WIKIPEDIA
The Free Encyclopedia

WIKIPEDIA

8b/10b encoding

In telecommunications, **8b/10b** is a line code that maps 8-bit words to 10-bit symbols to achieve DC balance and bounded disparity, and at the same time provide enough state changes to allow reasonable clock recovery. This means that the difference between the counts of ones and zeros in a string of *at least* 20 bits is no more than two, and that there are not more than five ones or zeros in a row. This helps to reduce the demand for the lower bandwidth limit of the channel necessary to transfer the signal.[1]

Fibre Channel
Layer 4. Protocol mapping
LUN masking
Layer 3. Common services
Layer 2. Network
Fibre Channel fabric Fibre Channel zoning Registered state change notification
Layer 1. Data link
Fibre Channel 8b/10b encoding
Layer 0. Physical

An 8b/10b code can be implemented in various ways with focus on different performance parameters. One implementation was designed by K. Odaka for the DAT digital audio recorder.[2] Kees Schouhamer Immink designed an 8b/10b code for the DCC audio recorder.[3] The IBM implementation was described in 1983 by Al Widmer and Peter Franaszek.[4][5]

IBM implementation

As the scheme name suggests, eight bits of data are transmitted as a 10-bit entity called a *symbol*, or *character*. The low five bits of data are encoded into a 6-bit group (the 5b/6b portion) and the top three bits are encoded into a 4-bit group (the 3b/4b portion). These code groups are concatenated together to form the 10-bit symbol that is transmitted on the wire. The *data symbols* are often referred to as D.x.y where x ranges over 0–31 and y over 0–7. Standards using the 8b/10b encoding also define up to 12 *special symbols* (or *control characters*) that can be sent in place of a *data symbol*. They are often used to indicate start-of-frame, end-of-frame, link idle, skip and similar link-level conditions. At least one of them (i.e. a "comma" symbol) needs to be used to define the alignment of the 10-bit symbols. They are referred to as K.x.y and have different encodings from any of the D.x.y symbols.

Because 8b/10b encoding uses 10-bit symbols to encode 8-bit words, some of the possible 1024 (10 bit, 2^{10}) symbols can be excluded to grant a run-length limit of 5 consecutive equal bits and to ensure the difference between the count of zeros and ones to be no more than two. Some of the 256 possible 8-bit words can be encoded in two different ways. Using these alternative encodings, the scheme is

IPR2024-001429
HP – EX1020, Page 1 of 8

able to achieve long-term DC-balance in the serial data stream. This permits the data stream to be transmitted through a channel with a high-pass characteristic, for example Ethernet's transformer-coupled unshielded twisted pair or optical receivers using automatic gain control.

Encoding tables and byte encoding

Note that in the following tables, for each input byte (represented as `HGF EDCBA`), *A* denotes the least significant bit (LSB), and *H* the most significant (MSB). The output gains two extra bits, **i** and **j**. The bits are sent from LSB to MSB: a, b, c, d, e, **i**, f, g, h, and **j**; i.e., the 5b/6b code followed by the 3b/4b code. This ensures the uniqueness of the special bit sequence in the comma symbols.

The residual effect on the stream to the number of zero and one bits transmitted is maintained as the *running disparity* (*RD*) and the effect of slew is balanced by the choice of encoding for following symbols.

The 5b/6b code is a paired disparity code, and so is the 3b/4b code. Each 6- or 4-bit code word has either equal numbers of zeros and ones (a disparity of zero), or comes in a pair of forms, one with two more zeros than ones (four zeros and two ones, or three zeros and one one, respectively) and one with two less. When a 6- or 4-bit code is used that has a non-zero disparity (count of ones minus count of zeros; i.e., −2 or +2), the choice of positive or negative disparity encodings must be the one that toggles the running disparity. In other words, the non zero disparity codes alternate.

Running disparity

8b/10b coding is DC-free, meaning that the long-term ratio of ones and zeros transmitted is exactly 50%. To achieve this, the difference between the number of ones transmitted and the number of zeros transmitted is always limited to ±2, and at the end of each symbol, it is either +1 or −1. This difference is known as the *running disparity* (RD).

This scheme needs only two states for the running disparity of +1 and −1. It starts at −1.[6]

For each 5b/6b and 3b/4b code with an unequal number of ones and zeros, there are two bit patterns that can be used to transmit it: one with two more "1" bits, and one with all bits inverted and thus two more zeros. Depending on the current running disparity of the signal, the encoding engine selects which of the two possible six- or four-bit sequences to send for the given data. Obviously, if the six-bit or four-bit code has equal numbers of ones and zeros, there is no choice to make, as the disparity would be unchanged, with the exceptions of sub-blocks D.07 (00111) and D.x.3 (011). In either case the disparity is still unchanged, but if RD is positive when D.07 is encountered 000111 is used, and if it is negative 111000 is used. Likewise, if RD is positive when D.x.3 is encountered 0011 is used, and if it is negative 1100 is used. This is accurately reflected in the charts below, but is worth making additional mention of as these are the only two sub-blocks with equal numbers of 1s and 0s that each have two possible encodings.

