

Efficiently transferable bit sequence with limited disparity

The present invention is directed to a method for generating an efficiently transferable bit sequence with a limited disparity and a limited run length. The proposed method allows data to be transmitted particularly efficiently via a transmission channel. Optimised disparity is a quality feature for the transferability of data. A disadvantageous disparity can lead to data not being transmitted properly via a data channel, as it cannot be interpreted properly by the recipient. Another quality feature is data efficiency with regard to the ratio of transmitted user data and a further data volume that is not directly related to the content of the user data. This includes so-called header data. The proposed invention makes it possible to generate data streams that can be read out particularly efficiently and are also very efficient in terms of overhead data. This minimises the so-called overhead of the user data, which in turn creates a particularly efficient process. The unambiguous interpretability on the receiver side also ensures that data does not have to be transmitted repeatedly, but can instead be read out on the receiver side with a high degree of error tolerance. In addition, the proposed method is particularly efficient, as data segments can be converted into sub-symbols or symbols made up of sub-symbols in parallel, and this parallel design only requires units that are technically simple to produce. Thus, the efficiency gain here also relates to the hardware or runtime to be used. The invention is also directed to a correspondingly configured system arrangement as well as to a computer program product and a memory-readable medium with control commands which execute the method.

EP 3323219 A1 shows a method which makes it possible to read out an analogue data stream via a data line in a particularly error-proof manner. Among other things, the amplitude of the signal is monitored and, particularly preferably, the signal is measured at the point at which the amplitude is at its maximum. This converts an analogue data stream into a digital data set and the maximum amplitude ensures that the threshold value between 0 and 1 on the line is reliably undercut or exceeded.

Various coding methods and data transmission methods are known from the state of

the art, but all of them relate to application scenarios that can only be used in a disadvantageous way in an automobile. For example, the state of the art often assumes that high computing power is available and that there are no high real-time requirements. In addition, the state of the art often assumes that the weight or
5 reliability of the components to be used plays a subordinate role. The state of the art often refers to conventional computer networks, where reliability and low technical complexity are less important.

Based on this state of the art, there is a need to create a method or a system arrangement that enables data to be processed as quickly as possible due to the
10 safety requirements in the automobile and also requires little technical effort and minimises the error rate during transmission, as retransmission is not possible if an error is detected. The low technical complexity should consist of installing components that are as simple as possible, have a low weight and can also be produced efficiently in large quantities. Known methods and system arrangements
15 from computer network technology cannot typically be used here, as weight savings and real-time running behaviour are not decisive for a stand-alone PC or server. Although the heat to be dissipated generally poses a challenge for computer arrangements, energy efficiency in an automobile is of even greater importance, as in electromobility, for example, power consumption even influences the range of the
20 car.

Further prior art relates to the transmission of data in a serial data stream. For example, the state of the art provides for extensive description data to be transmitted with the user data, indicating where the user data is located and how it is to be interpreted. In addition, it is known in the state of the art to discard individual data
25 packets if they are not transmitted correctly. It is also known in the state of the art to resend data packets if they do not arrive at a sender on time or in an unexpected format.

When transmitting data serially, it is necessary to have as many ones and zeros as possible in the serial data stream. This is called disparity. A disparity of zero on
30 average and also over a short period of time is desirable to avoid baseline drift

during transmission. The baseline drift (DC voltage fluctuation) of the serial signal leads to bit errors. In extreme cases, transmission is not possible.

A minimum number of 0->1 or 1->0 transitions is required to be able to reliably recover the serial bits in the serial data stream on the receiving end without the need to transmit a clock. This means that the clock for recovering the serial data is generated locally at the receiver from the serial data stream. The so-called run length specifies how many identical bits (ones or zeros) can occur in succession without changing. A short run length is always desirable, as long run lengths no longer allow the clock to be reliably recovered from the serial data stream.

The task of the line code (in this case a block code) is now to generate a symbol with guaranteed disparity and guaranteed run length from any data words with any disparity and infinite run length. This leads to an overhead during transmission. This means that more bits (in the form of symbols) have to be transmitted than occur in the net data word to be transmitted. This means that the required transmission speed (bandwidth) must be greater than the data rate of the data to be transmitted. This in turn means that systems require higher error rates or more effort, power, etc. than would be necessary to transmit the raw data.

The state of the art either has a high overhead (8B10B) or the quality of the coded signal in terms of disparity and run length is very poor, so that additional measures (complexity) such as scramblers are often necessary to improve the quality in terms of disparity or run length.

Accordingly, one task of the present invention is to create a method that generates a bit sequence that can be transmitted particularly efficiently. In this context, efficient can refer to hardware efficiency, efficient decryption on the receiver side, the lack of need for redundant data transmission due to signals that cannot be interpreted and/or to the ratio of user data to overhang data. Furthermore, according to the invention, it should be possible to create or use particularly efficient hardware that enables runtime optimisation through parallel processing. Furthermore, it is a task to provide a correspondingly equipped system arrangement, as well as a computer

program product and a computer-readable storage medium with control commands which execute the method or operate the system arrangement.

The problem is solved with the features of claim 1. Further advantageous embodiments are given in the subclaims.

- 5 Accordingly, a method is proposed in an automobile for generating an efficiently transmittable bit sequence with a restricted disparity and a restricted run length, comprising providing an arbitrary bit sequence; segmenting the provided bit sequence into a predefined sequence of segments according to a respective predefined bit length; and encoding each segment into a respective sub-symbol
10 using a respective encoding unit per segment from a plurality of encoding units, wherein a first subset of encoding units represents a sign of the bit sequence; and encoding each segment into a respective subsymbol, using a respective encoding unit per segment from a plurality of encoding units, wherein a first subset of encoding units actively controls a sign of the disparity of the subsymbol by inverting the
15 disparity of the generated subsymbol to compensate for a disparity of a second subset of encoding units, wherein a juxtaposition of the subsymbols results in the efficiently transmittable bit sequence.

