WIKIPEDIA

Gray code

The **reflected binary code** (**RBC**), also known just as **reflected binary** (**RB**) or **Gray code** after Frank Gray, is an ordering of the binary numeral system such that two successive values differ in only one bit (binary digit).

For example, the representation of the decimal value "1" in binary would normally be "001" and "2" would be "010". In Gray code, these values are represented as "001" and "011". That way, incrementing a value from 1 to 2 requires only one bit to change, instead of two.

Gray codes are widely used to prevent spurious output from electromechanical switches and to facilitate error correction in digital communications such as digital terrestrial television and some cable TV systems.

	Lucal code[1][2]				
	5	**4**	**3**	**2**	**1**
	Gray code				
	4	**3**	**2**	**1**	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	1	0
3	0	0	1	0	1
4	0	1	1	0	0
5	0	1	1	1	1
6	0	1	0	1	0
7	0	1	0	0	1
8	1	1	0	0	0
9	1	1	0	1	1
10	1	1	1	1	0
11	1	1	1	0	1
12	1	0	1	0	0
13	1	0	1	1	1
14	1	0	0	1	0
15	1	0	0	0	1

Contents

Motivation and name

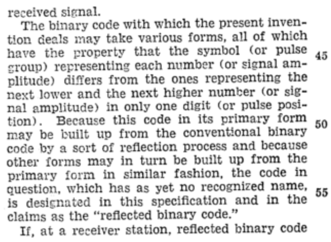
Many devices indicate position by closing and opening switches. If that device uses natural binary codes, positions 3 and 4 are next to each other but all three bits of the binary representation differ:

Decimal	Binary
...	...
3	011
4	100
...	...

The problem with natural binary codes is that physical switches are not ideal: it is very unlikely that physical switches will change states exactly in synchrony. In the transition between the two states shown above, all three switches change state. In the brief period while all are changing, the switches will read some spurious position. Even without keybounce, the transition might look like 011 — 001 — 101 — 100. When the switches appear to be in position 001, the observer cannot tell if that is the "real" position 001, or a transitional state between two other positions. If the output feeds into a sequential system, possibly via combinational logic, then the sequential system may store a false value.

This problem can be solved by changing only one switch at a time, so there is never any ambiguity of position, resulting in codes assigning to each of a contiguous set of integers, or to each member of a circular list, a word of symbols such that no two code words are identical and each two adjacent code words differ by exactly one symbol. These codes are also known as *unit-distance*,[3][4][5][6][7] *single-distance*, *single-step*, *monostrophic*[8][9][6][7] or *syncopic codes*,[8] in reference to the Hamming distance of 1 between adjacent codes.

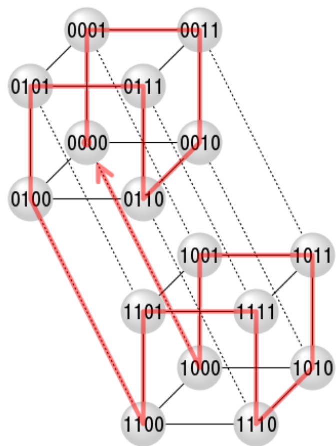
In principle, there can be more than one such code for a given word length, but the term Gray code was first applied to a particular binary code for non-negative integers, the *binary-reflected Gray code*, or **BRGC**. Bell Labs researcher George R. Stibitz described such a code in a 1941 patent application, granted in 1943.[10][11][12] Frank Gray introduced the term *reflected binary code* in his 1947 patent application, remarking that the code had "as yet no recognized name".[13] He derived the name from the fact that it "may be built up from the conventional binary code by a sort of reflection process".


received signal.
The binary code with which the present invention deals may take various forms, all of which have the property that the symbol (or pulse group) representing each number (or signal amplitude) differs from the ones representing the next lower and the next higher number (or signal amplitude) in only one digit (or pulse position). Because this code in its primary form may be built up from the conventional binary code by a sort of reflection process and because other forms may in turn be built up from the primary form in similar fashion, the code in question, which has as yet no recognized name, is designated in this specification and in the claims as the "reflected binary code."
If, at a receiver station, reflected binary code

Gray's patent introduces the term "reflected binary code"

In the standard encoding the least significant bit follows a repetitive pattern of 2 on, 2 off (... 11001100 ...); the next digit a pattern of 4 on, 4 off; the n-th least significant bit a pattern of $\mathbf{2}^n$ on $\mathbf{2}^n$ off. The four-bit version of this is shown below:

Decimal	Binary	Gray	Gray Decimal
0	0000	0000	0
1	0001	0001	1
2	0010	0011	3
3	0011	0010	2
4	0100	0110	6
5	0101	0111	7
6	0110	0101	5
7	0111	0100	4
8	1000	1100	12
9	1001	1101	13
10	1010	1111	15
11	1011	1110	14
12	1100	1010	10
13	1101	1011	11
14	1110	1001	9
15	1111	1000	8



Visualized as a traversal of vertices of a tesseract

For decimal 15 the code rolls over to decimal 0 with only one switch change. This is called the cyclic or adjacency property of the code.[14]

In modern digital communications, Gray codes play an important role in error correction. For example, in a digital modulation scheme such as QAM where data is typically transmitted in symbols of 4 bits or more, the signal's constellation diagram is arranged so that the bit patterns conveyed by adjacent constellation points differ by only one bit. By combining this with forward error correction capable of correcting single-bit errors, it is possible for a receiver to correct any transmission errors that cause a constellation point to deviate into the area of an adjacent point. This makes the transmission system less susceptible to noise.

Despite the fact that Stibitz described this code[10][11][12] before Gray, the reflected binary code was later named after Gray by others who used it. Two different 1953 patent applications use "Gray code" as an alternative name for the "reflected binary code";[15][16] one of those also lists "minimum error code" and "cyclic permutation code" among the names.[16] A 1954 patent application refers to "the Bell Telephone Gray code".[17] Other names include "cyclic binary code",[11] "cyclic progression code",[18][11] "cyclic permuting binary"[19] or "cyclic permuted binary" (CPB).[20][21]

The Gray code was sometimes attributed, incorrectly,[12] to Elisha Gray.[22][23][24]

History and practical application

Mathematical puzzles

Reflected binary codes were applied to mathematical puzzles before they became known to engineers.

The binary-reflected Gray code represents the underlying scheme of the classical Chinese rings puzzle, a sequential mechanical puzzle mechanism described by the French Louis Gros in 1872.[25][12]

It can serve as a solution guide for the Towers of Hanoi problem, based on a game by the French Édouard Lucas in 1883.[26][27][28][29] Similarly, the so called Towers of Bucharest and Towers of Klagenfurt game configurations yield ternary and pentary Gray codes.[30]

Martin Gardner wrote a popular account of the Gray code in his August 1972 Mathematical Games column in Scientific American.[31]

The code also forms a Hamiltonian cycle on a hypercube, where each bit is seen as one dimension.

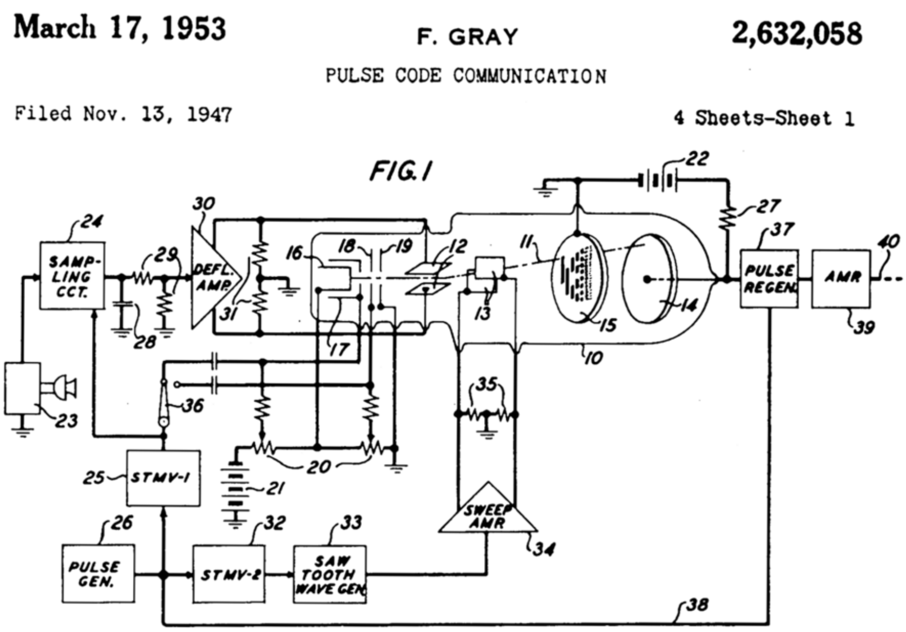
Telegraphy codes

When the French engineer Émile Baudot changed from using a 6-unit (6-bit) code to 5-unit code for his printing telegraph system, in 1875[32] or 1876,[33][34] he ordered the alphabetic characters on his print wheel using a reflected binary code, and assigned the codes using only three of the bits to vowels. With vowels and consonants sorted in their alphabetical order,[35][36][37] and other symbols appropriately placed, the 5-bit character code has been recognized as a reflected binary code.[12] This code became known as Baudot code[38] and, with minor changes, was eventually adopted as International Telegraph Alphabet No. 1 (ITA1, CCITT-1) in 1932.[39][40][37]

About the same time, the German-Austrian Otto Schäffler[41] demonstrated another printing telegraph in Vienna using a 5-bit reflected binary code for the same purpose, in 1874.[42][12]

Analog-to-digital signal conversion

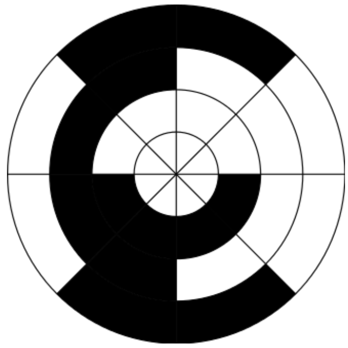
Frank Gray, who became famous for inventing the signaling method that came to be used for compatible color television, invented a method to convert analog signals to reflected binary code groups using vacuum tube-based apparatus. Filed in 1947, the method and apparatus were granted a patent in 1953,[13] and the name of Gray stuck to the codes. The "PCM tube" apparatus that Gray patented was made by Raymond W. Sears of Bell Labs, working with Gray and William M. Goodall, who credited Gray for the idea of the reflected binary code.[43]



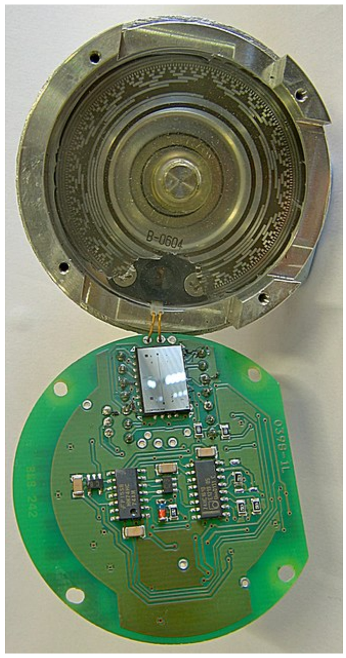
Part of front page of Gray's patent, showing PCM tube (10) with reflected binary code in plate (15)

Gray was most interested in using the codes to minimize errors in converting analog signals to digital; his codes are still used today for this purpose.

Position encoders



Rotary encoder for angle-measuring devices marked in 3-bit binary-reflected Gray code (BRGC)



A Gray code absolute rotary encoder with 13 tracks. Housing, interrupter disk, and light source are in the top; sensing element and support components are in the bottom.

Gray codes are used in linear and rotary position encoders (absolute encoders and quadrature encoders) in preference to weighted binary encoding. This avoids the possibility that, when multiple bits change in the binary representation of a position, a misread will result from some of the bits changing before others.

For example, some rotary encoders provide a disk which has an electrically conductive Gray code pattern on concentric rings (tracks). Each track has a stationary metal spring contact that provides electrical contact to the conductive code pattern. Together, these contacts produce output signals in the form of a Gray code. Other encoders employ non-contact mechanisms based on optical or magnetic sensors to produce the Gray code output signals.

Regardless of the mechanism or precision of a moving encoder, position measurement error can occur at specific positions (at code boundaries) because the code may be changing at the exact moment it is read (sampled). A binary output code could cause significant position measurement errors because it is impossible to make all bits change at exactly the same time. If, at the moment the position is sampled, some bits have changed and others have not, the sampled position will be incorrect. In the case of absolute encoders, the indicated position may be far away from the actual position and, in the case of incremental encoders, this can corrupt position tracking.

In contrast, the Gray code used by position encoders ensures that the codes for any two consecutive positions will differ by only one bit and, consequently, only one bit can change at a time. In this case, the maximum position error will be small, indicating a position adjacent to the actual position.

Genetic algorithms

Due to the Hamming distance properties of Gray codes, they are sometimes used in genetic algorithms.[14] They are very useful in this field, since mutations in the code allow for mostly incremental changes, but occasionally a single bit-change can cause a big leap and lead to new properties.

Boolean circuit minimization

Gray codes are also used in labelling the axes of Karnaugh maps since 1953[44][45][46] as well as in Händler circle graphs since 1958,[47][48][49][50] both graphical methods for logic circuit minimization.