IPR2024-001429
HP – EX1020, Page 2 of 8

Rules for running disparity

previous RD	Disparity of code word	Disparity chosen	next RD
−1	0	0	−1
−1	±2	+2	+1
+1	0	0	+1
+1	±2	−2	−1

5b/6b code (abcdei)

Input		RD = −1	RD = +1
Code	**EDCBA**	**a b c d e i**	
D.00	00000	100111	011000
D.01	00001	011101	100010
D.02	00010	101101	010010
D.03	00011	110001	
D.04	00100	110101	001010
D.05	00101	101001	
D.06	00110	011001	
D.07	00111	111000	000111
D.08	01000	111001	000110
D.09	01001	100101	
D.10	01010	010101	
D.11	01011	110100	
D.12	01100	001101	
D.13	01101	101100	
D.14	01110	011100	
D.15	01111	010111	101000
not used		**1111**00	**0000**11

Input		RD = −1	RD = +1	
Code	**EDCBA**	**a b c d e i**		
D.16	10000	011011	100100	
D.17	10001	100011		
D.18	10010	010011		
D.19	10011	110010		
D.20	10100	001011		
D.21	10101	101010		
D.22	10110	011010		
D.23 †	10111	111010	000101	*also used for the K.23.7 symbol*
D.24	11000	110011	001100	
D.25	11001	100110		
D.26	11010	010110		
D.27 †	11011	110110	001001	*also used for the K.27.7 symbol*
D.28	11100	001110		
D.29 †	11101	101110	010001	*also used for the K.29.7 symbol*
D.30 †	11110	011110	100001	*also used for the K.30.7 symbol*
D.31	11111	101011	010100	
K.28 ‡	11100	00**1111**	11**0000**	exclusively used for K.28.x symbols

† also used for the 5b/6b code of K.x.7

IPR2024-001429
HP – EX1020, Page 3 of 8

‡ exclusively used for the 5b/6b code of K.28.y

3b/4b code (fghj)

Input		RD = −1	RD = +1
Code	HGF	f g h j	
D.x.0	000	1011	0100
D.x.1	001	1001	
D.x.2	010	0101	
D.x.3	011	1100	0011
D.x.4	100	1101	0010
D.x.5	101	1010	
D.x.6	110	0110	
D.x.P7 †	111	1110	0001
D.x.A7 †		0111	1000

Input		RD = −1	RD = +1
Code	HGF	f g h j	
K.x.0	000	1011	0100
K.x.1 ‡	001	**0**110	**1**001
K.x.2	010	1010	0101
K.x.3	011	1100	0011
K.x.4	100	1101	0010
K.x.5 ‡	101	**0**101	**1**010
K.x.6	110	1001	0110
K.x.7 ‡	111	**0**111	**1**000

† For D.x.7, either the Primary (D.x.P7), or the Alternate (D.x.A7) encoding must be selected in order to avoid a run of five consecutive 0s or 1s when combined with the preceding 5b/6b code.
Sequences of exactly five identical bits are used in comma symbols for synchronization issues.
D.x.A7 is used only

- when RD = −1: for x = 17, 18 and 20 and
- when RD = +1: for x = 11, 13 and 14.

With x = 23, x = 27, x = 29, and x = 30, the 3b/4b code portion used for control symbols K.x.7 is the same as that for D.x.A7.
Any other D.x.A7 code can't be used as it would result in chances for misaligned comma sequences.

‡ Only K.28.1, K.28.5, and K.28.7 generate comma symbols, that contain a bit sequence of five 0s or 1s.
The symbol has the format 11**0000 0**1xx or 00**1111 1**0xx.

Control symbols

The control symbols within 8b/10b are 10b symbols that are valid sequences of bits (no more than six 1s or 0s) but do not have a corresponding 8b data byte. They are used for low-level control functions. For instance, in Fibre Channel, K28.5 is used at the beginning of four-byte sequences (called "Ordered Sets") that perform functions such as Loop Arbitration, Fill Words, Link Resets, etc.