In a preparatory process step, it is possible to provide a potentially infinite bit stream that contains the arbitrary bit sequence. Depending on the application scenario, the
20 data stream is already of any length and can in turn be divided into words or any bit sequence. This provides an output data stream with a bit sequence that is potentially of any length. However, this arbitrary length can be defined in a preparatory process step and can preferably be defined as 112 bits. As soon as the length or the bit length of the arbitrary bit sequence is defined, it is fixed in accordance with one
25 aspect of the present invention. In this respect, an arbitrary length of the bit sequence in the sense of the present invention cannot be understood as arbitrary. Rather, the synonym of the arbitrary bit sequence according to the invention may be a bit sequence whose length can be freely selected in advance and/or whose content corresponds to the data to be transmitted or at least part of the data to be
30 transmitted.

In a preparatory process step, it is therefore possible to provide an output data stream containing the arbitrary bit sequence. This arbitrary bit sequence is then read from the output data stream and provided in a first process step.

- Typically, the output data stream or the output bit sequence can have so many bits that the process is carried out iteratively in such a way that several arbitrary bit sequences are generated from the output bit sequence, segmented, converted into sub-symbols, optimised with regard to disparity and then transmitted. This means that the output bit sequence can also be of any length and can ultimately be transmitted in several overall symbols.
- 10 The coding units are plural, whereby each segment, which in turn corresponds to a part of the arbitrary or freely selectable bit sequence, is assigned a coding unit. This then converts the segment into a sub-symbol, whereby the set of sub-symbols concatenated corresponds to the symbol or the overall symbol of the bit sequence to be transmitted efficiently. Thus, according to one aspect of the present invention, an
- 15 encoding unit is located in a logical path of the processing chain or the structural arrangement between a segment and a partial symbol.

- The proposed method is particularly efficient, as the transferable bit sequence has a particularly high level of user data compared to the state of the art. For example, it is possible to transmit 128 bits, which have 112 bits of user data. The proposed
- 20 method is therefore already superior to the state of the art in this aspect. In addition, the creation of the transferable bit sequence is particularly efficient, as this can be carried out in parallel and coding units can be used for this purpose that are particularly simple in design. In this context, simple means, for example, that very few circuits need to be installed in the coding units. The coding units do not have to
- 25 have extensive logic and can even be optimised for a certain number of bits. According to the invention, it is thus possible for the input and the output of the respective coding unit to be fixed with regard to the number of bits.

Due to the optimised disparity of the bit sequence to be transmitted, it is possible to avoid errors when interpreting on a serial channel. Efficiency therefore also refers to

the fact that the bit sequence is particularly error-resistant and can therefore only be reliably transmitted once. Redundant transmission is avoided due to the high detectability, again due to the optimised disparity.

When transmitting data serially, it is advantageous to keep the number of ones and
5 zeros in the serial data stream as equal as possible. This is generally referred to as disparity. For reliable clock recovery at the receiver, a run length restriction can be imposed on the generated channel sequence. This limits the maximum number of consecutive ones and zeros. The proposed method can therefore also be described as a method for efficient coding of a bit sequence. According to the invention, the
10 disparity is optimised by cleverly setting partial disparities. This can be used particularly advantageously if the run length of the bit sequence is limited. The restricted disparity and the restricted run length can also relate to the arbitrary bit sequence provided. This means that it does not have to be the efficiently transferable bit sequence. Overall, the arbitrary bit sequence provided is efficiently
15 transferable or a bit sequence to be transferred is generated or created from this bit sequence, which is then efficiently transferable.

In a preparatory process step, an arbitrary bit sequence encoding user data is provided. Problems can occur in this arbitrary bit sequence, for example if there is an unfavourable disparity. For example, too many zeros can lead to problems during
20 transmission. This is to be avoided and therefore the arbitrary bit sequence is optimised in further process steps so that it can now be transmitted efficiently. The arbitrary bit sequence provided therefore represents any user data that is to be sent from a transmitter to a receiver via a serial data channel. The arbitrary bit sequence can, for example, be control data in an automobile.

25 According to the invention, the bit sequence provided is segmented into a predefined sequence of segments according to a predefined bit length. Thus, the input data stream, i.e. the arbitrary bit sequence, is divided according to a predefined procedure so that individual data segments are created. The segments therefore cumulatively result in the arbitrary bit sequence. The predefined bit length has the advantage that
30 coding units can be optimised in the same way that the respective bit length is taken

into account. This makes it possible to create particularly efficient circuits that are highly specialised. Defined bit lengths are mentioned below, but these are merely examples.

Each segment is encoded into a sub-symbol using one encoding unit per segment
5 from a plurality of encoding units. The coding itself takes place in one coding unit, which finds one segment at the input and then converts this segment into a partial symbol. The partial symbol is also a bit sequence. Overall, the arbitrary bit sequence is therefore divided into segments, these segments are each converted into a sub-symbol by a coding unit and the totality of the sub-symbols results in the coding of
10 the arbitrary bit sequence to be transmitted. Overall, it is advantageous that the number of segments corresponds to the number of coding units and thus to the number of partial symbols. It is therefore possible for each segment to have exactly one coding unit, which in turn generates exactly one partial symbol from the segment. The number of coding units describes all the coding units to be used,
15 which corresponds to the number of segments. The number of segments is predefined, as a predefined bit length is specified. The method is therefore deterministic overall.

In order to achieve a favourable disparity overall, there is a first subset of coding units that actively controls the sign of the disparity of the sub-symbol by inverting the
20 disparity of the generated sub-symbol to compensate for a disparity of a second subset of coding units. This therefore means that there are subsets of coding units that control the sign of the disparity or not. Coding units of the first subset control this sign and coding units of the second subset do not control it. Coding units of the first subset can therefore be described as active and coding units of the second subset
25 as passive. Due to the different subsets or types of coding units, it is possible to connect the coding units in series in such a way that coding units of the first subset also favourably shape the overall disparity of the partial symbols from the coding units of the second subset.

According to the invention, therefore, coding units are used which are arbitrary with
30 respect to the disparity of the subsymbol. A coding unit of the first subset can then

be connected in parallel, which controls the sign of the disparity of the preceding coding unit and its own coding unit or its subsymbols depending on the disparity of the coding unit of the second subset. This means that a certain number of coding units of the second subset are used and then another certain number of coding units of the first subset are used. In this way, the types or subsets of the coding units change in such a way that the next coding unit connected in parallel adapts the disparity of the previous subsymbol(s) and/or its own subsymbol. In this way, it is avoided that non-controllable coding units are connected in parallel one after the other in such a way that an unfavourable disparity occurs. Each coding unit of the first subset thus corrects the sign of the previously parallel-connected coding units or their sub-symbols. This parallel connection of coding units is described in more detail below with reference to Figure 4.

