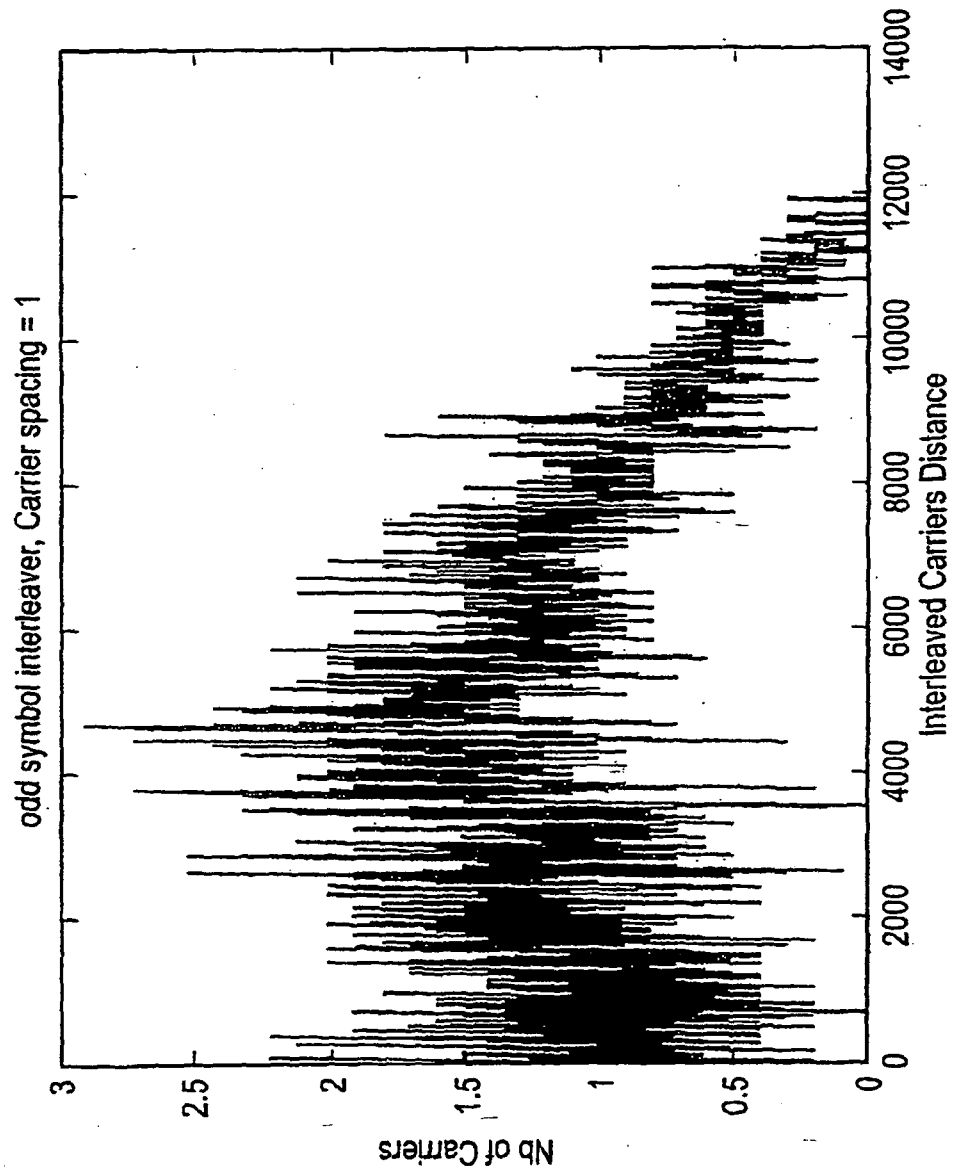


FIG. 9(b)