Resulting from the 5b/6b and 3b/4b tables the following 12 control symbols are allowed to be sent:

IPR2024-001429
HP – EX1020, Page 4 of 8

Control symbols

Input				RD = −1	RD = +1
Symbol	**DEC**	**HEX**	**HGF EDCBA**	**abcdei fghj**	**abcdei fghj**
K.28.0	28	1C	000 11100	001111 0100	110000 1011
K.28.1 †	60	3C	001 11100	00**1111 1**001	11**0000 0**110
K.28.2	92	5C	010 11100	001111 0101	110000 1010
K.28.3	124	7C	011 11100	001111 0011	110000 1100
K.28.4	156	9C	100 11100	001111 0010	110000 1101
K.28.5 †	188	BC	101 11100	00**1111 1**010	11**0000 0**101
K.28.6	220	DC	110 11100	001111 0110	110000 1001
K.28.7 ‡	252	FC	111 11100	00**1111 1**000	11**0000 0**111
K.23.7	247	F7	111 10111	111010 1000	000101 0111
K.27.7	251	FB	111 11011	110110 1000	001001 0111
K.29.7	253	FD	111 11101	101110 1000	010001 0111
K.30.7	254	FE	111 11110	011110 1000	100001 0111

† Within the control symbols, K.28.1, K.28.5, and K.28.7 are "comma symbols". Comma symbols are used for synchronization (finding the alignment of the 8b/10b codes within a bit-stream). If K.28.7 is not used, the unique comma sequences 00**111110** or 11**000001** cannot inadvertently appear at any bit position within any combination of normal codes.

‡ If K.28.7 is allowed in the actual coding, a more complex definition of the synchronization pattern than suggested by † needs to be used, as a combination of K.28.7 with several other codes forms a false misaligned comma symbol overlapping the two codes. A sequence of multiple K.28.7 codes is not allowable in any case, as this would result in undetectable misaligned comma symbols.

K.28.7 is the only comma symbol that cannot be the result of a single bit error in the data stream.

Example encoding of D31.1

D31.1 for both running disparity cases

Input				RD = −1	RD = +1
Code	**DEC**	**HEX**	**HGF EDCBA**	**abcdei fghj**	**abcdei fghj**
D31.1	63	3F	001 11111	101011 1001	010100 1001

Technologies that use 8b/10b

After the above-mentioned IBM patent expired, the scheme became even more popular and was chosen as a DC-free line code for several communication technologies.

Among the areas in which 8b/10b encoding finds application are the following:

IPR2024-001429
HP – EX1020, Page 5 of 8

- Aurora
- Camera Serial Interface
- CoaXPress
- Common Public Radio Interface (CPRI)
- DVB Asynchronous serial interface (ASI)
- DVI and HDMI Video Island (transition-minimized differential signaling)
- DisplayPort 1.x
- ESCON (Enterprise Systems Connection)
- Fibre Channel
- Gigabit Ethernet (except for the twisted pair–based 1000BASE-T)
- IEEE 1394b (FireWire and others)
- InfiniBand
- JESD204B
- OBSAI RP3 interface
- PCI Express 1.x and 2.x
- Serial RapidIO
- SD UHS-II
- Serial ATA
- SAS 1.x, 2.x and 3.x
- SSA
- ServerNet (starting with ServerNet2)
- SGMII
- UniPro M-PHY[7]
- USB 3.0
- Thunderbolt 1.x and 2.x
- XAUI
- SLVS-EC

Fibre Channel (4GFC and 8GFC variants only)

The FC-0 standard defines what encoding scheme is to be used (8b/10b or 64b/66b) in a Fibre Channel system[8] – higher speed variants typically use 64b/66b to optimize bandwidth efficiency (since bandwidth overhead is 20% in 8b/10b versus approximately 3% (~ 2/66) in 64b/66b systems). Thus, 8b/10b encoding is used for 4GFC and 8GFC variants; for 10GFC and 16GFC variants, it is 64b/66b.[9] The Fibre Channel *FC1* data link layer is then responsible for implementing the 8b/10b encoding and decoding of signals.

The Fibre Channel 8b/10b coding scheme is also used in other telecommunications systems. Data is expanded using an algorithm that creates one of two possible 10-bit output values for each input 8-bit value. Each 8-bit input value can map either to a 10-bit output value with odd disparity, or to one with even disparity. This mapping is usually done at the time when parallel input data is converted into a serial output stream for transmission over a fibre channel link. The odd/even selection is done in such a way that a long-term zero disparity between ones and zeroes is maintained. This is often called "DC balancing".

IPR2024-001429
HP – EX1020, Page 6 of 8

The 8-bit to 10-bit conversion scheme uses only 512 of the possible 1024 output values. Of the remaining 512 unused output values, most contain either too many ones (or too many zeroes) and therefore are not allowed. This still leaves enough spare 10-bit odd+even coding pairs to allow for at least 12 special non-data characters.

The codes that represent the 256 data values are called the data (D) codes. The codes that represent the 12 special non-data characters are called the control (K) codes.