In summary, it can therefore be concluded that coding units of the first subset optimise coding units of the second subset with regard to disparity. Optimising a disparity means that the disparity is 0. How the skilled person calculates disparities or sets them, for example by changing the sign, is sufficiently well known to the skilled person.

According to the proposed method, sub-symbols are therefore created which are optimised in their sequence with regard to disparity. As each segment is converted into a sub-symbol, the (overall) symbol to be transmitted can be generated by stringing together the sub-symbols. This can be transferred particularly efficiently or error-tolerantly, as the disparity or partial disparities are optimised. This results in a particularly favourable bit sequence to be transmitted.

According to one aspect of the present invention, the arbitrary bit sequence is unrestricted in its disparity and run length. This has the advantage that any amount of user data can be transmitted or can be converted into a bit sequence which is restricted in its disparity and restricted in its run length. This means that any bit sequence is encoded into a bit sequence to be transmitted, which is optimised in terms of its disparity and run length.

According to a further aspect of the present invention, the disparity of the second subset of coding units is not controllable. This has the advantage that any partial symbol can be generated using the coding units of the second subset, whereby particularly simple coding units are to be used. These can be designed to be particularly simple, as the generated partial symbol of this coding unit is not subject to any restrictions in terms of disparity or run length.

According to a further aspect of the present invention, the active control of the sign is performed in each case by means of conditional inversion of the subsymbol. This has the advantage that the corresponding subsymbol of the coding units of the first subset can be controlled in a simple manner. Only the disparity or individual bits of the partial symbol need to be inverted. The sign refers to the disparity of the partial symbol, which can be positive or negative.

According to a further aspect of the present invention, the conditional inversion takes place as a function of the disparity of an overall symbol, which is formed from all partial symbols. This has the advantage that not only partial symbols are optimised, but also the entire, i.e. the combined partial symbols, i.e. the overall symbol is optimised with respect to the disparity. This results in a particularly favourable overall symbol.

According to a further aspect of the present invention, the dependency is influenced in such a way that a magnitude value of the disparity is minimised. This has the advantage that the lowest possible disparity, preferably 0, is achieved. Disparities are thus linked with one another in such a way that the amount of the disparities is as low as possible or as low as possible.

According to a further aspect of the present invention, the absolute value is minimised in such a way that a positive total symbol disparity is counteracted by a negative parity of the sub-symbols. This has the advantage that the positive total symbol disparity is minimised or eliminated.

According to a further aspect of the present invention, the absolute value is minimised in such a way that a negative overall symbol disparity is counteracted by a positive parity of the sub-symbols. This has the advantage that the total symbol disparity is minimised or eliminated overall.

- 5 According to a further aspect of the present invention, active control takes place as a function of already transmitted total symbols, such that the disparity of all total symbols is minimised. This has the advantage that several total symbols are minimised with regard to their disparity or the disparity is eliminated and thus several sequences of total symbols are also optimised with regard to their transferability.
- 10 According to a further aspect of the present invention, coding units of the first subset encode segments of 11 bits to subsymbols of 13 bits. This has the advantage that 11 bits are encoded particularly efficiently and only an additional expenditure of 2 bits is incurred for this. This is particularly advantageous if a total symbol of 128 bits is to be created. In general, the specific values of the proposed technical teaching
- 15 mentioned here were determined empirically and can be proven by the fact that only an additional effort of $128 - 112$ bits, i.e. 14 %, is necessary. Thus, the values listed here are proven to be advantageous for a transmission of 112 bits.

- According to a further aspect of the present invention, coding units of the first subset have a disparity between +3 and +9, which are specifically inverted to -3 to -9 by
- 20 inversion. This has the advantage that, for example, a disparity of +3 is cancelled with a disparity of -3, which is done analogously when adjusting the disparity of +9 with a disparity of -9. This in turn is particularly advantageous for an arbitrary bit sequence of 112 bits, which is to be coded as 128 bits.

- According to a further aspect of the present invention, the run length in sub-symbols
- 25 is a maximum of 7. This has the advantage that a maximum of 7 equal instances of zeros and ones are generated, which is particularly advantageous in the proposed scenario of 112 bits or 128 bits.

According to a further aspect of the present invention, the run length of the

subsymbol for coding units of the first subset is a maximum of 5, starting from the most significant and/or the least significant bit. This has the advantage that a maximum of 5 identical bits can be present at the end or at the beginning of a subsymbol. This has also proved to be particularly advantageous in the scenario
5 described.

According to a further aspect of the present invention, coding units of the second subset encode 11 bit segments on 12 bit sub-symbols or 7 bit segments on 8 bit sub-symbols or 6 bit segments on 8 bit sub-symbols. This has the advantage that this coding produces particularly advantageous values in the scenario of 112 bits to be
10 transmitted in a symbol of 128 bits, which have a minimised overhang of only 14%.

According to a further aspect of the present invention, a disparity between -2 and +2 is generated for coding units of the second subset. This has the advantage that again particularly favourable disparities are generated.

According to a further aspect of the present invention, in the case of coding units of
15 the second subset, the run length in the generated subsymbol is 6. This has the advantage that, again, particularly optimised subsymbols are generated.

According to a further aspect of the present invention, encoding units of the second subset generate partial symbols which have a maximum run length of 3 at the edge. This has the advantage that partial symbols which are created by the second subset
20 of coding units have a maximum run length of 3 at the beginning or at the end, which is a particularly advantageous value.

According to a further aspect of the present invention, the coding units are addressed in parallel and one segment is coded into a sub-symbol in each case. This has the advantage that a segment is converted into exactly one sub-symbol
25 with exactly one coding unit. The coding units can therefore be addressed in parallel, since segments are formed from the arbitrary bit sequence, which can be converted into partial symbols in parallel.