Error correction

In modern digital communications, Gray codes play an important role in error correction. For example, in a digital modulation scheme such as QAM where data is typically transmitted in symbols of 4 bits or more, the signal's constellation diagram is arranged so that the bit patterns conveyed by adjacent constellation points differ by only one bit. By combining this with forward error correction capable of correcting single-bit errors, it is possible for a receiver to correct any transmission errors that cause a constellation point to deviate into the area of an adjacent point. This makes the transmission system less susceptible to noise.

Communication between clock domains

Digital logic designers use Gray codes extensively for passing multi-bit count information between synchronous logic that operates at different clock frequencies. The logic is considered operating in different "clock domains". It is fundamental to the design of large chips that operate with many different clocking frequencies.

Cycling through states with minimal effort

If a system has to cycle through all possible combinations of on-off states of some set of controls, and the changes of the controls require non-trivial expense (e.g. time, wear, human work), a Gray code minimizes the number of setting changes to just one change for each combination of states. An example would be testing a piping system for all combinations of settings of its manually operated valves.

A balanced Gray code can be constructed,[51] that flips every bit equally often. Since bit-flips are evenly distributed, this is optimal in the following way: balanced Gray codes minimize the maximal count of bit-flips for each digit.

Gray code counters and arithmetic

George R. Stibitz utilized a reflected binary code in a binary pulse counting device in 1941 already.[10][11][12]

A typical use of Gray code counters is building a FIFO (first-in, first-out) data buffer that has read and write ports that exist in different clock domains. The input and output counters inside such a dual-port FIFO are often stored using Gray code to prevent invalid transient states from being captured when the count crosses clock domains.[52] The updated read and write pointers need to be passed between clock domains when they change, to be able to track FIFO empty and full status in each domain. Each bit of the pointers is sampled non-deterministically for this clock domain transfer. So for each bit, either the old value or the new value is propagated. Therefore, if more than one bit in the multi-bit pointer is changing at the sampling point, a "wrong" binary value (neither new nor old) can be propagated. By guaranteeing only one bit can be changing, Gray codes guarantee that the only possible sampled values are the new or old multi-bit value. Typically Gray codes of power-of-two length are used.

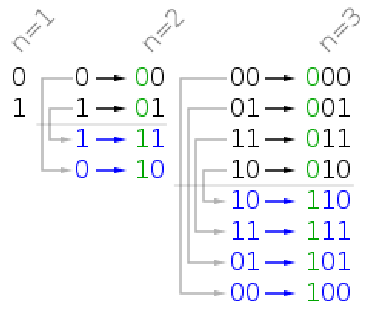
Sometimes digital buses in electronic systems are used to convey quantities that can only increase or decrease by one at a time, for example the output of an event counter which is being passed between clock domains or to a digital-to-analog converter. The advantage of Gray codes in these applications is that differences in the propagation delays of the many wires that represent the bits of the code cannot cause the received value to go through states that are out of the Gray code sequence. This is similar to the advantage of Gray codes in the construction of mechanical encoders, however the source of the Gray code is an electronic counter in this case. The counter itself must count in Gray code, or if the counter runs in binary then the output value from the counter must be reclocked after it has been converted to Gray code, because when a value is converted from binary to Gray code,[nb 1] it is possible that differences in the arrival times of the binary data bits into the binary-to-Gray conversion circuit will mean that the code could go briefly through states that are wildly out of sequence. Adding a clocked register after the circuit that converts the count value to Gray code may introduce a clock cycle of latency, so counting directly in Gray code may be advantageous.[53]

To produce the next count value in a Gray-code counter, it is necessary to have some combinational logic that will increment the current count value that is stored. One way to increment a Gray code number is to convert it into ordinary binary code,[54] add one to it with a standard binary adder, and then convert the result back to Gray code.[55] Other methods of counting in Gray code are discussed in a report by Robert W. Doran, including taking the output from the first latches of the master-slave flip flops in a binary ripple counter.[56]

Gray code addressing

As the execution of program code typically causes an instruction memory access pattern of locally consecutive addresses, bus encodings using Gray code addressing instead of binary addressing can reduce the number of state changes of the address bits significantly, thereby reducing the CPU power consumption in some low-power designs.[57][58]

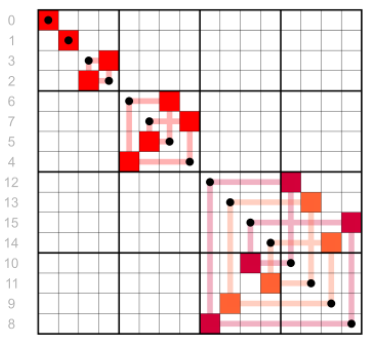
Constructing an *n*-bit Gray code



The first few steps of the reflect-and-prefix method.

The binary-reflected Gray code list for n bits can be generated recursively from the list for $n - 1$ bits by reflecting the list (i.e. listing the entries in reverse order), prefixing the entries in the original list with a binary 0, prefixing the entries in the reflected list with a binary 1, and then concatenating the original list with the reversed list.[12] For example, generating the $n = 3$ list from the $n = 2$ list:

2-bit list:	00, 01, 11, 10	
Reflected:		10, 11, 01, 00
Prefix old entries with *0*:	000, 001, 011, 010,	
Prefix new entries with *1*:		110, 111, 101, 100
Concatenated:	000, 001, 011, 010,	110, 111, 101, 100



4-bit Gray code permutation

The one-bit Gray code is $G_1 = (0, 1)$. This can be thought of as built recursively as above from a zero-bit Gray code $G_0 = (\,\Lambda\,)$ consisting of a single entry of zero length. This iterative process of generating G_{n+1} from G_n makes the following properties of the standard reflecting code clear:

- G_n is a permutation of the numbers 0, …, $2^n - 1$. (Each number appears exactly once in the list.)
- G_n is embedded as the first half of G_{n+1}.
- Therefore, the coding is *stable*, in the sense that once a binary number appears in G_n it appears in the same position in all longer lists; so it makes sense to talk about *the* reflective Gray code value of a number: $G(m)$ = the m-th reflecting Gray code, counting from 0.
- Each entry in G_n differs by only one bit from the previous entry. (The Hamming distance is 1.)
- The last entry in G_n differs by only one bit from the first entry. (The code is cyclic.)

These characteristics suggest a simple and fast method of translating a binary value into the corresponding Gray code. Each bit is inverted if the next higher bit of the input value is set to one. This can be performed in parallel by a bit-shift and exclusive-or operation if they are available: the nth Gray code is obtained by computing $n \oplus \lfloor n/2 \rfloor$. Prepending a 0 leaves the order of the code words unchanged, prepending a 1 reverses the order of the code words. If the bits at position i of codewords are inverted, the order of neighbouring blocks of 2^i codewords is reversed. E.g. if bit 0 is inverted in a 3 bit codeword sequence, the order of two neighbouring codewords is reversed

{000,001,010,011,100,101,110,111} → {001,000,011,010,101,100,111,110} (invert bit 0)

If bit 1 is inverted, blocks of 2 codewords change order:

{000,001,010,011,100,101,110,111} → {010,011,000,001,110,111,100,101} (invert bit 1)

If bit 2 is inverted, blocks of 4 codewords reverse order:

{000,001,010,011,100,101,110,111} → {100,101,110,111,000,001,010,011} (invert bit 2)

Thus, ex-oring a bit b_i at position i with the bit b_{i+1} at position $i+1$ leaves the order of codewords intact if $b_{i+1} = 0$, and reverses the order of blocks of 2^{i+1} codewords if $b_{i+1} = 1$. Now, this is exactly the same operation as the reflect-and-prefix method to generate the Gray code.

A similar method can be used to perform the reverse translation, but the computation of each bit depends on the computed value of the next higher bit so it cannot be performed in parallel. Assuming g_i is the ith Gray-coded bit (g_0 being the most significant bit), and b_i is the ith binary-coded bit (b_0 being the most-significant bit), the reverse translation can be given recursively: $b_0 = g_0$, and $b_i = g_i \oplus b_{i-1}$. Alternatively, decoding a Gray code into a binary number can be described as a prefix sum of the bits in the Gray code, where each individual summation operation in the prefix sum is performed modulo two.

To construct the binary-reflected Gray code iteratively, at step 0 start with the $\mathbf{code}_0 = 0$, and at step $i > 0$ find the bit position of the least significant 1 in the binary representation of i and flip the bit at that position in the previous code $\mathbf{code}_{i-1}$ to get the next code $\mathbf{code}_i$. The bit positions start 0, 1, 0, 2, 0, 1, 0, 3, ...[nb 2] See find first set for efficient algorithms to compute these values.

Converting to and from Gray code

The following functions in C convert between binary numbers and their associated Gray codes. While it may seem that Gray-to-binary conversion requires each bit to be handled one at a time, faster algorithms exist.[59][54][nb 1]

```
typedef unsigned int uint;

// This function converts an unsigned binary number to reflected binary Gray code.
uint BinaryToGray(uint num)
{
    return num ^ (num >> 1); // The operator >> is shift right. The operator ^ is exclusive or.
}

// This function converts a reflected binary Gray code number to a binary number.
uint GrayToBinary(uint num)
{
    uint mask = num;
    while (mask) {           // Each Gray code bit is exclusive-ored with all more significant bits.
        mask >>= 1;
        num   ^= mask;
    }
    return num;
}

// A more efficient version for Gray codes 32 bits or fewer through the use of SWAR (SIMD within a register) techniques.
// It implements a parallel prefix XOR function. The assignment statements can be in any order.
//
// This function can be adapted for longer Gray codes by adding steps.

uint GrayToBinary32(uint num)
{
    num ^= num >> 16;
    num ^= num >>  8;
    num ^= num >>  4;
    num ^= num >>  2;
    num ^= num >>  1;
    return num;
}
// A Four-bit-at-once variant changes a binary number (abcd)2 to (abcd)2 ^ (00ab)2, then to (abcd)2 ^ (00ab)2 ^ (0abc)2 ^ (000a)2.
```

Special types of Gray codes

In practice, "Gray code" almost always refers to a binary-reflected Gray code (BRGC). However, mathematicians have discovered other kinds of Gray codes. Like BRGCs, each consists of a list of words, where each word differs from the next in only one digit (each word has a Hamming distance of 1 from the next word).

n-ary Gray code

There are many specialized types of Gray codes other than the binary-reflected Gray code. One such type of Gray code is the ***n*-ary Gray code**, also known as a **non-Boolean Gray code**. As the name implies, this type of Gray code uses non-Boolean values in its encodings.

For example, a 3-ary (ternary) Gray code would use the values {0, 1, 2}.[30] The (n, k)-*Gray code* is the n-ary Gray code with k digits.[60] The sequence of elements in the (3, 2)-Gray code is: {00, 01, 02, 12, 11, 10, 20, 21, 22}. The (n, k)-Gray code may be constructed recursively, as the BRGC, or may be constructed iteratively. An algorithm to iteratively generate the (N, k)-Gray code is presented (in C):

```
// inputs: base, digits, value
// output: Gray
// Convert a value to a Gray code with the given base and digits.
// Iterating through a sequence of values would result in a sequence
// of Gray codes in which only one digit changes at a time.
void toGray(unsigned base, unsigned digits, unsigned value, unsigned gray[digits])
{
    unsigned baseN[digits]; // Stores the ordinary base-N number, one digit per entry
    unsigned i;     // The loop variable

    // Put the normal baseN number into the baseN array. For base 10, 109
    // would be stored as [9,0,1]
    for (i = 0; i < digits; i++) {
        baseN[i] = value % base;
        value    = value / base;
    }

    // Convert the normal baseN number into the Gray code equivalent. Note that
    // the loop starts at the most significant digit and goes down.
    unsigned shift = 0;
    while (i--) {
        // The Gray digit gets shifted down by the sum of the higher
        // digits.
        gray[i] = (baseN[i] + shift) % base;
        shift = shift + base - gray[i]; // Subtract from base so shift is positive
    }
}
// EXAMPLES
// input: value = 1899, base = 10, digits = 4
// output: baseN[] = [9,9,8,1], gray[] = [0,1,7,1]
// input: value = 1900, base = 10, digits = 4
// output: baseN[] = [0,0,9,1], gray[] = [0,1,8,1]
```

Ternary number → ternary Gray code

```
  0 → 000
  1 → 001
  2 → 002
 10 → 012
 11 → 011
 12 → 010
 20 → 020
 21 → 021
 22 → 022
100 → 122
101 → 121
102 → 120
110 → 110
111 → 111
112 → 112
120 → 102
121 → 101
122 → 100
200 → 200
201 → 201
202 → 202
210 → 212
211 → 211
212 → 210
220 → 220
221 → 221
222 → 222
```

There are other Gray code algorithms for (n,k)-Gray codes. The (n,k)-Gray code produced by the above algorithm is always cyclical; some algorithms, such as that by Guan,[60] lack this property when k is odd. On the other hand, while only one digit at a time changes with this method, it can change by wrapping (looping from $n - 1$ to 0). In Guan's algorithm, the count alternately rises and falls, so that the numeric difference between two Gray code digits is always one.

Gray codes are not uniquely defined, because a permutation of the columns of such a code is a Gray code too. The above procedure produces a code in which the lower the significance of a digit, the more often it changes, making it similar to normal counting methods.