All of the codes can be described by stating 3 octal values. This is done with a naming convention of "Dxx.x" or "Kxx.x". (Note that the tables in earlier sections are using decimal, rather than octal, values for Dxx.x or Kxx.x)

Example:

Input Data Bits: ABCDEFGH
Data is split: ABC DEFGH
Data is shuffled: DEFGH ABC

Now these bits are converted to decimal in the way they are paired.

Input data

```
C3 (HEX) = 11000011
         = 110 00011
         = 00011 110
         =   3    6
```

E 8B/10B = D03.6

Digital audio

Encoding schemes 8b/10b have found a heavy use in digital audio storage applications, namely

- Digital Audio Tape, US Patent 4,456,905, June 1984 by K. Odaka.
- Digital Compact Cassette (DCC), US Patent 4,620,311, October 1986 by Kees Schouhamer Immink.

A differing but related scheme is used for audio CDs and CD-ROMs:

- Compact disc Eight-to-fourteen modulation

Alternatives

Note that 8b/10b is the encoding scheme, not a specific code. While many applications do use the same code, there exist some incompatible implementations; for example, Transition Minimized Differential Signaling, which also expands 8 bits to 10 bits, but it uses a completely different method to do so.

IPR2024-001429
HP – EX1020, Page 7 of 8

64b/66b encoding, introduced for 10 Gigabit Ethernet's 10GBASE-R Physical Medium Dependent (PMD) interfaces, is a lower-overhead alternative to 8b/10b encoding, having a two-bit overhead per 64 bits (instead of eight bits) of encoded data. This scheme is considerably different in design from 8b/10b encoding, and does not explicitly guarantee DC balance, short run length, and transition density (these features are achieved statistically via scrambling). 64b/66b encoding has been extended to the 128b/130b and 128b/132b encoding variants for PCI Express 3.0 and USB 3.1, respectively, replacing the 8b/10b encoding in earlier revisions of each standard.[10]

References

1. Kees Schouhamer Immink (March 1997). "Performance Assessment of DC-Free Multimode Codes". *IEEE Transactions on Communications*. **45** (3): 293–299. doi:10.1109/26.558690 (https://doi.org/10.1109%2F26.558690). "The *dc-balanced* or *dc-free* codes, as they are often called, have a long history and their application is certainly not confined to recording practice."
2. US 4,456,905 (https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US4,456,905), "Method and apparatus for encoding binary data", published 1984-06-26
3. US 4,620,311 (https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US4,620,311), "Method of transmitting information, encoding device for use in the method, and decoding device for use in the method", published 1986-10-28
4. Al X. Widmer, Peter A. Franaszek (1983). "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code" (http://domino.research.ibm.com/tchjr/journalindex.nsf/0/b4e28be4a69a153585256bfa0067f59a?OpenDocument). *IBM Journal of Research and Development*. **27** (5): 440–451. doi:10.1147/rd.275.0440 (https://doi.org/10.1147%2Frd.275.0440).
5. US 4,486,739 (https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US4,486,739), "Byte oriented DC balanced (0,4) 8B/10B partitioned block transmission code", published 1984-12-04
6. Thatcher, Jonathan (April 1, 1996). "Thoughts on Gigabit Ethernet Physical" (https://grouper.ieee.org/groups/802/3/z/public/presentations/mar1996/JTtgep.txt). *IBM*. Retrieved August 17, 2008.
7. "Physical Layer Specifications" (http://www.mipi.org/specifications/physical-layer#MPHY). *Mipi.org*. MIPI Alliance. Retrieved April 20, 2014.
8. Fibre Channel Framing and Signaling - 3 (FC-FS-3) Rev 1.1 Sections 5.2.1 and 5.3.1 [1] (http://www.t11.org/index.html)
9. FIBRE CHANNEL Physical Interface-5 (FC-PI-5) REV 6.10 Section 5.7 [2] (http://www.t11.org/index.html)
10. Mahesh Wagh (August 6, 2011). "PCIe 3.0 Encoding & PHY Logical" (https://www.pcisig.com/developers/main/training_materials/get_document?doc_id=9ee1d6372dbafcfd4b20f4f6508a05200d817846) (PDF). *pcisig.com*. Retrieved June 5, 2015.

External links

- "Definition of 8b/10b encoding" (https://web.archive.org/web/20140608160037/http://www.knowledgetransfer.net/dictionary/Storage/en/8b10b_encoding.htm). Archived from the original (http://www.knowledgetransfer.net/dictionary/Storage/en/8b10b_encoding.htm) on June 8, 2014. Retrieved September 2, 2014.

Retrieved from "https://en.wikipedia.org/w/index.php?title=8b/10b_encoding&oldid=1242619524"

IPR2024-001429
HP – EX1020, Page 8 of 8