According to a further aspect of the present invention, the coding units are addressed in the sequence 21212221212, where a 1 stands for a coding unit of the first subset and a 2 for a coding unit of the second subset. This has the advantage that a non-actively controlled number of coding units is always followed by a single coding unit which can be actively controlled. It has been empirically determined that the proposed coding units result in a particularly favourable overall symbol.

According to a further aspect of the present invention, when a positive data stream with respect to the disparity and a negative data stream are applied, a multiplexer selects the data stream that contributes to minimising the overall disparity of the overall symbol. This has the advantage that a suitable data stream can be selected which has a sign that minimises or eliminates the disparity. If, for example, the disparity to be optimised is negative, a data stream with positive disparity is selected, which then minimises or compensates for this data stream with regard to its disparity.

One aspect of the present invention is to bundle several data streams (video, audio and data) in one transport frame and transmit them serially. The different data formats not only have different bandwidth requirements, but also different latency, reassembly sublayer and bit error rate requirements. In particular, the transmission of today's video data formats requires not only the transmission of pure video data and its frame information, but also the support of encryption methods such as HDCP. All of this requires many different data channels with a wide range of requirements in terms of bandwidth, latency, reassembly sublayer, etc. Added to this is the desire for far more complex network architectures than a simple transmitter-receiver architecture offers. Architectures with several repeaters, where data paths can begin and end, branches (Y) and the possibility of reintegrating data paths into a link are advantageous.

According to one aspect of the present invention, the technology follows the basic idea of consistently bundling services, but offers completely new possibilities with regard to network architectures and allows new approaches in the implementation of today's video interfaces. In addition, it can be used as a universal data transport

layer, for example also for the transmission of Ethernet or camera data or any type of sensor data.

With a virtual path, all packets/cells take the same path, in contrast to IP, where a packet could reach its destination via a different route than previous and subsequent packets. Latency and reassembly sublayers over a virtual path are therefore constant.

Virtual paths also have the advantage that they can be used as multiplexing offshoots for different services (video, audio, Ethernet), as the properties of the virtual paths can be configured differently without the different virtual paths interfering with each other.

Virtual paths only consume bandwidth when data is actually being transferred.

The concept of virtual paths also makes it possible to realise complex and far-reaching diagnostic and network configuration functions at runtime with separate (virtual) data channels.

According to one aspect of the present invention, a virtual path layer is provided between the physical layer (serialiser and framer) and the various application data interfaces.

According to one aspect of the present invention, this is used to multiplex the various data paths and support more complex architectures with repeaters and branches.

This is mainly done in the cell layer.

According to an aspect of the present invention, a further part of the virtual path layer is an application adaptation layer which performs the conversion of video (stream) or, for example, Ethernet (packet) data into the cells. This application adaptation layer also comprises the OAM functions for network diagnostics and management.

According to one aspect of the present invention, the technology can be the basis for transmitting a variety of data formats via a serial connection in the car (and elsewhere). It thus forms the basis for a new generation of devices.

5 The high serial bandwidths make it necessary to define architectures, cell formats and interfaces that enable flexible internal data bus widths in order to adapt the speed of the internal timing system to the possibilities of the chip technology.

According to one aspect of the present invention, the virtual path layer is the physical layer comprising the transmission sublayer and the physical medium sublayer, the cell layer and the application adaptation layer comprising the segmentation and
10 reassembly sublayers and the functions for adapting the data formats to the corresponding application.

The main task is to establish the physical connection to other physical layers. This connection is basically bidirectional. Theoretically, this connection can be realised via a wide variety of media. In practice, two serial differential GBps connections are
15 used. In this layer, the line coding, the insertion of empty cells to decouple the cell rate from the connection rate and the integration of the cell stream into the serial frame take place.

In the cell layer, the segmented data (cell payload data) of the segmentation & reassembly sub-layer above is assembled into complete cells with header, VP
20 identifier and CRC or cells are CRC-checked and the payload is passed on to the segmentation and reassembly sub-layer. This is also where the different cell streams of the application customisation functions are multiplexed and the cell payloads are distributed to the application customisation functions according to the VP identifier.
(Feed-in / Feed-out)

25 According to one aspect of the present invention, multiplexing and demultiplexing of cell streams in repeaters and splitters (forwarding) also takes place in the cell layer.

According to one aspect of the present invention, the task of the application adaptation functions is to adapt the data of the application interfaces to the format of the user data field of the cell and to transmit control information to the opposite side or to transmit control information of the opposite side for the adaptation to be used
 5 (time generation, frame formation).

According to one aspect of the present invention, all virtual data paths are unidirectional, i.e. they start at an initiator and end at one or more targets. If virtual data paths logically belong together, e.g. HDCP for a video channel, and thus form a bidirectional data path, these paths should have the same VP identifiers.

10 The virtual data path begins with an initiator and ends with one or more targets. It is realised by the

cell sublayer and executes the following functions on the virtual path:

- Add/delete multiplexing
- VP translation

15 Stream data (continuous data stream)

The stream data function combines the time domain crossing and the bit width conversion of data from the application interface to the N bits of the cell rows. The cell row payload is already preformatted so that the cell footer and header fit into the first and last cell row.

20 Streamed data is (normally) source-synchronised. The clock domain crossing of the data path from the application clock domain to the Virtual Path Layer clock domain is performed here.

A data buffer is provided in the transmission direction, into which the source-synchronised data is written with the source clock. The segmentation layer retrieves

the data from this buffer as required in order to perform the data format conversion into the N-bit wide rows of the cells. Frame data (e.g.: Hsync, Vsync, DE) is coded in payload info bits so that the frame can be reconstructed on the receiver side.

- 5 In the receive direction, the cell data from the reassembly sublayer is written to a data buffer, whereby the cell row bit width is (with cell row bit width). The frame information is reconstructed based on the payload info bits. The source clock is regenerated, for example, using buffer fill level and clock synthesis.

If data encryption is required (HDCP), the cell data is encrypted or decrypted in this function.

- 10 Due to the different types of streamed data, such as: Audio, video with and without encryption, there may be different implementations of this basic function (e.g.: VStream In/Out; AStream In/Out; EncVStream In/Out).

The interface to the segmentation & reassembly sub-layer is the same for all functions.