See also Skew binary number system, a variant ternary number system where at most 2 digits change on each increment, as each increment can be done with at most one digit carry operation.

Balanced Gray code

Although the binary reflected Gray code is useful in many scenarios, it is not optimal in certain cases because of a lack of "uniformity".[51] In **balanced Gray codes**, the number of changes in different coordinate positions are as close as possible. To make this more precise, let G be an R-ary complete Gray cycle having transition sequence (δ_k); the *transition counts* (*spectrum*) of G are the collection of integers defined by

$$\lambda_k = |\{j \in \mathbb{Z}_{R^n} : \delta_j = k\}| \,, \text{ for } k \in \mathbb{Z}_n$$

A Gray code is *uniform* or *uniformly balanced* if its transition counts are all equal, in which case we have $\lambda_k = R^n/n$ for all k. Clearly, when $R = 2$, such codes exist only if n is a power of 2. Otherwise, if n does not divide R^n evenly, it is possible to construct *well-balanced* codes where every transition count is either $\lfloor R^n/n \rfloor$ or $\lceil R^n/n \rceil$.[51] Gray codes can also be *exponentially balanced* if all of their transition counts are adjacent powers of two, and such codes exist for every power of two.[61]

For example, a balanced 4-bit Gray code has 16 transitions, which can be evenly distributed among all four positions (four transitions per position), making it uniformly balanced:[51]

```
0 1 1 1 1 1 1 0 0 0 0 0 0 1 1 0
0 0 1 1 1 1 0 0 1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 1 0 0 1 1 1 0 0
0 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1
```

whereas a balanced 5-bit Gray code has a total of 32 transitions, which cannot be evenly distributed among the positions. In this example, four positions have six transitions each, and one has eight:[51]

```
1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 0 0 0
1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1
1 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1
1 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1
```

We will now show a construction[62] and implementation[63] for well-balanced binary Gray codes which allows us to generate an n-digit balanced Gray code for every n. The main principle is to inductively construct an $(n + 2)$-digit Gray code G' given an n-digit Gray code G in such a way that the balanced property is preserved. To do this, we consider partitions of $G = g_0, \ldots, g_{2^n-1}$ into an even number L of non-empty blocks of the form

$$\{g_0\}, \{g_1, \ldots, g_{k_2}\}, \{g_{k_2+1}, \ldots, g_{k_3}\}, \ldots, \{g_{k_{L-2}+1}, \ldots, g_{-2}\}, \{g_{-1}\}$$

where $k_1 = 0, k_{L-1} = -2$, and $k_L = -1 \pmod{2^n})$. This partition induces an $(n+2)$-digit Gray code given by

$$\begin{aligned}
&00g_0, \\
&00g_1, \ldots, 00g_{k_2}, 01g_{k_2}, \ldots, 01g_1, 11g_1, \ldots, 11g_{k_2}, \\
&11g_{k_2+1}, \ldots, 11g_{k_3}, 01g_{k_3}, \ldots, 01g_{k_2+1}, 00g_{k_2+1}, \ldots, 00g_{k_3}, \ldots, \\
&00g_{-2}, 00g_{-1}, 10g_{-1}, 10g_{-2}, \ldots, 10g_0, 11g_0, 11g_{-1}, 01g_{-1}, 01g_0
\end{aligned}$$

If we define the *transition multiplicities* $m_i = |\{j : \delta_{k_j} = i, 1 \leq j \leq L\}|$ to be the number of times the digit in position i changes between consecutive blocks in a partition, then for the $(n + 2)$-digit Gray code induced by this partition the transition spectrum λ'_i is

$$\lambda'_i = \begin{cases} 4\lambda_i - 2m_i, & \text{if } 0 \leq i < n \\ L, & \text{otherwise} \end{cases}$$

The delicate part of this construction is to find an adequate partitioning of a balanced n-digit Gray code such that the code induced by it remains balanced, but for this only the transition multiplicities matter; joining two consecutive blocks over a digit i transition and splitting another block at another digit i transition produces a different Gray code with exactly the same transition spectrum λ'_i, so one may for example[61] designate the first m_i transitions at digit i as those that fall between two blocks. Uniform codes can be found when $R \equiv 0 \pmod{4}$ and $R^n \equiv 0 \pmod{n}$, and this construction can be extended to the R-ary case as well.[62]

Long Run Gray codes

Long Run Gray codes maximize the distance between consecutive changes of digits in the same position. That is, the minimum run-length of any bit remains unchanged for as long as possible.

Monotonic Gray codes

Monotonic codes are useful in the theory of interconnection networks, especially for minimizing dilation for linear arrays of processors.[64] If we define the *weight* of a binary string to be the number of 1s in the string, then although we clearly cannot have a Gray code with strictly increasing weight, we may want to approximate this by having the code run through two adjacent weights before reaching the next one.

We can formalize the concept of monotone Gray codes as follows: consider the partition of the hypercube $Q_n = (V_n, E_n)$ into *levels* of vertices that have equal weight, i.e.

$$V_n(i) = \{v \in V_n : v \text{ has weight } i\}$$

for $0 \leq i \leq n$. These levels satisfy $|V_n(i)| = \binom{n}{i}$. Let $Q_n(i)$ be the subgraph of Q_n induced by $V_n(i) \cup V_n(i+1)$, and let $E_n(i)$ be the edges in $Q_n(i)$. A monotonic Gray code is then a Hamiltonian path in Q_n such that whenever $\delta_1 \in E_n(i)$ comes before $\delta_2 \in E_n(j)$ in the path, then $i \leq j$.

An elegant construction of monotonic n-digit Gray codes for any n is based on the idea of recursively building subpaths $P_{n,j}$ of length $2\binom{n}{j}$ having edges in $E_n(j)$.[64] We define $P_{1,0} = (0, 1)$, $P_{n,j} = \emptyset$ whenever $j < 0$ or $j \geq n$, and

$$P_{n+1,j} = 1P^{\pi_n}_{n,j-1}, 0P_{n,j}$$

otherwise. Here, π_n is a suitably defined permutation and P^π refers to the path P with its coordinates permuted by π. These paths give rise to two monotonic n-digit Gray codes $G^{(1)}_n$ and $G^{(2)}_n$ given by

$$G^{(1)}_n = P_{n,0} P^R_{n,1} P_{n,2} P^R_{n,3} \cdots \text{ and } G^{(2)}_n = P^R_{n,0} P_{n,1} P^R_{n,2} P_{n,3} \cdots$$

The choice of π_n which ensures that these codes are indeed Gray codes turns out to be $\pi_n = E^{-1}(\pi_{n-1}^2)$. The first few values of $P_{n,j}$ are shown in the table below.

These monotonic Gray codes can be efficiently implemented in such a way that each subsequent element can be generated in $O(n)$ time. The algorithm is most easily described using coroutines.

Subpaths in the Savage–Winkler algorithm

$P_{n,j}$	j = 0	j = 1	j = 2	j = 3
n = 1	0, 1			
n = 2	00, 01	10, 11		
n = 3	000, 001	100, 110, 010, 011	101, 111	
n = 4	0000, 0001	1000, 1100, 0100, 0110, 0010, 0011	1010, 1011, 1001, 1101, 0101, 0111	1110, 1111

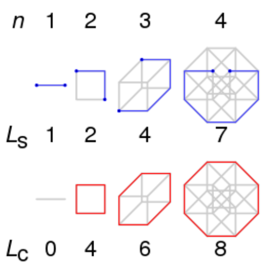
Monotonic codes have an interesting connection to the Lovász conjecture, which states that every connected vertex-transitive graph contains a Hamiltonian path. The "middle-level" subgraph $Q_{2n+1}(n)$ is vertex-transitive (that is, its automorphism group is transitive, so that each vertex has the same "local environment"" and cannot be differentiated from the others, since we can relabel the coordinates as well as the binary digits to obtain an automorphism) and the problem of finding a Hamiltonian path in this subgraph is called the "middle-levels problem", which can provide insights into the more general conjecture. The question has been answered affirmatively for $n \leq 15$, and the preceding construction for monotonic codes ensures a Hamiltonian path of length at least 0.839N where N is the number of vertices in the middle-level subgraph.[65]

Beckett–Gray code

Another type of Gray code, the **Beckett–Gray code**, is named for Irish playwright Samuel Beckett, who was interested in symmetry. His play "Quad" features four actors and is divided into sixteen time periods. Each period ends with one of the four actors entering or leaving the stage. The play begins with an empty stage, and Beckett wanted each subset of actors to appear on stage exactly once.[66] Clearly the set of actors currently on stage can be represented by a 4-bit binary Gray code. Beckett, however, placed an additional restriction on the script: he wished the actors to enter and exit so that the actor who had been on stage the longest would always be the one to exit. The actors could then be represented by a first in, first out queue, so that (of the actors onstage) the actor being dequeued is always the one who was enqueued first.[66] Beckett was unable to find a Beckett–Gray code for his play, and indeed, an exhaustive listing of all possible sequences reveals that no such code exists for n = 4. It is known today that such codes do exist for n = 2, 5, 6, 7, and 8, and do not exist for n = 3 or 4. An example of an 8-bit Beckett–Gray code can be found in Donald Knuth's *Art of Computer Programming*.[12] According to Sawada and Wong, the search space for n = 6 can be explored in 15 hours, and more than 9,500 solutions for the case n = 7 have been found.[67]

Snake-in-the-box codes

Snake-in-the-box codes, or *snakes*, are the sequences of nodes of induced paths in an n-dimensional hypercube graph, and coil-in-the-box codes,[68] or *coils*, are the sequences of nodes of induced cycles in a hypercube. Viewed as Gray codes, these sequences have the property of being able to detect any single-bit coding error. Codes of this type were first described by William H. Kautz in the late 1950s;[4] since then, there has been much research on finding the code with the largest possible number of codewords for a given hypercube dimension.



Maximum lengths of snakes (L_s) and coils (L_c) in the snakes-in-the-box problem for dimensions n from 1 to 4

Single-track Gray code

Yet another kind of Gray code is the **single-track Gray code** (STGC) developed by Norman B. Spedding[69][70] and refined by Hiltgen, Paterson and Brandestini in "Single-track Gray codes" (1996).[71][72] The STGC is a cyclical list of P unique binary encodings of length n such that two consecutive words differ in exactly one position, and when the list is examined as a $P \times n$ matrix, each column is a cyclic shift of the first column.[73]

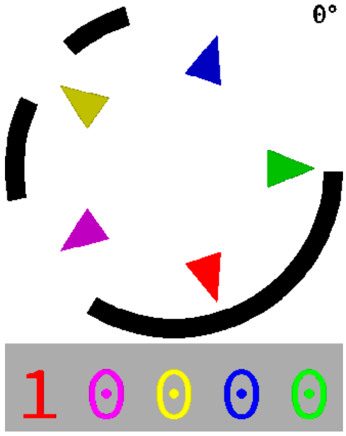
The name comes from their use with rotary encoders, where a number of tracks are being sensed by contacts, resulting for each in an output of 0 or 1. To reduce noise due to different contacts not switching at exactly the same moment in time, one preferably sets up the tracks so that the data output by the contacts are in Gray code. To get high angular accuracy, one needs lots of contacts; in order to achieve at least 1 degree accuracy, one needs at least 360 distinct positions per revolution, which requires a minimum of 9 bits of data, and thus the same number of contacts.

Single-track Gray code with 5 sensors.

If all contacts are placed at the same angular position, then 9 tracks are needed to get a standard BRGC with at least 1 degree accuracy. However, if the manufacturer moves a contact to a different angular position (but at the same distance from the center shaft), then the corresponding "ring pattern" needs to be rotated the same angle to give the same output. If the most significant bit (the inner ring in Figure 1) is rotated enough, it exactly matches the next ring out. Since both rings are then identical, the inner ring can be cut out, and the sensor for that ring moved to the remaining, identical ring (but offset at that angle from the other sensor on that ring). Those two sensors on a single ring make a quadrature encoder. That reduces the number of tracks for a "1 degree resolution" angular encoder to 8 tracks. Reducing the number of tracks still further can't be done with BRGC.

For many years, Torsten Sillke[74] and other mathematicians believed that it was impossible to encode position on a single track such that consecutive positions differed at only a single sensor, except for the 2-sensor, 1-track quadrature encoder. So for applications where 8 tracks were too bulky, people used single-track incremental encoders (quadrature encoders) or 2-track "quadrature encoder + reference notch" encoders.

Norman B. Spedding, however, registered a patent in 1994 with several examples showing that it was possible.[69] Although it is not possible to distinguish 2^n positions with n sensors on a single track, it *is* possible to distinguish close to that many. Etzion and Paterson conjecture that when n is itself a power of 2, n sensors can distinguish at most $2^n - 2n$ positions and that for prime n the limit is $2^n - 2$ positions.[75] The authors went on to generate a 504 position single track code of length 9 which they believe is optimal. Since this number is larger than $2^8 = 256$, more than 8 sensors are required by any code, although a BRGC could distinguish 512 positions with 9 sensors.



Animated and color-coded version of the STGC rotor.