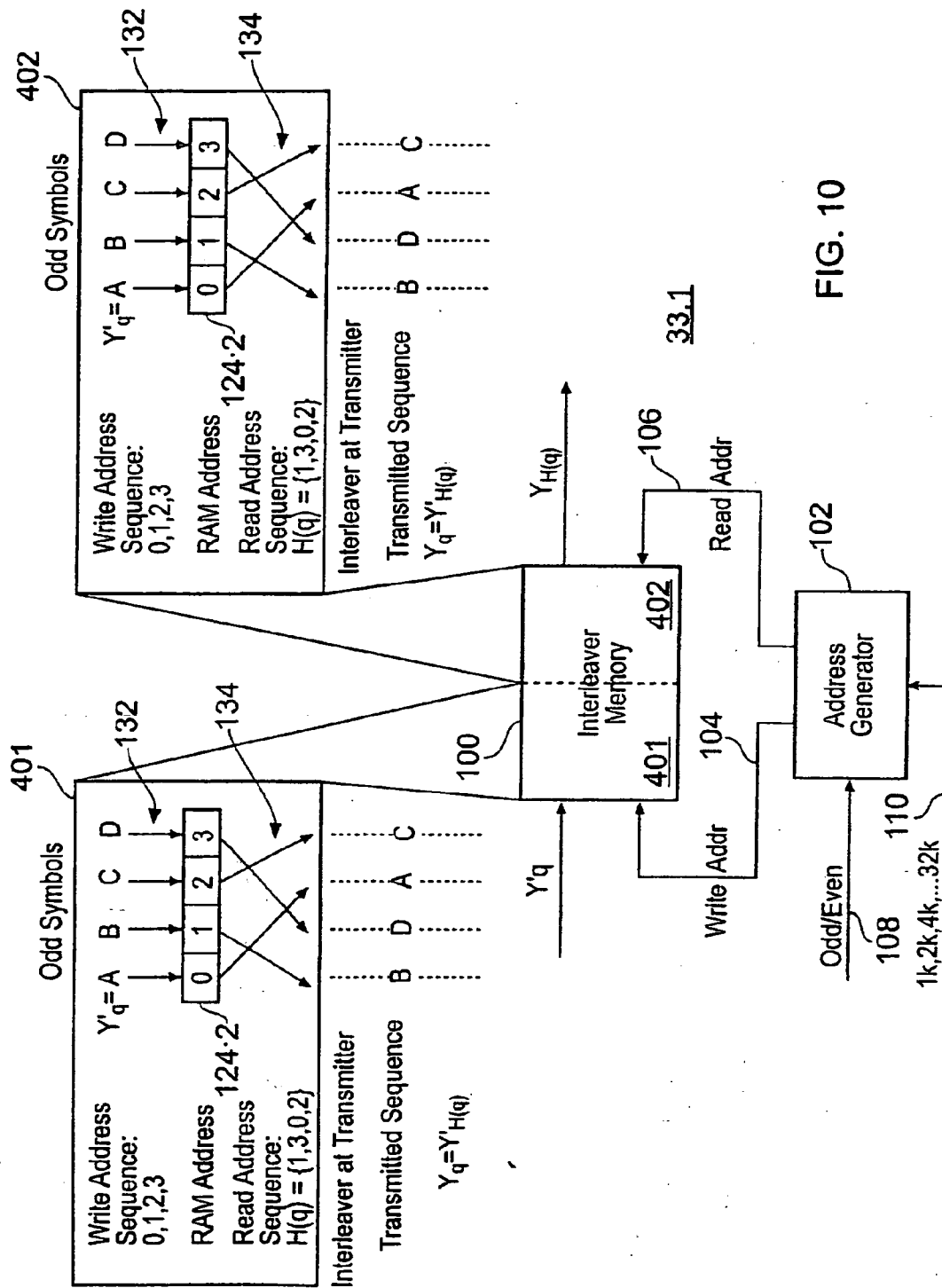


FIG. 10

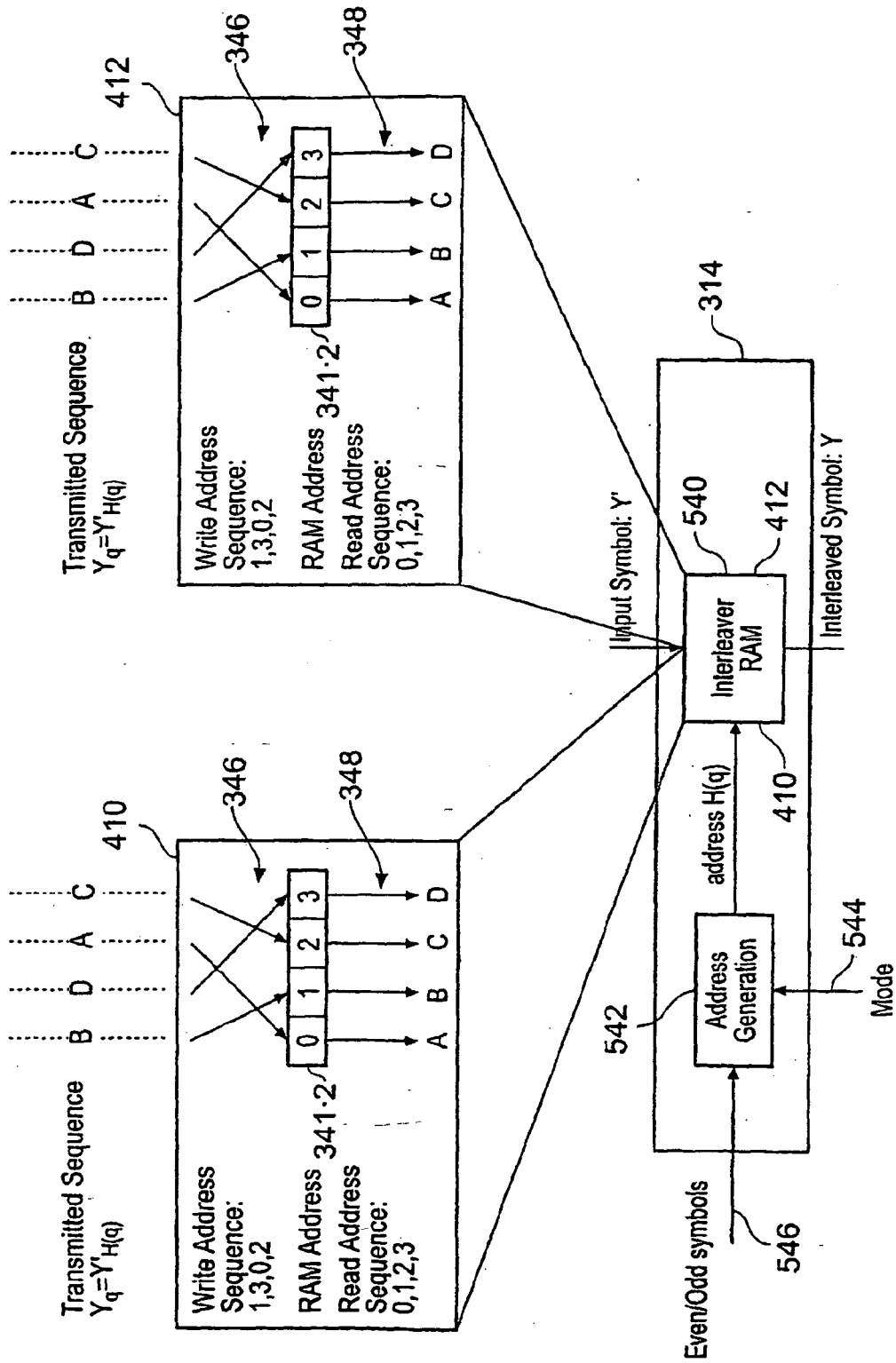


FIG. 11

**REFERENCES CITED IN THE DESCRIPTION**

*This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.*

**Patent documents cited in the description**

- EP 04251667 A [0005] [0080]
- EP 1463256 A [0007]
- US 6353900 B [0008]
- GB 0722553 A [0080]