- 15 Burst data (discontinuous data stream)

The burst data function combines the clock domain crossing and data bit wide conversion of data from the application interface to the N bits of the cell rows. The cell row payload is already preformatted so that the cell footer and header fit into the first and last cell row.

- 20 Burst data is (normally) synchronised to an external time and has different identification signals for direction and data type (address / data / ByteEnable).

This data is normally accompanied by control lines in order to realise a specific protocol.

A data buffer is provided in the transmission direction, into which the burst data is written with the interface clock. The segmentation layer retrieves the data from this buffer as required in order to perform the data format conversion into the N-bit wide row cells.

- 5 In the receive direction, the cell data from the reassembly sublayer is written to a data buffer, whereby the cell row is bit-wide. The interface control signals are reconstructed based on the payload info bits.

The payload info bits are used to generate the control signals of the application-specific interfaces or to synchronise the protocol state machines in the application-specific interfaces.

Due to the different interfaces that provide burst-like data (SPI, I2C, MII), there may be different implementations of this basic function (e.g.: SPIBurst, I2CBurst, MIIBurst).

Accordingly, there will also be (slightly) different stream in/out interfaces, although their structure should be the same.

The problem is also solved by a system arrangement in an automobile for generating an efficient transmittable bit sequence with a restricted disparity and a restricted run length, comprising an interface unit set up for providing any bit sequence; a segmentation unit set up for segmenting the provided bit sequence into a predefined sequence of segments according to a respective predefined bit length; and a coding arrangement set up for coding each segment into a respective sub-symbol, using a respective coding unit per segment from a plurality of coding units, wherein a first subset of coding units actively controls a sign of the disparity of the sub-symbol by inverting the disparity of the generated sub-symbol to equalise a disparity of a second subset of coding units, wherein a juxtaposition of the sub-symbols results in the efficiently transmittable bit sequence.

The task is also solved by a computer program product with control commands that implement the proposed method or operate the proposed device.

According to the invention, it is particularly advantageous that the method can be used to operate the proposed devices and units. Furthermore, the proposed devices
5 and units are suitable for carrying out the method according to the invention. Thus, in each case the device implements structural features which are suitable for carrying out the corresponding method. However, the structural features can also be designed as process steps. The proposed method also provides steps for implementing the function of the structural features. In addition, physical components
10 can also be provided virtually or virtualised.

Further advantages, features and details of the invention are apparent from the following description, in which aspects of the invention are described in detail with reference to the drawings. The features mentioned in the claims and in the description may be essential to the invention individually or in any combination.
15 Likewise, the above-mentioned features and the features further described herein may be used individually or in any combination. Functionally similar or identical parts or components are sometimes provided with the same reference signs. The terms "left", "right", "top" and "bottom" used in the description of the embodiments refer to the drawings in an orientation with a normally legible figure designation or normally
20 legible reference signs. The embodiments shown and described are not to be understood as conclusive, but are of an exemplary nature to explain the invention. The detailed description is for the information of the skilled person, therefore known circuits, structures and methods are not shown or explained in detail in the description, so as not to complicate the understanding of the present description.
25 The figures show:

Figure 1: A schematic flowchart of a method for generating an efficient transferable bit sequence with a restricted disparity and a restricted run length according to an aspect of the present invention;

Figure 2: a basic frame format and the application of a so-called block code

according to one aspect of the present invention;

Figure 3: a schematic diagram of an assembly and structure of the so-called block code according to a further aspect of the present invention;

Figure 4: a schematic block diagram of the system arrangement for generating
5 an efficiently transferable bit sequence with a limited disparity and a limited run length according to a further aspect of the present invention;

Figure 5: a schematic diagram of a frame format as it may be used according to the invention; and

Figures 6A, 6B, 6C, 6D: exemplary encodings of data segments to symbols such
10 that the disparity is optimised according to an aspect of the present invention.

Some of these figures show parameters which are familiar to the skilled person in the English designation and which are used as parameters and cannot be translated in this way.

Figure 1 shows in a schematic flow chart a method in an automobile for generating
15 an efficiently transferable bit sequence with a restricted disparity and a restricted run length, comprising providing 100 an arbitrary bit sequence; segmenting 101 the provided bit sequence into a predefined sequence of segments according to a predefined bit length in each case; and encoding 102 each segment into a respective subsymbol, using a respective encoding unit per segment from a plurality of
20 encoding units, wherein a first subset of encoding units actively controls a sign of the disparity of the subsymbol by inverting the disparity of the generated subsymbol to equalise a disparity of a second subset of encoding units, wherein a juxtaposition of the subsymbols results in the efficiently transmittable bit sequence.

Figure 2 shows a data format that displays an arbitrary bit sequence on the left-hand
25 side and sub-symbols on the right-hand side. The data to be encoded has 112 bits and the sub-symbols have 128 bits. In this way, an arbitrary bit sequence of 112 bits

is encoded to a total symbol of 128 bits. The coded 128 bits are optimised in terms of disparity. The arrow in the centre indicates that the coding units translate the data segments on the left-hand side into partial symbols on the right-hand side. The figure also shows that the method can be used multiple times, meaning that any number of
 5 bit sequences can be translated into any number of total symbols. In addition, the data can be divided into different data cells or data frames.

Even if the data on the left-hand side has the same semantic content as the data on the right-hand side, the data on the right-hand side is encoded in such a way that its disparity is optimised. In general, this method can be applied to any data, hence the
 10 arbitrary bit sequence, and both user data and header data can be transferred.

The data fields entered are to be understood merely as examples and form an application example of the present invention.

Figure 3 shows the coding units in the centre together with the inputs and outputs. The output data consists of 112 bits, which in this case have the index 0-111. These
 15 are broken down into segments which have 11, 6, 7 or other bit length assignments. In this example, these segments are translated into sub-symbols of 12, 13, 8 or other data lengths. The proposed example is particularly advantageous because it encodes 112 bits into 128 bits, thereby achieving a particularly high degree of efficiency. The 128 bits have the same content as the bit sequence to be encoded
 20 and are only 16 bits longer.

The figure on the left shows that the arbitrary bit sequence of 112 bits is segmented into data segments of 11 bits and then encoded into 12 bits by means of the encoding unit 11B12B.