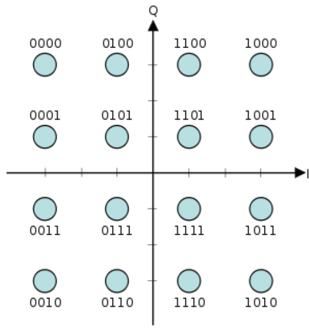
An STGC for $P = 30$ and $n = 5$ is reproduced here:

Single-track Gray code for 30 positions

Angle	Code	Angle	Code	Angle	Code	Angle	Code	Angle	Code
0°	10000	72°	01000	144°	00100	216°	00010	288°	00001
12°	10100	84°	01010	156°	00101	228°	10010	300°	01001
24°	11100	96°	01110	168°	00111	240°	10011	312°	11001
36°	11110	108°	01111	180°	10111	252°	11011	324°	11101
48°	11010	120°	01101	192°	10110	264°	01011	336°	10101
60°	11000	132°	01100	204°	00110	276°	00011	348°	10001

Each column is a cyclic shift of the first column, and from any row to the next row only one bit changes.[76] The single-track nature (like a code chain) is useful in the fabrication of these wheels (compared to BRGC), as only one track is needed, thus reducing their cost and size. The Gray code nature is useful (compared to chain codes, also called De Bruijn sequences), as only one sensor will change at any one time, so the uncertainty during a transition between two discrete states will only be plus or minus one unit of angular measurement the device is capable of resolving.[77]

Two-dimensional Gray code



A Gray-coded constellation diagram for rectangular 16-QAM

Two-dimensional Gray codes are used in communication to minimize the number of bit errors in quadrature amplitude modulation (QAM) adjacent points in the constellation. In a typical encoding the horizontal and vertical adjacent constellation points differ by a single bit, and diagonal adjacent points differ by 2 bits.[78]

Two-dimensional Gray codes also have uses in location identifications schemes, where the code would be applied to area maps such as a Mercator projection of the earth's surface and an appropriate cyclic two-dimensional distance function such as the Mannheim metric be used to calculate the distance between two encoded locations, thereby combining the characteristics of the Hamming distance with the cyclic continuation of a Mercator projection.[79]

Gray isometry

The bijective mapping { 0 ↔ 00, 1 ↔ 01, 2 ↔ 11, 3 ↔ 10 } establishes an isometry between the metric space over the finite field $\mathbb{Z}_2^2$ with the metric given by the Hamming distance and the metric space over the finite ring $\mathbb{Z}_4$ (the usual modular arithmetic) with the metric given by the Lee distance. The mapping is suitably extended to an isometry of the Hamming spaces $\mathbb{Z}_2^{2m}$ and $\mathbb{Z}_4^m$. Its importance lies in establishing a correspondence between various "good" but not necessarily linear codes as Gray-map images in $\mathbb{Z}_2^2$ of ring-linear codes from $\mathbb{Z}_4$.[80][81]

Related codes

There are a number of binary codes similar to Gray codes, including:

- Datex codes aka Giannini codes (1954), as described by Carl P. Spaulding,[8][82][83][84][85][7] use a variant of O'Brien code II.
- Codes used by Varec (ca. 1954),[86][87][88][89] use a variant of O'Brien code I as well as base-12 and base-16 Gray code variants.
- Lucal code (1959)[1][2][56] aka modified reflected binary code (MRB)[1][2][nb 3]
- Gillham code (1961/1962),[83][90][7][91][92] uses a variant of Datex code and O'Brien code II.
- Leslie and Russell code (1964)[93][9][94][90]
- Royal Radar Establishment code[90]

- Hoklas code (1988)[95][96][97]

The following binary-coded decimal (BCD) codes are Gray code variants as well:

- Petherick code (1953),[18][98][99][100][54][96][nb 4] also known as Royal Aircraft Establishment (RAE) code.[101]
- O'Brien codes I and II (1955)[102][103][104][84][85][96] (An O'Brien type-I code[nb 5] was already described by Frederic A. Foss of IBM[105][106] and used by Varec in 1954. Later, it was also known as Watts code or Watts reflected decimal (WRD) code and is sometimes ambiguously referred to as reflected binary modified Gray code.[107][19][20][108][109][110][111][112][113][nb 1][nb 3] An O'Brien type-II code was already used by Datex in 1954.[nb 4])
- Excess-3 Gray code (1956)[114] (aka Gray excess-3 code,[84][85][7] Gray 3-excess code, reflex excess-3 code, excess Gray code,[96] Gray excess code, 10-excess-3 Gray code or Gray–Stibitz code), described by Frank P. Turvey Jr. of ITT.[114]
- Tompkins codes I and II (1956)[3][103][104][84][85][96]
- Glixon code (1957), sometimes ambiguously also called modified Gray code[115][54][116][117][103][104][84][85][96][nb 3][nb 5]

4-bit unit-distance BCD codes[nb 6]

Name	Bit	0	1	2	3	4	5	6	7	8	9	Weights[nb 7]	Tracks	Compl.	Cyclic	5s	Comment
Gray BCD	**4**	0	0	0	0	0	0	0	0	1	1	0..3	4 (3[nb 8])	-	(2, 4, 8, 16)	-	[103][104]
	3	0	0	0	0	1	1	1	1	1	1						
	2	0	0	1	1	1	1	0	0	0	0						
	1	0	1	1	0	0	1	1	0	0	1						
Paul	**4**	1	0	0	0	0	0	0	0	1	1	1..3	4 (3[nb 8])	-	2, 10	-	[118]
	3	0	0	0	0	1	1	1	1	1	1						
	2	0	0	1	1	1	1	0	0	0	0						
	1	1	1	1	0	0	1	1	0	0	1						
Glixon	**4**	0	0	0	0	0	0	0	0	1	1	0..3	4	-	2, 4, 8, 10	(shifted +1)	[115][103][104][116][117][nb 5]
	3	0	0	0	0	1	1	1	1	1	0						
	2	0	0	1	1	1	1	0	0	0	0						
	1	0	1	1	0	0	1	1	0	0	0						
Tompkins I	**4**	0	0	0	0	0	1	1	1	1	1	0..4	2	-	2, 4, 10	+	[3][103][104]
	3	0	0	0	0	1	1	1	1	1	0						
	2	0	0	1	1	1	1	1	0	0	0						
	1	0	1	1	0	0	0	1	1	0	0						
O'Brien I (Watts)	**4**	0	0	0	0	0	1	1	1	1	1	0..3	4	9[96][97][nb 9]	2, 4, 10	+	[102][103][104][nb 5]
	3	0	0	0	0	1	1	0	0	0	0						
	2	0	0	1	1	1	1	1	1	0	0						
	1	0	1	1	0	0	0	0	1	1	0						
Petherick (RAE)	**4**	0	0	0	0	0	1	1	1	1	1	1..3	3	9[96][97][nb 9]	2, 10	+	[18][100][nb 4]
	3	1	0	0	0	1	1	0	0	0	1						
	2	0	0	1	1	1	1	1	1	0	0						
	1	1	1	1	0	0	0	0	1	1	1						
O'Brien II	**4**	0	0	0	0	0	1	1	1	1	1	1..3	3	9[84][96][97][nb 9]	2, 10	+	[102][103][104][nb 4]
	3	0	0	0	1	1	1	1	0	0	0						
	2	0	1	1	1	0	0	1	1	1	0						
	1	1	1	0	0	0	0	0	0	1	1						
Susskind	**4**	0	0	0	0	0	1	1	1	1	1	1..4	3	9[nb 9]	2, 10	+	[5]
	3	0	0	1	1	1	1	1	1	0	0						
	2	0	1	1	1	0	0	1	1	1	0						
	1	1	1	1	0	0	0	0	1	1	1						
Klar	**4**	0	0	0	0	0	1	1	1	1	1	0..4	4 (3[nb 8])	9[nb 9]	2, 10	+	[119][120]
	3	0	0	0	1	1	1	1	0	0	0						
	2	0	0	1	1	1	1	1	1	0	0						
	1	0	1	1	1	0	0	1	1	1	0						
Tompkins II	**4**	0	0	0	0	0	1	1	1	1	1	1..3	2	9[nb 10]	2, 10	+	[3][103][104]
	3	0	0	1	1	1	1	1	0	0	0						
	2	1	1	1	0	0	0	0	0	1	1						
	1	0	1	1	1	0	0	1	1	1	0						
Excess-3 Gray	**4**	0	0	0	0	0	1	1	1	1	1	1..4	4	9[96][97][nb 9]	2, 10	+	[7][96]
	3	0	1	1	1	1	1	1	1	1	0						
	2	1	1	1	0	0	0	0	1	1	1						
	1	0	0	1	1	0	0	1	1	0	0						

See also

- Linear feedback shift register
- De Bruijn sequence – a Gray code on a larger alphabet than $\{0,1\}$.
- Steinhaus–Johnson–Trotter algorithm – an algorithm that generates Gray codes for the factorial number system
- Minimum distance code
- Prouhet–Thue–Morse sequence – related to inverse Gray code
- Ryser formula
- Hilbert curve

Notes

1. By applying a simple *inversion rule*, the Gray code and the O'Brien code I can be translated into the 8421 pure binary code and the 2421 Aiken code, respectively, to ease arithmetic operations.[C]
2. Sequence 0, 1, 0, 2, 0, 1, 0, 3, … (sequence A007814 in the OEIS).
3. There are several Gray code variants which are called "modified" of some sort: The Glixon code is sometimes called modified Gray code.[D] The Lucal code is also called modified reflected binary code (MRB).[E] The O'Brien code I or Watts code is sometimes referred to as reflected binary modified Gray code.[F]
4. By interchanging and inverting three bit rows, the O'Brien code II and the Petherick code can be transferred into each other.
5. By swapping two pairs of bit rows, individually shifting four bit rows and inverting one of them, the Glixon code and the O'Brien code I can be transferred into each other.
6. Other unit-distance BCD codes include the non-Gray code related 5-bit Libaw–Craig and the 1-2-1 code.
7. Depending on a code's target application, the Hamming weights of a code can be important properties beyond coding-theoretical considerations also for physical reasons. Under some circumstances the all-cleared and/or all-set states must be omitted (f.e. to avoid non-conductive or short-circuit conditions), it may be desirable to keep the highest used weight as low as possible (f.e. to reduce power consumption of the reader circuit) or to keep the variance of used weights small (f.e. to reduce acoustic noise or current fluctuations).
8. For Gray BCD, Paul and Klar codes, the number of necessary reading tracks can be reduced from 4 to 3 if inversion of one of the middle tracks is acceptable.
9. For O'Brien codes I and II and Petherick, Susskind, Klar as well as Excess-3 Gray codes, a 9s complement can be derived by inverting the most-significant (fourth) binary digit.
10. For Tompkins code II, a 9s complement can be derived by inverting the first three digits and swapping the two middle binary digits.

References

1. Lucal, Harold M. (December 1959). "Arithmetic Operations for Digital Computers Using a Modified Reflected Binary" (http://ieeexplore.ieee.org/document/5222057/?reload=true). *IRE Transactions on Electronic Computers*. **EC-8** (4): 449–458. doi:10.1109/TEC.1959.5222057 (https://doi.org/10.1109%2FTEC.1959.5222057). ISSN 0367-9950 (https://www.worldcat.org/issn/0367-9950). S2CID 206673385 (https://api.semanticscholar.org/CorpusID:206673385). (10 pages)
2. Sellers, Jr., Frederick F.; Hsiao, Mu-Yue; Bearnson, Leroy W. (November 1968). *Error Detecting Logic for Digital Computers* (1st ed.). New York, USA: McGraw-Hill Book Company. pp. 152–164. LCCN 68-16491 (https://lccn.loc.gov/68-16491). OCLC 439460 (https://www.worldcat.org/oclc/439460).
3. Tompkins, Howard E. (September 1956) [1956-07-16]. "Unit-Distance Binary-Decimal Codes for Two-Track Commutation" (https://dokumen.tips/documents/unit-distance-binary-decimal-codes-for-two-track-commutation.html). *IRE Transactions on Electronic Computers*. Correspondence. Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, USA. **EC-5** (3): 139. doi:10.1109/TEC.1956.5219934 (https://doi.org/10.1109%2FTEC.1956.5219934). ISSN 0367-9950 (https://www.worldcat.org/issn/0367-9950). Retrieved 2020-05-18. (1 page)
4. Kautz, William H. (June 1958). "Unit-Distance Error-Checking Codes". *IRE Transactions on Electronic Computers*. **EC-7** (2): 179–180. doi:10.1109/TEC.1958.5222529 (https://doi.org/10.1109%2FTEC.1958.5222529). ISSN 0367-9950 (https://www.worldcat.org/issn/0367-9950). S2CID 26649532 (https://api.semanticscholar.org/CorpusID:26649532). (2 pages)
5. Susskind, Alfred Kriss; Ward, John Erwin (1958-03-28) [1957, 1956]. "III.F. Unit-Distance Codes / VI.E.2. Reflected Binary Codes". Written at Cambridge, Massachusetts, USA. In Susskind, Alfred Kriss (ed.). *Notes on Analog-Digital Conversion Techniques*. Technology Books in Science and Engineering. **1** (3 ed.). New York, USA: Technology Press of the Massachusetts Institute of Technology / John Wiley & Sons, Inc. / Chapman & Hall, Ltd. pp. 3-10–3-16 [3-13–3-16], 6-65–6-60 [6-60]. (x+416+2 pages) (NB. The contents of the book was originally prepared by staff members of the Servomechanisms Laboraratory, Department of Electrical Engineering, MIT, for Special Summer Programs held in 1956 and 1957. Susskind's "reading-type code" is actually a minor variant of the code shown here with the two most significant bit rows swapped to better illustrate symmetries. Also, by swapping two bit rows and inverting one of them, the code can be transferred into the Petherick code, whereas by swapping and inverting two bit rows, the code can be transferred into the O'Brien code II.)
6. Chinal, Jean P. (January 1973). "3.3. Unit Distance Codes". Written at Paris, France. *Design Methods for Digital Systems* (https://books.google.com/books?id=8FnxCAAAQBAJ). Translated by Preston, Alan; Summer, Arthur (1st English ed.). Berlin, Germany: Akademie-Verlag / Springer-Verlag. p. 50. doi:10.1007/978-3-642-86187-1 (https://doi.org/10.1007%2F978-3-642-86187-1). ISBN 978-0-387-05871-9. License No. 202-100/542/73. Order No. 7617470(6047) ES 19 B 1 / 20 K 3. Retrieved 2020-06-21. (xviii+506 pages) (NB. The French 1967 original book was named "Techniques Booléennes et Calculateurs Arithmétiques", published by Éditions Dunod.)