Figure 4 above shows the arbitrary bit sequence of 112 bits, which is segmented into
 25 11, 6 or 7 bits. The coding units are then addressed in parallel, which convert the bits into sub-symbols in such a way that they are optimised in terms of disparity. For example, 11 bits are coded after 12 bits or 11 bits after 13 bits.

In this Figure 4, a coding unit from the second subset is shown on the far left-hand side in the centre, which is labelled 11B12B. This provides a partial symbol with any sign, i.e. with any disparity. To compensate for this disparity, the coded unit 11B13B is connected downstream in parallel with respect to the bit sequence. This means

5 that a data stream is formed which converts the most significant bits of two times 11 into two sub-symbols, namely once the 11-bit data segment is formed by 11B12B into a 12-bit sub-symbol with any sign, i.e. disparity, and once the data segment is encoded from 11 bits into 13 bits by means of the encoding unit 11B13B. In the figure below the coding units, the second coding unit from the left 11B13B is a

10 coding unit of the first subset. This has an inverter and a multiplexer. The first 13 bits are therefore available as a data stream, which is divided in such a way that it is inverted once with regard to the sign, i.e. the disparity, and once remains unchanged. It is shown below that the positive or negative, i.e. the original or inverted data stream that equalises the sign from the leftmost coding unit is used

15 under disparity feedback. There are therefore two data streams at the first multiplexer on the left, each of which represents the partial symbol, one with a conventional sign, i.e. as it is output from the 11-bit 13-bit coding unit, and one with an inverted sign or inverted disparity.

Based on the feedback from the unit at the top, the disparity resulting from the

20 leftmost encoder 11B12B is thus determined and the multiplexer, at the bottom left, compensates for or minimises the disparity of the partial symbol of the leftmost encoder 11B12B. This is carried out in parallel in such a way that the coding units from the second subset are followed by coding units from the first subset, which minimise or eliminate the disparity. Finally, the total symbol is output at the bottom

25 right. This total symbol has 128 bits and is made up of the sub-symbols as they are inserted into the bold line below using the slanted arrows. This output line therefore contains the partial symbols that are optimised or minimised in terms of disparity and these partial symbols form the entire symbol, which can then be output and transmitted.

30 According to one aspect of the present invention, the invention encodes a 112-bit wide data word, with any disparity (disparity max: 112) and any run length (run

length max: 112), on a 128-bit wide symbol. Thus, the overhead resulting from the encoding is 14.2%.

The maximum run length occurring in the symbol, as well as with any sequencing of any symbols, is 8 identical bits.

- 5 The maximum disparity in the long mean is 0. The disparity in a symbol is less than 9.

The complexity of the logic is minimal, comparable with 10 8B/10B encoders (with the well-known disadvantage of the large overhead).

- 10 This is achieved by the use, or parallel use, of several "small" encoders that are optimally matched to each other in terms of their disparity and run length characteristics.

The encoders 11B12B, 7B8B and 6B8B generate all symbols with a guaranteed maximum run length of 6, even with any sequencing of the (partial) symbols.

- 15 According to one aspect of the present invention, the encoder (11B13B) generates symbols with a guaranteed maximum run length of 7 or at the beginning or end of the symbol of 5. Due to the sequencing (Fig2) of (11B13B) with the other encoders, a maximum run length of 8 can thus be generated in the symbol.

See properties of the encoders, as follows:

- 20 - 11B13B: 11-bit data is mapped to 2048 symbols with 13 bits. The symbols can be transmitted in inverted or non-inverted form.

Disparity:

+3...+9 or controllable -3...-9

Running length in word: 7

Running length at the edge: 5.

- 5 - 11B13B: 11-bit data is mapped to 2048 symbols with 13 bits. The symbols can be transmitted in inverted or non-inverted form.

Disparity: +3...+9 or controllable -3...-9

Running length in word: 7

Running length at the edge: 5

10

- 11B12B: 11-bit data is mapped to 2048 symbols with 12 bits. The symbols are only transmitted non-inverted.

Disparity: -2, -1, 0, 1, 2

15

Running length in word: 6

Running length at the edge: 3

- 7B8B: 7-bit data is mapped to 128 symbols with 8 bits. The symbols are only transmitted non-inverted.

20

Disparity: -2, -1, 0, 1, 2

Running length in word: 6

Running length at the edge: 3

- 25 - 6B8B: 6-bit data is mapped to 64 symbols with 8 bits. The symbols are only transmitted non-inverted.

Disparity: 0

Running length in word: 6

30

Running length at the edge: 3

With the four 11B13B encoders, a controllable disparity of at least ± 12 can be generated in order to compensate for the non-controllable disparity of a maximum of

± 12 ($6 \times \pm 2$) of the 11B12B and 7B8B encoders, so that a balanced disparity can be reliably achieved regardless of the data to be transmitted.

In order to further reduce the complexity of the hardware, according to one aspect of the present invention, 4 small encoders (11B13B) are used, the disparity of which
5 can be controlled with respect to sign ($\pm$).

According to one aspect of the present invention, the symbol disparity is controlled in such a way that each encoder calculates the parity of "its" subsymbol. This is done with little effort, since the subsymbol has only a few bits.

With four of the eleven encoders, the sign of the disparity of the subsymbol can be
10 actively controlled by inverting the generated subsymbol. For this purpose, the encoders (11B13B) have the special feature that their symbols generate a symbol with positive disparity ($+3 \dots +9$) for all input data. By inverting the partial symbol, a symbol with negative disparity ($-3 \dots -9$) is obtained.

This allows the disparity ($-2, -1, 0, 1, 2$) of the sub-symbols of the other encoders
15 (11B12B and 7B8B) to be compensated for. The encoder 6B8B generates symbols whose disparity is always 0. Then all (partial) parities of the encoders (11B12B and 7B8B) are added together and the result controls the decision as to how many inverted and non-inverted symbols of the encoder (11B13B) are used.

The minimum (smallest) disparity of the encoder is 11B13B ± 3 . In total, a disparity
20 of ± 12 ($4 \times \pm 3$) per symbol can be compensated for with these four encoders.

Furthermore, five (11B12B) encoders and one (7B8B) encoder are used whose maximum disparity is ± 2 . So in the extreme case, these six encoders generate a disparity of exactly ± 12 ($2 \times \pm 6$). This can be safely compensated by the 11B13B encoder.

25 According to one aspect of the present invention, the method achieves the same quality as an 8B10B code but with half the overhead (loss due to encoding).

The implementation of the encoding and decoding hardware requires only minimal resources (logic) due to the use of several small encoders instead of one large one.

Coding can typically take place completely in one cycle of the parallel data path (no pipelining necessary).

- 5 The control of the disparity of the 128 bit symbol can be realised with (very) little logic, and can be realised completely within one clock of the data path (slow), instead of calculating the disparity by counting the one and zero bits in the serial data stream with the very fast serial clock.

- 10 Due to the deterministic disparity and run length, further scrambling is not necessary and therefore fast synchronisation to the data stream on the receiver side is possible (no scrambler synchronisation required).

Among other things, this is very useful for power save modes in which the link can be switched off to save energy and switched on again when required. Fast synchronisation between the transmitter and receiver is a must for this.

- 15 According to one aspect of the present invention, the invention encodes a 112-bit wide data word, with any disparity (disparity max: 112) and any run length (run length max: 112), on a 128-bit wide symbol. Thus, the overhead resulting from the encoding is 14.2%.

- 20 The maximum run length occurring in the symbol, as well as with any sequencing of any symbols, is 8 identical bits.

The maximum disparity in the long mean is 0. The disparity in a symbol is less than 9.

The complexity of the logic is minimal, comparable with 10 8B/10B encoders (with the well-known disadvantage of the large overhead).

This is achieved by the use, or parallel use, of several "small" encoders that are optimally matched to each other in terms of their disparity and run length characteristics.

5 The encoders 11B12B, 7B8B and 6B8B generate all symbols with a guaranteed maximum run length of 6, even with any sequencing of the (partial) symbols.

According to one aspect of the present invention, the encoder (11B13B) generates symbols with a guaranteed maximum run length of 7 or at the beginning or end of the symbol of 5. Due to the sequencing (Fig2) of (11B13B) with the other encoders, a maximum run length of 8 can thus be generated in the symbol.

10 See properties of the encoders as follows:

To further reduce the complexity of the hardware, four small encoders (11B13B) are used whose disparity can be controlled in terms of sign (+-).

15 According to one aspect of the present invention, the symbol disparity is controlled in such a way that each encoder calculates the parity of "its" subsymbol. This is done with little effort, since the subsymbol has only a few bits.

20 With four of the eleven encoders, the sign of the disparity of the subsymbol can be actively controlled by inverting the generated subsymbol. For this purpose, the encoders (11B13B) have the special feature that their symbols generate a symbol with positive disparity (+3... +9) for all input data. By inverting the partial symbol, a symbol with negative disparity (-3...-9) is obtained.

25 This allows the disparity (-2,-1,0,1,2) of the sub-symbols of the other encoders (11B12B and 7B8B) to be compensated for. The encoder 6B8B generates symbols whose disparity is always 0. Then all (partial) parities of the encoders (11B12B and 7B8B) are added together and the result controls the decision as to how many inverted and non-inverted symbols of the encoder (11B13B) are used.

The minimum (smallest) disparity of the encoder is 11B13B ± 3 . In total, a disparity of ± 12 ($4 \times \pm 3$) per symbol can be compensated for with these four encoders.

Furthermore, five (11B12B) encoders and one (7B8B) encoder are used whose maximum disparity is ± 2 . So in the extreme case, these six encoders generate a
 5 disparity of exactly ± 12 ($2 \times \pm 6$). This can be safely compensated by the 11B13B encoder.

The process achieves the same quality as an 8B10B code but with half the overhead (loss due to encoding).

The implementation of the encoding and decoding hardware requires only minimal
 10 resources (logic) due to the use of several small encoders instead of one large one.

Coding can typically take place completely in one cycle of the parallel data path (no pipelining necessary).

The control of the disparity of the 128 bit symbol can be realised with (very) little logic, and can be realised completely within one clock of the data path (slow),
 15 instead of calculating the disparity by counting the one and zero bits in the serial data stream with the very fast serial clock.

Due to the deterministic disparity and run length, further scrambling is not necessary and therefore fast synchronisation to the data stream is possible on the receiver side (no scrambler synchronisation required).

20 Among other things, this is very useful for power save modes in which the link can be switched off to save energy and switched on again when required. Fast synchronisation between the transmitter and receiver is a must for this.

Figure 5 shows a data format that can be used in Figures 2, 3 and/or 4, for example. Again, the 128-bit total symbol and the arbitrary data sequence of 112 bits are
 25 shown. Overall, the arbitrary bit sequence and the total symbol can have different

header data or frame data.

In the following, some concrete possibilities are created as to how segments of the arbitrary bit sequence can be translated into sub-symbols so that the disparity is minimised or eliminated. A first table shows a translation from 6 bits to 8 bits, a
 5 second table shows a translation from 7 bits to 8 bits, a third table shows a translation from 11 bits to 12 bits and a fourth table shows a translation from 11 bits to 13 bits. Segments of 6, 7 or 11 bits are thus converted into sub-symbols of 8, 12 or 13 bits. The encodings shown are exemplary and illustrate the technical effect that is achieved in the present case. The present invention has been empirically
 10 evaluated and, using the proposed encoding, achieves that 112 bits can be optimised with regard to their disparity in such a way that only 128 bits are required. This corresponds to a so-called overhead of only 14%.

In the present case, cells are used as a synonym for frames. These can also be packages.

15 Cell format / Frame format

According to one aspect of the present invention, the cell comprises a header with a fixed bit length, a payload area with 4 selectable bit lengths and a footer, again with a fixed bit length.

The cell structure is a sequence of bits as follows:

- 20 - A 7-bit virtual path identifier (VP) that represents a unique address of the virtual path.
- A 3-bit sequence number (SN) that numbers the cells consecutively in their sequence.
- A 2-bit wide cell type (CT) identifier that specifies the length of the user data.