7. *Military Handbook: Encoders – Shaft Angle To Digital* (http://everyspec.com/MIL-HDBK/MIL-HDBK-0200-0299/download.php?spec=MIL_HDBK_231A.1809.pdf) (PDF). United States Department of Defense. 1991-09-30. MIL-HDBK-231A. Archived (https://web.archive.org/web/20200725051128/http://everyspec.com/MIL-HDBK/MIL-HDBK-0200-0299/download.php?spec=MIL_HDBK_231A.1809.pdf) (PDF) from the original on 2020-07-25. Retrieved 2020-07-25. (NB. Supersedes MIL-HDBK-231(AS) (1970-07-01).)
8. Spaulding, Carl P. (1965-01-12) [1954-03-09]. "Digital coding and translating system" (https://patentimages.storage.googleapis.com/7f/1d/09/6a9b1fa3e67cb8/US3165731.pdf) (PDF). Monrovia, California, USA: Datex Corporation. U.S. Patent 3165731A (https://patents.google.com/patent/US3165731A). Serial No. 415058. Archived (https://web.archive.org/web/20200805101618/https://patentimages.storage.googleapis.com/7f/1d/09/6a9b1fa3e67cb8/US3165731.pdf) (PDF) from the original on 2020-08-05. Retrieved 2018-01-21. (28 pages)
9. Russell, A. (August 1964). "Some Binary Codes and a Novel Five-Channel Code" (https://books.google.com/books?id=jf4eAAAAMAAJ). *Control (Systems, Instrumentation, Data Processing, Automation, Management, incorporating Automation Progress)*. Special Features. London, UK: Morgan-Grampain (Publishers) Limited. **8** (74): 399–404. Retrieved 2020-06-22. (6 pages)
10. Stibitz, George Robert (1943-01-12) [1941-11-26]. "Binary counter" (https://patents.google.com/patent/US2307868?oq=US2307868). New York, USA: Bell Telephone Laboratories, Incorporated. U.S. Patent 2,307,868 (https://patents.google.com/patent/US2307868). Serial No. 420537. Retrieved 2020-05-24. p. 2, right column, rows 43–73: "[…] A clearer idea of the position of the balls after each pulse will be obtained if the set of balls is represented by a number having a similar number of digits, each of which may have one of two arbitrary values, for example 0 and 1. If the upper position is called 0 and the lower position […] 1, then the setting of the counter […] may be read from left to right as 0,100,000. […] Following is a translation of the number of pulses received into this form of binary notation for the first sixteen pulses as received on the first five balls […] Pulse number […] Binary notation […]" [1] (https://web.archive.org/web/20201217181617/https://patentimages.storage.googleapis.com/51/a5/54/f2f6faf23da3fb/US2307868.pdf) (4 pages)
11. Winder, C. Farrell (October 1959). "Shaft Angle Encoders Afford High Accuracy" (https://web.archive.org/web/20200928132232/https://worldradiohistory.com/Archive-Tele-Tech/50s/Electronic-Industries-1959-10.pdf) (PDF). *Electronic Industries*. Chilton Company. **18** (10): 76–80. Archived from the original (https://worldradiohistory.com/Archive-Tele-Tech/50s/Electronic-Industries-1959-10.pdf) (PDF) on 2020-09-28. Retrieved 2018-01-14. p. 78: "[…] The type of code wheel most popular in optical encoders contains a cyclic binary code pattern designed to give a cyclic sequence of "on-off" outputs. The cyclic binary code is also known as the cyclic progression code, the reflected binary code, and the Gray code. This code was originated by G. R. Stibitz, of Bell Telephone Laboratories, and was first proposed for pulse-code modulation systems by Frank Gray, also of BTL. Thus the name Gray code. The Gray or cyclic code is used mainly to eliminate the possibility of errors at code transition which could result in gross ambiguities. […]"
12. Knuth, Donald Ervin (2014-09-12). "Enumeration and Backtracking / Generating all *n*-tuples" (https://books.google.com/books?id=IkuEBAAAQBAJ). *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. **4A** (1 ed.). Addison-Wesley Professional. pp. 442–443. ISBN 978-0-13348885-2. ISBN 0-13348885-3. (912 pages)
13. Gray, Frank (1953-03-17) [1947-11-13]. *Pulse Code Communication* (https://patentimages.storage.googleapis.com/a3/d7/f2/0343f5f2c0cf50/US2632058.pdf) (PDF). New York, USA: Bell Telephone Laboratories, Incorporated. U.S. Patent 2,632,058 (https://patents.google.com/patent/US2632058). Serial No. 785697. Archived (https://web.archive.org/web/20200805094312/https://patentimages.storage.googleapis.com/a3/d7/f2/0343f5f2c0cf50/US2632058.pdf) (PDF) from the original on 2020-08-05. Retrieved 2020-08-05. (13 pages)
14. Goldberg, David Edward (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning* (1 ed.). Reading, Massachusetts, USA: Addison-Wesley.
15. Breckman, Jack (1956-01-31) [1953-12-31]. *Encoding Circuit* (https://patentimages.storage.googleapis.com/46/90/d6/2ed9fb7a77e643/US2733432.pdf) (PDF). Long Branch, New Jersey, USA: US Secretary of the Army. U.S. Patent 2,733,432 (https://patents.google.com/patent/US2733432). Serial No. 401738. Archived (https://web.archive.org/web/20200805094857/https://patentimages.storage.googleapis.com/46/90/d6/2ed9fb7a77e643/US2733432.pdf) (PDF) from the original on 2020-08-05. Retrieved 2020-08-05. (8 pages)
16. Ragland, Earl Albert; Schultheis, Jr., Harry B. (1958-02-11) [1953-10-16]. *Direction-Sensitive Binary Code Position Control System* (https://patentimages.storage.googleapis.com/ad/74/b1/2b0aa965d0c1df/US2823345.pdf) (PDF). North Hollywood, California, USA: Bendix Aviation Corporation. U.S. Patent 2,823,345 (https://patents.google.com/patent/US2823345). Serial No. 386524. Archived (https://web.archive.org/web/20200805095445/https://patentimages.storage.googleapis.com/ad/74/b1/2b0aa965d0c1df/US2823345.pdf) (PDF) from the original on 2020-08-05. Retrieved 2020-08-05. (10 pages)
17. Domeshek, Sol; Reiner, Stewart (1958-06-24) [1954-01-08]. *Automatic Rectification System* (https://patentimages.storage.googleapis.com/9d/bf/65/e676a661e1217e/US2839974.pdf) (PDF). US Secretary of the Navy. U.S. Patent 2,839,974 (https://patents.google.com/patent/US2839974). Serial No. 403085. Archived (https://web.archive.org/web/20200805100327/https://patentimages.storage.googleapis.com/9d/bf/65/e676a661e1217e/US2839974.pdf) (PDF) from the original on 2020-08-05. Retrieved 2020-08-05. (8 pages)
18. Petherick, Edward John (October 1953). *A Cyclic Progressive Binary-coded-decimal System of Representing Numbers* (Technical Note MS15). Farnborough, UK: Royal Aircraft Establishment (RAE). (4 pages) (NB. Sometimes referred to as *A Cyclic-Coded Binary-Coded-Decimal System of Representing Numbers*.)
19. Evans, David Silvester (1960). *Fundamentals of Digital Instrumentation* (https://books.google.com/books?id=gpVNAAAAYAAJ) (1 ed.). London, UK: Hilger & Watts Ltd. Retrieved 2020-05-24. (39 pages)

20. Evans, David Silvester (March 1961). "Chapter Three: Direct Reading from Coded Scales". *Digital Data: Their derivation and reduction for analysis and process control* (https://books.google.com/books?id=WOIJAAAAMAAJ) (1 ed.). London, UK: Hilger & Watts Ltd / Interscience Publishers. pp. 18–23. Retrieved 2020-05-24. p. 20–23: "[…] Decoding. […] To decode C.P.B. or W.R.D. codes, a simple inversion rule can be applied. The readings of the higher tracks determine the way in which the lower tracks are translated. The inversion rule is applied line by line for the C.P.B. and for the W.R.D. it is applied decade by decade or line by line. Starting therefore with the top or slowest changing track of the C.P.B., if the result is odd (1) the next track value has to be inverted, i.e. 0 for 1 and 1 for 0. If, however, the first track is even (0), the second track is left as read, i.e. 0 for 0 and 1 for 1. Again, if the resultant reading of the second track is odd, the third track reading is inverted and so on. When an odd is changed to an even the line below is not inverted and when an even is changed to an odd the line below is inverted. The result of applying this rule to the pattern […] is the pure binary (P.B.) pattern […] where each track or digit can be given a definite numerical value (in this instance 1, 2, 4, 8, etc.). […] Using the line-by-line inversion rule on the W.R.D. code produces [a] pattern [of 1, 2, 4, 2 code] where again the digits can be given numerical values and summed decade by decade. The summing of the digits can be very useful, for example, in a high-speed scanning system; but in a parallel decoding system […], it is usual to treat each binary quartet or decade as an entity. In other words, if the first or more significant decade is odd, the second decade is rectified or complemented by inverting the D track and so on, the result being the repeating pattern of [rectified W.R.D. code]. This is an extremely easy thing to achieve since the only change required is the inversion of the meaning of the D track or complementing digit. […]" (8+82 pages) (NB. The author does not mention Gray at all and calls the standard Gray code "Cyclic Permuted Binary Code" (C.P.B.), the book index erroneously lists it as "cyclic pure binary code".)
21. Newson, P. A. (1965). *Tables for the Binary Encoding of Angles* (https://books.google.com/books?id=aUEgAQAAIAAJ) (1 ed.). United Kingdom Atomic Energy Authority, Research Group, Atomic Energy Research Establishment, Harwell, UK: H. M. Stationery Office. Retrieved 2020-05-24. (12 pages)
22. Heath, F. G. (September 1961). "Pioneers Of Binary Coding" (https://ieeexplore.ieee.org/document/5324416). *Journal of the Institution of Electrical Engineers*. Manchester College of Science and Technology, Faculty of Technology of the University of Manchester, Manchester, UK: Institution of Engineering and Technology (IET). **7** (81): 539–541. doi:10.1049/jiee-3.1961.0300 (https://doi.org/10.1049%2Fjiee-3.1961.0300). Retrieved 2020-06-22. (3 pages)
23. Cattermole, Kenneth W. (1969). Written at Harlow, Essex, UK. *Principles of pulse code modulation* (1 ed.). London, UK / New York, USA: Iliffe Books Ltd. / American Elsevier Publishing Company, Inc. pp. 245, 434. ISBN 978-0-444-19747-4. LCCN 78-80432 (https://lccn.loc.gov/78-80432). SBN 444-19747-8. p. 245: "[…] There seems to be some confusion about the attributation of this code, because two inventors named Gray have been associated with it. When I first heard the name I took it as referring to Elisha Gray, and Heath testifies to his usage of it. Many people take it as referring to Frank Gray of Bell Telephone Laboratories, who in 1947 first proposed its use in coding tubes: his patent is listed in the bibliography. […]" (2+448+2 pages)
24. Edwards, Anthony William Fairbank (2004). *Cogwheels of the Mind: The Story of Venn Diagrams* (https://books.google.com/books?id=7_0Thy4V3JIC&pg=PA65). Baltimore, Maryland, USA: Johns Hopkins University Press. pp. 48, 50. ISBN 0-8018-7434-3. ISBN 978-0-8018-7434-5.
25. Gros, Luc-Agathon-Louis (1872). *Théorie du baguenodier par un clerc de notaire lyonnais* (https://books.google.com/books?id=EcoBJRekd-sC&pg=PP1) (in French) (1 ed.). Lyon, France: Aimé Vingtrinier. Archived (http://archive.wikiwix.com/cache/?url=https%3A%2F%2Fbooks.google.fr%2Fbooks%3Fid%3DEcoBJRekd-sC%26pg%3DPP1) from the original on 2020-12-17. Retrieved 2020-12-17. [2] (https://web.archive.org/web/20201217202507/https://books.googleusercontent.com/books/content?req=AKW5Qacg5gq7SW9L1sUUqK7LQ0yJLqaCu90GFZZYzcH7dsoifr-n9hxiB30SJ1mXq3FhDejHQ7fXY2ZdzlhFywe8pQGNTgMHjX0ANYxRohedbG1FmoPTuKibswKOO7qS2X4MUwQP0dBn-vt3dfTIBFRvTBOVoZZA1ROBXTGTMybnA4gGndl8v0qIAeKsRNUuEddqabMqMngisr8fp-iNt7NMFfwq_tPk0we0yXNJWj74AFsFr1JGOlAmdQ6B_OJVTOj33IhoYybhf8LMzfpaHusXsY2BWT2ZwVQulW0mgBaem9xvRZ58TSw)(2+16+4 pages and 4 pages foldout) (NB. This booklet was published anonymously, but is known to have been authored by Louis Gros.)
26. Lucas, Édouard (November 1883). *La tour d'Hanoï : Véritable casse tête annamite - Jeu rapporté du Tonkin par le Professeur N. Claus (de Siam) Mandarin du Collège Li Sou Stian!* (in French). Imprimerie Paul Bousrez, Tours. (NB. N. Claus de Siam is an anagram of Lucas d'Amiens, pseudonym of the author Édouard Lucas.)
27. de Parville, Henri, ed. (1883-12-27). "La tour d'Hanoï, véritable casse-tête annamite, jeu rapporté du Tonkin par le professeur N. Claus (de Siam), mandarin du collège Li-Sou-Stian. Un vrai casse-tête, en effet, mais intéressant. Nous ne saurions mieux remercier le mandarin de son aimable intention à l'égard d'un profane qu'en signalant la Tour d'Hanoï aux personnes patientes possédées par le démon du jeu" (https://gallica.bnf.fr/ark:/12148/bpt6k462461g/f2.item). *Journal des Débats Politiques et Littéraires* (Review). Revue des science (in French) (Matin ed.). Paris, France: 1–2 [2]. ark:/12148/bpt6k462461g. Archived (https://web.archive.org/web/20201218125345/https://gallica.bnf.fr/ark:/12148/bpt6k462461g/f2.item) from the original on 2020-12-18. Retrieved 2020-12-18. (1 page)
28. Allardice, Robert Edgar; Fraser, Alexander Yule, eds. (February 1883). "La Tour d'Hanoï" (https://www.cambridge.org/core/journals/proceedings-of-the-edinburgh-mathematical-society/article/la-tour-d-hanoi/082EFE016BF3313A7BAA9335A9C0BCC1#article-tab). *Proceedings of the Edinburgh Mathematical Society* (in English and French). Edinburgh Mathematical Society. **2** (5): 50–53. doi:10.1017/S0013091500037147 (https://doi.org/10.1017%2FS0013091500037147). eISSN 1464-3839 (https://www.worldcat.org/issn/1464-3839). ISSN 0013-0915 (https://www.worldcat.org/issn/0013-0915). Archived (https://web.archive.org/web/20201218111951/https://www.cambridge.org/core/journals/proceedings-of-the-edinburgh-mathematical-society/article/la-tour-d-hanoi/082EFE016BF3313A7BAA9335A9C0BCC1#article-tab) from the original on 2020-12-18. Retrieved 2020-12-18. [3] (https://web.archive.org/web/20201218112132/https://www.cambridge.org/core/services/aop-cambridge-core/content/view/082EFE016BF3313A7BAA9335A9C0BCC1/S0013091500037147a.pdf/la-tour-d-hanoi.pdf) (4 pages)
29. Lucas, Édouard (1979) [1892]. *Récréations mathématiques* (in French). **3** (Librairie Albert Blanchard reissue ed.). p. 58. (The first edition of this book was published post-humously.)
30. Herter, Felix; Rote, Günter (2018-11-14) [2018-08-09, 2017-12, 2017-08-09, 2016-04-22]. "Loopless Gray Code Enumeration and the Tower of Bucharest" (https://page.mi.fu-berlin.de/rote/Papers/pdf/Loopless+Gray+code+enumeration+and+the+Tower+of+Bucharest.pdf) (PDF). *Theoretical Computer Science*. Berlin, Germany: Institut für Informatik, Freie Universität Berlin. **748**: 40–54. arXiv:1604.06707 (https://arxiv.org/abs/1604.06707). doi:10.1016/j.tcs.2017.11.017 (https://doi.org/10.1016%2Fj.tcs.2017.11.017). ISSN 0304-3975 (https://www.worldcat.org/issn/0304-3975). S2CID 4014870 (https://api.semanticscholar.org/CorpusID:4014870). doi:10.4230/LIPIcs.FUN.2016.19 (https://doi.org/10.4230%2FLIPIcs.FUN.2016.19). Archived (https://web.archive.org/web/20201216150701/https://page.mi.fu-berlin.de/rote/Papers/pdf/Loopless+Gray+code+enumeration+and+the+Tower+of+Bucharest.pdf) (PDF) from the original on 2020-12-16. Retrieved 2020-12-16. [4] (https://web.archive.org/web/20201216230211/https://arxiv.org/pdf/1604.06707.pdf) (15/18/19/24 pages)

31. Gardner, Martin (August 1972). "The curious properties of the Gray code and how it can be used to solve puzzles". *Scientific American*. Mathematical Games. Vol. 227 no. 2. p. 106. (1 page)
32. Zeman, Johann; Fischer, Ferdinand, eds. (1877). "Einige neuere Vorschläge zur mehrfachen Telegraphie: A. Absatzweise vielfache Telegraphie" (http://dingler.culture.hu-berlin.de/article/pj226/ar226125). *Dingler's Polytechnisches Journal* (in German). Augsburg, Germany: J. G. Cotta'sche Buchhandlung. **226**: 499–507. Archived (https://web.archive.org/web/20201221163504/http://dingler.culture.hu-berlin.de/article/pj226/ar226125) from the original on 2020-12-21. Retrieved 2020-12-21. p. 499: "[…] Der um die Mitte des J[ahres] 1874 patenti[e]rte, ebenfalls dem Highton'schen verwandte Typendrucker des französischen Telegraphen-Verwaltungsbeamten Baudot wurde bei seiner 1875 patenti[e]rten Weiterentwicklung in einen fünffachen umgewandelt […]"
33. Butrica, Andrew J. (1991-06-21). "Baudot, Jean Maurice Emile". In Froehlich, Fritz E.; Kent, Allen; Hall, Carolyn M. (eds.). *The Froehlich/Kent Encyclopedia of Telecommunications: Volume 2 - Batteries to Codes-Telecommunications* (https://books.google.com/books?id=l3Rn_VefKaoC&pg=PA31&lpg=PA32). **2**. Marcel Dekker Inc. / CRC Press. pp. 31–34. ISBN 0-8247-2901-3. LCCN 90-3966 (https://lccn.loc.gov/90-3966). ISBN 978-0-8247-2901-1. Retrieved 2020-12-20. p. 31: "[…] A Baudot prototype (4 years in the making) was built in 1876. The transmitter had 5 keys similar to those of a piano. Messages were sent in a special 5-element code devised by Baudot […]"
34. Fischer, Eric N. (2000-06-20). "The Evolution of Character Codes, 1874–1968" (https://archive.org/details/enf-ascii). ark:/13960/t07x23w8s. Retrieved 2020-12-20. "[…] In 1872, [Baudot] started research toward a telegraph system that would allow multiple operators to transmit simultaneously over a single wire and, as the transmissions were received, would print them in ordinary alphabetic characters on a strip of paper. He received a patent for such a system on June 17, 1874. […] Instead of a variable delay followed by a single-unit pulse, Baudot's system used a uniform six time units to transmit each character. […] his early telegraph probably used the six-unit code […] that he attributes to Davy in an 1877 article. […] in 1876 Baudot redesigned his equipment to use a five-unit code. Punctuation and digits were still sometimes needed, though, so he adopted from Hughes the use of two special letter space and figure space characters that would cause the printer to shift between cases at the same time as it advanced the paper without printing. The five-unit code he began using at this time […] was structured to suit his keyboard […], which controlled two units of each character with switches operated by the left hand and the other three units with the right hand. […]" [5] (https://web.archive.org/web/20180919020435/http://index-of.es/Varios-2/ASCII%20The%20Evolution%20of%20Character%20Codes.pdf)[6] (https://ia800606.us.archive.org/17/items/enf-ascii/ascii.pdf)
35. Rothen, Timotheus (1884-12-25). "Le télégraphe imprimeur Baudot" (https://gallica.bnf.fr/ark:/12148/bpt6k5725454q). *Journal Télégraphique* (in French). Berne, Switzerland: Le Bureau International des Administrations Télégraphiques. VIII / #16 (12): 241–253 [249]. eISSN 2725-738X (https://www.worldcat.org/issn/2725-738X). ISSN 2223-1420 (https://www.worldcat.org/issn/2223-1420). ark:/12148/bpt6k5725454q. Retrieved 2020-12-20. [7] (https://web.archive.org/web/20201221162054/https://gallica.bnf.fr/ark:/12148/bpt6k5725454q/f9.item.zoom)
36. Pendry, Henry Walter (1920) [October 1919]. Written at London, UK. *The Baudôt Printing Telegraph System* (https://books.google.com/books/about/The_Baud%C3%B4t_Printing_Telegraph_System.html?id=CfQKAQAAIAAJ) (2 ed.). London, Bath, Melbourne, New York: Sir Isaac Pitman and Sons, Ltd. pp. 43–44. LCCN 21005277 (https://lccn.loc.gov/21005277). OCLC 778309351 (https://www.worldcat.org/oclc/778309351). OL 6633244M (https://openlibrary.org/books/OL6633244M). Retrieved 2020-12-20. (vii+184 pages) (NB. A first edition was published in 1913.)
37. MacMillan, David M. (2010-04-27) [2010-04-25, 2010-04-23]. "Codes that Don't Count - Some Printing Telegraph Codes as Products of their Technologies (With Particular Attention to the Teletypesetter)" (http://www.circuitousroot.com/artifice/telegraphy/tty/codes/index.html). *lemur.com*. Revision 3. Mineral Point, Wisconsin, USA. Archived (https://web.archive.org/web/20201218233813/http://www.circuitousroot.com/artifice/telegraphy/tty/codes/index.html) from the original on 2020-12-18. Retrieved 2020-12-20.
38. Written at Lisbon, Portugual. *Convention télégraphique internationale de Saint-Pétersbourg et Règlement et tarifs y annexés, Revision de Lisbonne, 1908 / Extraits de la publication: Documents de la Conférence télégraphique internationale de Lisbonne* (in French). Berne, Switzerland: Bureau Internationale de L'Union Télégraphique. 1909 [1908].
39. "Chapter IX. Signaux de transmission, Article 35. Signaux de transmission des alphabets télegraphiques internationaux 'nos 1 et 2, signaux d.u code Morse, de l'appareil Hughes et de l'appareil Siemens". Written at Madrid, Spain. *Règlement télégraphique annexé à la convention internationale des télécommunications - protocol finale audit règlement - Madrid, 1932* (http://search.itu.int/history/HistoryDigitalCollectionDocLibrary/4.5.43.fr.201.pdf) (PDF) (in French). Berne, Switzerland: Bureau Internationale de L'Union Télégraphique. 1933 [1932]. pp. 31–40 [33]. Archived (https://web.archive.org/web/20201221011133/http://search.itu.int/history/HistoryDigitalCollectionDocLibrary/4.5.43.fr.201.pdf) (PDF) from the original on 2020-12-21. Retrieved 2020-12-21. (1+188 pages) [8] (https://web.archive.org/web/20201221012741/https://www.itu.int/en/history/Pages/PlenipotentiaryConferences.aspx?conf=4.5)
40. "Chapter IX. Transmission Signals. Article 35. Transmission Signals of the International Telegraph Alphabets Nos. 1 and 2, Morse Code Signals and Signals of the Hughes and Siemens Instruments.". *Telegraph Regulations Annexed To The International Telecommunication Convention - Final Protocol To The Telegraph Regulations - Madrid 1932* (http://search.itu.int/history/HistoryDigitalCollectionDocLibrary/4.5.43.en.101.pdf) (PDF) (in English and French). London, UK: General Post Office / His Majesty's Stationery Office. 1933 [1932]. pp. 32–40 [34]. 43-152-2 / 18693. Archived (https://web.archive.org/web/20201221011748/http://search.itu.int/history/HistoryDigitalCollectionDocLibrary/4.5.43.en.101.pdf) (PDF) from the original on 2020-12-21. Retrieved 2020-12-21. (1+2*120+26 pages) [9] (https://web.archive.org/web/20201221012741/https://www.itu.int/en/history/Pages/PlenipotentiaryConferences.aspx?conf=4.5)
41. Zemanek, Heinrich "Heinz" Josef (1983-12-01). *Otto Schäffler (1838-1928). Pionier des Telephons, der Telegraphie und der Lochkarte sowie Erbauer der ersten Wiener Telephonzentrale*. Blätter für Technikgeschichte (in German and English). 41–43 (1979–1981) (1 ed.). Vienna, Austria: Technisches Museum für Industrie und Gewerbe, Forschungsinstitut für Technikgeschichte / Springer-Verlag. pp. 81–118. ISBN 3-21181779-4. ISSN 0067-9127 (https://www.worldcat.org/issn/0067-9127). OCLC 952698275 (https://www.worldcat.org/oclc/952698275). ISBN 978-3-21181779-7.