- A 3-bit wide payload information (PI) that contains additional information about the payload. This can also be used to synchronise payload data and frame data or control data.
- A 10-bit wide CRC polynomial (HCRC) for error protection of the header information. The polynomial has a Hamming distance of 5 up to a bit sequence of 21 bits ($P=0x2B9$).
- The payload (PL) range has a length of: 187, 411, 635 or 859 bits, depending on the CT value. The shortest payload is selected so that it is still larger than the largest supported (video) streaming bus width. (Should simplify the mapping of streaming data to the cell payload).
- Finally, a 12-bit wide CRC polynomial (PCRC) for error protection of the user data. The polynomial has a Hamming distance of 4 up to a bit sequence of 2035 bits ($P=0x8F3$).

Format of the transfer frames

- According to one aspect of the present invention, the transmission frame comprises a sequence of M-bit wide words. The frame starts with an M-bit wide "comma" word from a defined sequence of comma words for the frame alignment. This is followed by K cells. The cells consist of 2, 4, 6 or 8 N-bit wide words that carry the header, payload and footer. These N-bit wide words are coded into M-bit wide symbols (line coding).

This format is selected to enable cell data to be processed at appropriate time frequencies, provided that the serialiser/deserialiser always processes a block of M bits.

Figure 6A shows a section of an exemplary coding of data segments

- to symbols, where 6 bits are coded to 8 bits in such a way that the disparity

according to

is optimised in one aspect of the present invention. For example, a

Segment 000000 coded to a partial symbol 00101011, i.e. a 6B8B encoder.

Furthermore, Figure 6B shows a 7B8B encoder, Figure 6C shows an 11B12B
5 encoder and Figure 6C shows an 11B12B encoder.

Figure 6D an 11B13B encoder.

Claims

1. A method in an automobile for generating an efficiently transferable bit sequence with a restricted disparity and a restricted run length, comprising
 - a provision (100) of any bit sequence;
 - 5 - segmenting (101) the provided bit sequence into a predefined sequence of segments according to a predefined bit length;
 - encoding (102) each segment into a respective sub-symbol, using a respective encoding unit per segment from a plurality of encoding units, wherein a first subset of encoding units actively controls a sign of the disparity of the sub-symbol by inverting
 - 10 the disparity of the generated sub-symbol to compensate for a disparity of a second subset of encoding units, wherein a juxtaposition of the sub-symbols results in the efficiently transmittable bit sequence.
2. The method according to claim 1, characterised in that the arbitrary bit sequence is unrestricted in its disparity and run length.
- 15 3. The method according to claim 1 or 2, characterised in that the disparity of the second subset of coding units is not controllable.
4. The method according to one of the preceding claims, characterised in that the active control of the sign is carried out in each case by means of conditional bitwise inversion of the symbol.
- 20 5. The method according to one of the preceding claims, characterised in that the conditional inversion takes place as a function of the disparity of an overall symbol, which is formed from all partial symbols.

6. The method according to one of the preceding claims, characterised in that the dependence is influenced in such a way that a magnitude value of the disparity is minimised.
7. The method according to one of the preceding claims, characterised in that the absolute value is minimised in such a way that a positive overall symbol disparity is counteracted by a negative parity of the sub-symbols.
8. The method according to one of the preceding claims, characterised in that the absolute value is minimised in such a way that, in the case of negative overall symbol disparity, it is counteracted by a positive parity of the sub-symbols.
9. The method according to one of the preceding claims, characterised in that the active control takes place as a function of total symbols already transmitted, in such a way that the disparity of all total symbols is minimised.
10. The method according to one of the preceding claims, characterised in that encoding units of the first subset encode segments of 11 bits to subsymbols of 13 bits.
11. The method according to one of the preceding claims, characterised in that coding units of the first subset have a disparity between +3 and +9, which are specifically inverted to -3 to -9 by bitwise inversion of the partial symbol.
12. The method according to one of the preceding claims, characterised in that the run length in partial symbols is a maximum of 7.
13. The method according to one of the preceding claims, characterised in that the run length of the subsymbol for coding units of the first subset is a maximum of 5 starting from the most significant and/or the least significant bit.
14. The method according to one of the preceding claims, characterised in that, in the case of coding units of the second subset, 11 bit segments are coded to 12 bit

sub-symbols or 7 bit segments are coded to 8 bit sub-symbols or 6 bit segments are coded to 8 bit sub-symbols.

15. The method according to one of the preceding claims, characterised in that a disparity between -2 and +2 is generated for coding units of the second subset.

5 16. The method according to one of the preceding claims, characterised in that for coding units of the second subset, the run length in the generated partial symbol is 6.

17. The method according to one of the preceding claims, characterised in that coding units of the second subset generate partial symbols which have a maximum run length of 3 at the edge.

10 18. The method according to one of the preceding claims, characterised in that the coding units are addressed in parallel and each code a segment into a partial symbol.

15 19. The method according to one of the preceding claims, characterised in that the coding units are addressed in the sequence 21212221212, wherein a 1 stands for a coding unit of the first subset and a 2 stands for a coding unit of the second subset.

20. The method according to one of the preceding claims, characterised in that a multiplexer selects the data stream which contributes to minimising the overall disparity of the overall symbol when a positive data stream and a negative data stream are present with respect to the disparity.

20 21. A system arrangement in an automobile for generating an efficiently transferable bit sequence with a limited disparity and a limited run length, comprising

- an interface unit set up to provide (100) any bit sequence;

- a segmentation unit set up for segmenting (101) the bit sequence provided into a predefined sequence of segments according to a predefined bit length in each case;

- an encoding arrangement arranged for encoding (102) each segment into a respective subsymbol, using a respective encoding unit per segment from a plurality of encoding units, wherein a first subset of encoding units actively controls a sign of the disparity of the subsymbol by inverting the disparity of the generated subsymbol
- 5 to compensate for a disparity of a second subset of encoding units, wherein a juxtaposition of the subsymbols results in the efficiently transmittable bit sequence.

22. A computer program product comprising instructions which, when the program is executed by at least one computer, cause the computer to perform the steps of the method according to any one of claims 1 to 20.

- 10 23. A computer-readable storage medium comprising instructions which, when executed by at least one computer, cause the computer to perform the steps of the method according to any one of claims 1 to 20.