42. Zemanek, Heinrich "Heinz" Josef (1976-06-07). "Computer prehistory and history in central Europe" (https://www.computer.org/csdl/pds/api/csdl/proceedings/download-article/12OmNzZEAtH/pdf). Written at Vienna, Austria. *International Workshop on Managing Requirements Knowledge*. AFIPS '76: Proceedings of the June 7–10, 1976, national computer conference and exposition June 1976. **1**. New York, USA: American Federation of Information Processing Societies, Association for Computing Machinery. pp. 15–20. doi:10.1145/1499799.1499803 (https://doi.org/10.1145%2F1499799.1499803). ISBN 978-1-4503-7917-5. S2CID 14114959 (https://api.semanticscholar.org/CorpusID:14114959). doi:10.1109/AFIPS.1976.41 (https://doi.org/10.1109%2FAFIPS.1976.41). Archived (https://web.archive.org/web/20201217230013/https://www.computer.org/csdl/pds/api/csdl/proceedings/download-article/12OmNzZEAtH/pdf) from the original on 2020-12-17. Retrieved 2020-12-17. p. 17: "[…] In 1874, Schaeffler invented another printing telegraph, a quadruple system like the Baudot, but mechanically more sophisticated. The Hughes telegraph had two synchronously rotating fingers, one in the sender and one in the receiver. By a piano-like keyboard the operator selected a letter and thereby made contact with the rotating finger in the corresponding direction. Since the receiving finger was in the same direction at this moment, the receiver could print the correct letter. The Baudot and the Schaeffler printing telegraphs use a five-bit binary code. ... Schaeffler's code is a reflected binary code! What F. Gray patented in 1953 for PCM, Schaeffler had applied in his telegraph in 1874, and for a similar reason: reliability. He had contact fingers sensing on five cams consecutively all combinations; the right one triggers printing. If the fingers are to make a minimal number of movements, the solution is the reflected binary code. For Schaeffler, this idea was a minor one. More exactly, the code is described in a letter by the Austrian Post employee, J[ohann] N[epomuk] Teufelhart, inserted there as a footnote and telling that Schaeffler found the code by combining wooden bars with the different combinations until he had the best solution. Another Post employee, Alexander Wilhelm Lambert of Linz, claims to have shown this code to Schaeffler as early as 1872, but this claim is not clear and cannot be checked. […]" (6 pages)
43. Goodall, William M. (January 1951). "Television by Pulse Code Modulation". *Bell System Technical Journal*. **30** (1): 33–49. doi:10.1002/j.1538-7305.1951.tb01365.x (https://doi.org/10.1002%2Fj.1538-7305.1951.tb01365.x). (NB. Presented orally before the I.R.E. National Convention, New York City, March 1949.)
44. Karnaugh, Maurice (November 1953) [1953-04-23, 1953-03-17]. "The Map Method for Synthesis of Combinational Logic Circuits" (https://web.archive.org/web/20170416232229/http://philectrosophy.com/documents/The%20Map%20Method%20For%20Synthesis%20of.pdf) (PDF). *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*. **72** (5): 593–599. doi:10.1109/TCE.1953.6371932 (https://doi.org/10.1109%2FTCE.1953.6371932). Paper 53-217. Archived from the original (http://philectrosophy.com/documents/The%20Map%20Method%20For%20Synthesis%20of.pdf) (PDF) on 2017-04-16. Retrieved 2017-04-16. (NB. Also contains a short review by Samuel H. Caldwell.)
45. Wakerly, John F. (1994). *Digital Design: Principles & Practices*. New Jersey, USA: Prentice Hall. pp. 48–49, 222. ISBN 0-13-211459-3. ISBN 978-0-13-211459-2. (NB. The two page sections taken together say that K-maps are labeled with Gray code. The first section says that they are labeled with a code that changes only one bit between entries and the second section says that such a code is called Gray code.)
46. Brown, Frank Markham (2012) [2003, 1990]. "3.9.2 Maps". *Boolean Reasoning – The Logic of Boolean Equations* (reissue of 2nd ed.). Mineola, New York, USA: Dover Publications, Inc. p. 49. ISBN 978-0-486-42785-0. p. 49: "[…] Karnaugh's map orders the arguments of the discriminants according to the reflected binary code, also called the Gray code. […]" (xii+291+3 pages) 1st edition (https://web.archive.org/web/20170416231752/http://www2.fiit.stuba.sk/~kvasnicka/Free%20books/Brown_Boolean%20Reasoning.pdf)
47. Händler, Wolfgang (1958). *Ein Minimisierungsverfahren zur Synthese von Schaltkreisen (Minimisierungsgraphen)* (https://books.google.com/books?id=D58TAQAAIAAJ) (Dissertation) (in German). Potsdam, Germany: Technische Hochschule Darmstadt. D 17. (73 pages+app.) [10] (https://www.tib.eu/de/suchen/id/TIBKAT%3A044782241/Ein-Minimisierungsverfahren-zur-Synthese-von-Schaltkreisen/)
48. Berger, Erich R.; Händler, Wolfgang (1967) [1962]. Steinbuch, Karl W.; Wagner, Siegfried W. (eds.). *Taschenbuch der Nachrichtenverarbeitung* (in German) (2 ed.). Berlin, Germany: Springer-Verlag OHG. pp. 64, 1034–1035, 1036, 1038. LCCN 67-21079 (https://lccn.loc.gov/67-21079). Title No. 1036. p. 64: "[…] Übersichtlich ist die Darstellung nach *Händler*, die sämtliche Punkte, numeriert nach dem *Gray-Code* […], auf dem Umfeld eines Kreises anordnet. Sie erfordert allerdings sehr viel Platz. […]" [*Händler's* diagram, where all points, numbered according to the *Gray code*, are arranged on the circumference of a circle, is easily comprehensible. It needs, however, a lot of space.]
49. "Informatik Sammlung Erlangen (ISER)" (https://web.archive.org/web/20170516154655/https://www.rrze.fau.de/wir-ueber-uns/kooperationen/iser.shtml) (in German). Erlangen, Germany: Friedrich-Alexander Universität. 2012-03-13. Archived from the original (https://www.rrze.fau.de/wir-ueber-uns/kooperationen/iser.shtml) on 2017-05-16. Retrieved 2017-04-12.
50. "Informatik Sammlung Erlangen (ISER) – Impressum" (http://www.iser.uni-erlangen.de:80/index.php?ort_id=327&tree=0) (in German). Erlangen, Germany: Friedrich-Alexander Universität. 2012-03-13. Archived (https://web.archive.org/web/20120226004316/http://www.iser.uni-erlangen.de/index.php?ort_id=327&tree=0) from the original on 2012-02-26. Retrieved 2017-04-15.
51. Bhat, Girish S.; Savage, Carla Diane (1996). "Balanced Gray Codes" (https://www.combinatorics.org/ojs/index.php/eljc/article/view/v3i1r25/). *Electronic Journal of Combinatorics*. **3** (1). doi:10.37236/1249 (https://doi.org/10.37236%2F1249).
52. Donohue, Ryan (2003). "Synchronization in Digital Logic Circuits" (http://www.stanford.edu/class/ee183/handouts_spr2003/synchronization_pres.pdf) (PDF). Archived (https://web.archive.org/web/20180115012747/https://web.stanford.edu/class/ee183/handouts_spr2003/synchronization_pres.pdf) (PDF) from the original on 2018-01-15. Retrieved 2018-01-15.
53. Hulst, George D. (1962-02-06) [1957-11-15]. *Reflected binary code counter* (https://patentimages.storage.googleapis.com/a2/e5/21/f5abbb073bf21c/US3020481.pdf) (PDF). Nutley, New Jersey, USA: International Telephone and Telegraph Corporation (ITT). U.S. Patent 3,020,481 (https://patents.google.com/patent/US3020481). Serial No. 696793. Archived (https://web.archive.org/web/20200806210248/https://patentimages.storage.googleapis.com/a2/e5/21/f5abbb073bf21c/US3020481.pdf) (PDF) from the original on 2020-08-06. Retrieved 2020-08-06. (5 pages)
54. Powell, E. Alexander (June 1968). "Codes particularly useful for analogue to digital conversions". *A short note on useful codes for Fluidic Control Circuits* (https://dspace.lib.cranfield.ac.uk/bitstream/handle/1826/9559/COA_Memo_156_June_1968.pdf) (PDF). Cranfield, UK: The College of Aeronautics, Department of Production Engineering. pp. 7, 9. S2CID 215864694 (https://api.semanticscholar.org/CorpusID:215864694). CoA Memo 156. Archived (https://web.archive.org/web/20201215124905/https://dspace.lib.cranfield.ac.uk/bitstream/handle/1826/9559/COA_Memo_156_June_1968.pdf) (PDF) from the original on 2020-12-15. Retrieved 2020-12-15. (18 pages) (NB. The paper names the Glixon code *modified Gray code* and misspells Richard W. Hamming's name.)

55. Mehta, Huzefa; Owens, Robert Michael; Irwin, Mary Jane "Janie" (1996-03-22). *Some issues in Gray code addressing* (http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=497616&isnumber=10625). *Proceedings of the 6th Great Lakes Symposium on VLSI (GLSVLSI 96)*. IEEE Computer Society. pp. 178–181. doi:10.1109/GLSV.1996.497616 (https://doi.org/10.1109%2FGLSV.1996.497616). ISBN 978-0-8186-7502-7. ISSN 1066-1395 (https://www.worldcat.org/issn/1066-1395).
56. Doran, Robert "Bob" William (March 2007). *The Gray Code* (https://www.cs.auckland.ac.nz/research/groups/CDMTCS//researchreports/304bob.pdf) (PDF). CDMTCS Research Report Series. Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland, New Zealand. CDMTCS-304. Archived (https://web.archive.org/web/20200522091749/https://www.cs.auckland.ac.nz/research/groups/CDMTCS//researchreports/304bob.pdf) (PDF) from the original on 2020-05-22. Retrieved 2020-05-23. (25 pages)
57. Su, Ching-Long; Tsui, Chi-Ying; Despain, Alvin M. (1994). Low Power Architecture Design and Compilation Techniques for High-Performance Processors (http://www.scarpaz.com/2100-papers/Power%20Estimation/su94-low%20power%20architecture%20and%20compilation.pdf) (PDF) (Report). Advanced Computer Architecture Laboratory. ACAL-TR-94-01. Archived (https://web.archive.org/web/20200726170626/http://www.scarpaz.com/2100-papers/Power%20Estimation/su94-low%20power%20architecture%20and%20compilation.pdf) (PDF) from the original on 2020-07-26. Retrieved 2020-12-17.
58. Guo, Hui; Parameswaran, Sri (April–June 2010). "Shifted Gray encoding to reduce instruction memory address bus switching for low-power embedded systems". *Journal of Systems Architecture*. **56** (4–6): 180–190. doi:10.1016/j.sysarc.2010.03.003 (https://doi.org/10.1016%2Fj.sysarc.2010.03.003).
59. Dietz, Henry Gordon "Hank" (2002). "The Aggregate Magic Algorithms: Gray Code Conversion" (http://aggregate.org/MAGIC/#Gray%20Code%20Conversion). *The Aggregate*. Electrical and Computer Engineering Department, College of Engineering, University of Kentucky. Archived (https://web.archive.org/web/20201216122620/http://aggregate.org/MAGIC/#Gray%2520Code%2520Conversion) from the original on 2020-12-16. Retrieved 2020-12-16.
60. Guan, Dah-Jyh (1998). "Generalized Gray Codes with Applications". *Proceedings of the National Scientific Council, Republic of China, Part A*. **22**: 841–848. CiteSeerX 10.1.1.119.1344 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.119.1344).
61. Suparta, I. Nengah (2005). "A simple proof for the existence of exponentially balanced Gray codes" (https://doi.org/10.37236%2F1986). *Electronic Journal of Combinatorics*. **12**. doi:10.37236/1986 (https://doi.org/10.37236%2F1986).
62. Flahive, Mary Elizabeth; Bose, Bella (2007). "Balancing cyclic *R*-ary Gray codes" (https://doi.org/10.37236%2F949). *Electronic Journal of Combinatorics*. **14**. doi:10.37236/949 (https://doi.org/10.37236%2F949).
63. Strackx, Raoul; Piessens, Frank (2016). "Ariadne: A Minimal Approach to State Continuity" (https://distrinet.cs.kuleuven.be/software/sce/ariadne.html). *Usenix Security*. **25**.
64. Savage, Carla Diane; Winkler, Peter (1995). "Monotone Gray codes and the middle levels problem" (https://doi.org/10.1016%2F0097-3165%2895%2990091-8). *Journal of Combinatorial Theory, Series A*. **70** (2): 230–248. doi:10.1016/0097-3165(95)90091-8 (https://doi.org/10.1016%2F0097-3165%2895%2990091-8). ISSN 0097-3165 (https://www.worldcat.org/issn/0097-3165).
65. Savage, Carla Diane (1997-01-16). "Long cycles in the middle two levels of the Boolean lattice" (http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=DACA73A3D791549CE05443B7948527EC?doi=10.1.1.39.2249&rep=rep1&type=pdf). *Ars Combinatoria*. North Carolina State University, Raleigh, North Carolina, USA. **35** (A): 97–108. CiteSeerX 10.1.1.39.2249 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.2249). ISSN 0381-7032 (https://www.worldcat.org/issn/0381-7032). S2CID 15975960 (https://api.semanticscholar.org/CorpusID:15975960). Archived (https://web.archive.org/web/20200513191647/http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=DACA73A3D791549CE05443B7948527EC?doi=10.1.1.39.2249&rep=rep1&type=pdf) from the original on 2020-05-13. Retrieved 2020-05-13. (15 pages)
66. Goddyn, Luis (1999). "MATH 343 Applied Discrete Math Supplementary Materials" (https://web.archive.org/web/20150217160033/http://people.math.sfu.ca/~goddyn/Courses/343/supMaterials.pdf) (PDF). Department of Mathematics, Simon Fraser University. Archived from the original (http://www.math.sfu.ca/~goddyn/Courses/343/supMaterials.pdf) (PDF) on 2015-02-17.
67. Sawada, Joseph "Joe"; Wong, Dennis Chi-Him (2007). "A Fast Algorithm to generate Beckett–Gray codes". *Electronic Notes in Discrete Mathematics*. **29**: 571–577. doi:10.1016/j.endm.2007.07.091 (https://doi.org/10.1016%2Fj.endm.2007.07.091).
68. Richards, Richard Kohler (January 1971). "Snake-in-the-Box Codes". Written at Ames, Iowa, USA. *Digital Design*. New York, USA: Wiley-Interscience, John Wiley & Sons, Inc. pp. 206–207. ISBN 0-471-71945-5. LCCN 73-147235 (https://lccn.loc.gov/73-147235). (12+577+1 pages)
69. NZ 264738 (https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=NZ264738), Spedding, Norman Bruce, "A position encoder", published 1994-10-28
70. Spedding, Norman Bruce (1994-10-28). "The following is a copy of the provisional patent filed on behalf of Industrial Research Limited on 1994-10-28 – NZ Patent 264738" (http://www.winzurf.co.nz/Single_Track_Grey_Code_Patent/Single_track_Grey_code_encoder_patent.pdf) (PDF). Industrial Research Limited. NZ Patent 264738. Archived (https://web.archive.org/web/20171029205005/http://www.winzurf.co.nz/Single_Track_Grey_Code_Patent/Single_track_Grey_code_encoder_patent.pdf) (PDF) from the original on 2017-10-29. Retrieved 2018-01-14.
71. Hiltgen, Alain P.; Paterson, Kenneth G.; Brandestini, Marco (September 1996). "Single-Track Gray Codes" (http://ieeexplore.ieee.org/iel1/18/11236/00532900.pdf) (PDF). *IEEE Transactions on Information Theory*. **42** (5): 1555–1561. doi:10.1109/18.532900 (https://doi.org/10.1109%2F18.532900). Zbl 857.94007 (https://zbmath.org/?format=complete&q=an:857.94007).
72. Hiltgen, Alain P.; Paterson, Kenneth G. (September 2001). "Single-Track Circuit Codes" (http://www.hpl.hp.com/techreports/2000/HPL-2000-81.pdf) (PDF). *IEEE Transactions on Information Theory*. **47** (6): 2587–2595. CiteSeerX 10.1.1.10.8218 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.8218). doi:10.1109/18.945274 (https://doi.org/10.1109%2F18.945274). Archived (https://web.archive.org/web/20180115013155/http://www.hpl.hp.com/techreports/2000/HPL-2000-81.pdf) (PDF) from the original on 2018-01-15. Retrieved 2018-01-15.
73. Etzion, Tuvi; Schwartz, Moshe (November 1999) [1998-05-17]. "The Structure of Single-Track Gray Codes" (http://etzion.net.technion.ac.il/files/2016/02/P54.pdf) (PDF). *IEEE Transactions on Information Theory*. IT-45 (7): 2383–2396. CiteSeerX 10.1.1.14.8333 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.8333). doi:10.1109/18.796379 (https://doi.org/10.1109%2F18.796379). Archived (https://web.archive.org/web/20180115022531/http://etzion.net.technion.ac.il/files/2016/02/P54.pdf) (PDF) from the original on 2018-01-15. Retrieved 2018-01-15. Technical Report CS0937 (https://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi?1998/CS/CS0937)
74. Sillke, Torsten (1997) [1993-03-01]. "Gray-Codes with few tracks (a question of Marco Brandestini)" (http://www.mathematik.uni-bielefeld.de/~sillke/PROBLEMS/gray). Archived (https://web.archive.org/web/20171029202303/https://www.math.uni-bielefeld.de/~sillke/PROBLEMS/gray) from the original on 2017-10-29. Retrieved 2017-10-29.

75. Etzion, Tuvi; Paterson, Kenneth G. (May 1996). "Near Optimal Single-Track Gray Codes" (http://etzion.net.technion.ac.il/files/2016/02/P36.pdf) (PDF). *IEEE Transactions on Information Theory*. IT-42 (3): 779–789. CiteSeerX 10.1.1.14.1527 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.1527). doi:10.1109/18.490544 (https://doi.org/10.1109%2F18.490544). Archived (https://web.archive.org/web/20161030214251/http://etzion.net.technion.ac.il/files/2016/02/P36.pdf) (PDF) from the original on 2016-10-30. Retrieved 2018-04-08.
76. Ruskey, Frank; Weston, Mark (2005-06-18). "A Survey of Venn Diagrams: Symmetric Diagrams" (http://www.combinatorics.org/Surveys/ds5/VennSymmEJC.html). Dynamic Surveys. *Electronic Journal of Combinatorics*. doi:10.37236/26 (https://doi.org/10.37236%2F26).
77. Alciatore, David G.; Histand, Michael B. (1999). *Mechatronics* (http://mechatronics.colostate.edu/). McGraw–Hill Education – Europe. ISBN 978-0-07-131444-2.
78. Krishna (2008-05-11). "Gray code for QAM" (http://www.dsprelated.com/showthread/comp.dsp/96917-1.php). Archived (https://web.archive.org/web/20171029192539/https://www.dsprelated.com/showthread/comp.dsp/96917-1.php) from the original on 2017-10-29. Retrieved 2017-10-29.
79. Strang, Thomas; Dammann, Armin; Röckl, Matthias; Plass, Simon (October 2009). *Using Gray codes as Location Identifiers* (http://elib.dlr.de/60489/3/paper.pdf) (PDF). *6. GI/ITG KuVS Fachgespräch Ortsbezogene Anwendungen und Dienste* (in English and German). Oberpfaffenhofen, Germany: Institute of Communications and Navigation, German Aerospace Center (DLR). CiteSeerX 10.1.1.398.9164 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.398.9164). Archived (https://web.archive.org/web/20150501063457/http://elib.dlr.de/60489/3/paper.pdf) (PDF) from the original on 2015-05-01. Retrieved 2020-12-16. Lay summary (https://web.archive.org/web/20200903200722/https://www.researchgate.net/profile/Simon_Plass/publication/225003251_Using_Gray_codes_as_Location_Identifiers/links/02e7e53a733e49b2f6000000/Using-Gray-codes-as-Location-Identifiers.pdf) (PDF). (5/8 pages) [11] (https://web.archive.org/web/20201216231728/https://elib.dlr.de/60489/2/Strang_Thomas.pdf)
80. Greferath, Marcus (2009). "An Introduction to Ring-Linear Coding Theory". In Sala, Massimiliano; Mora, Teo; Perret, Ludovic; Sakata, Shojiro; Traverso, Carlo (eds.). *Gröbner Bases, Coding, and Cryptography* (https://archive.org/details/grbnerbasescodin00sala). Springer Science & Business Media. p. 220 (https://archive.org/details/grbnerbasescodin00sala/page/n226). ISBN 978-3-540-93806-4.
81. Solé, Patrick (2016-04-17). Hazewinkel, Michiel (ed.). *Kerdock and Preparata codes* (https://www.encyclopediaofmath.org/index.php/Kerdock_and_Preparata_codes). *Encyclopedia of Mathematics*. Springer Science+Business Media. ISBN 1-4020-0609-8. ISBN 978-1-4020-0609-8. Archived (https://web.archive.org/web/20171029191032/https://www.encyclopediaofmath.org/index.php/Kerdock_and_Preparata_codes) from the original on 2017-10-29. Retrieved 2017-10-29.
82. Spaulding, Carl P. (1965-07-12). *How to Use Shaft Encoders*. Monrovia, California, USA: Datex Corporation. (85 pages)
83. Wheeler, Edwin L. (1969-12-30) [1968-04-05]. *Analog to digital encoder* (https://patentimages.storage.googleapis.com/f0/c0/60/9c3231f7e8ed44/US3487460.pdf) (PDF). New York, USA: Conrac Corporation. U.S. Patent 3487460A (https://patents.google.com/patent/US3487460A). Serial No. 719026 (397812). Archived (https://web.archive.org/web/20200805102804/https://patentimages.storage.googleapis.com/f0/c0/60/9c3231f7e8ed44/US3487460.pdf) (PDF) from the original on 2020-08-05. Retrieved 2018-01-21. p. 5, left column 9, rows 15–22: "[…] The MOA-GILLHAM code is essentially the combination of the Gray code discussed thereinabove and the well known Datex code; the Datex code is disclosed in U.S. Patent 3,165,731. The arrangement is such that the Datex code defines the bits for the units count of the encoder and the Gray code defines the bits for each of the higher order decades, the tens, hundreds, etc. […]" (11 pages)
84. Dokter, Folkert; Steinhauer, Jürgen (1973-06-18). "2.4. Coding numbers in the binary system". *Digital Electronics* (https://books.google.com/books?id=hlRdDwAAQBAJ). Philips Technical Library (PTL) / Macmillan Education (Reprint of 1st English ed.). Eindhoven, Netherlands: The Macmillan Press Ltd. / N. V. Philips' Gloeilampenfabrieken. pp. 32, 39, 50–53. doi:10.1007/978-1-349-01417-0 (https://doi.org/10.1007%2F978-1-349-01417-0). ISBN 978-1-349-01419-4. SBN 333-13360-9. Retrieved 2020-05-11. p. 53: "[…] The Datex code […] uses the O'Brien code II within each decade, and reflected decimal numbers for the decimal transitions. For further processing, code conversion to the natural decimal notation is necessary. Since the O'Brien II code forms a 9s complement, this does not give rise to particular difficulties: whenever the code word for the tens represents an odd number, the code words for the decimal units are given as the 9s complements by inversion of the fourth binary digit. […]" (270 pages)
85. Dokter, Folkert; Steinhauer, Jürgen (1975) [1969]. "2.4.4.6. Einschrittige Kodes". *Digitale Elektronik in der Meßtechnik und Datenverarbeitung: Theoretische Grundlagen und Schaltungstechnik*. Philips Fachbücher (in German). **I** (improved and extended 5th ed.). Hamburg, Germany: Deutsche Philips GmbH. pp. 41, 48, 51, 58, 60–61. ISBN 3-87145-272-6. (xii+327+3 pages)
86. "…accurate liquid level metering – at ANY DISTANCE!" (https://books.google.com/books?id=B2Q7AAAAMAAJ&q=%22Varec+pulse+code%22&dq=%22Varec+pulse+code%22). *Petroleum Refiner* (Advertisement). Gulf Publishing Company. **33** (9): 368. September 1954. ISSN 0096-6517 (https://www.worldcat.org/issn/0096-6517). p. 368: "[…] The complete dispatching operation, gauging, and remote control is integrated into one single unitized system when a "Varec" Pulse Code Telemetering System is installed. […]"
87. Bishup, Bernard W.; Repeta, Anthony A.; Giarrizzo, Frank C. (1968-08-13) [1963-04-03]. "Telemetering and supervisory control system having normally continuous telemetering signals" (https://patents.google.com/patent/US3397386A/en). Leeds and Northrup Co. US3397386A. [12] (https://patentimages.storage.googleapis.com/c8/f3/cc/936fef75655a2c/US3397386.pdf)
88. "Encoder Pulse Format". *Installation and Operations Manual for the Model 1900 Micro 4-Wire Transmitter* (https://www.varec.com/web/wp-content/uploads/2018/05/IOM016_1900.pdf) (PDF). Cypress, California, USA: Whessoe Varec, Inc. January 1993 [1991-07-01]. pp. 04-4–04-8. 33-08461. Archived (https://web.archive.org/web/20200516191438/https://www.varec.com/web/wp-content/uploads/2018/05/IOM016_1900.pdf) (PDF) from the original on 2020-05-16. Retrieved 2020-05-16. (38 pages) (NB. Position 5 for "Inches" on page 04-8 should read "0111" rather than "1111".)
89. "2.2.3.3 MSP Level Data Format". *Varec Model 1900 – Micro 4-Wire Transmitter (BSAP to Mark / Space Protocol (MSP)) – Application Notes* (https://www.emerson.com/documents/automation/manuals-guides-application-notes-varec-model-1900-protocol-ras-en-133360.pdf) (PDF). Emerson Electric. pp. 11–14. Archived (https://web.archive.org/web/20200516183338/https://www.emerson.com/documents/automation/manuals-guides-application-notes-varec-model-1900-protocol-ras-en-133360.pdf) (PDF) from the original on 2020-05-16. Retrieved 2020-05-16. (vi+33 pages)

90. Wightman, Eric Jeffrey (1972). "Chapter 6. Displacement measurement" (https://books.google.com/books?id=8WEhBQAAQBAJ&pg=PA122). *Instrumentation in Process Control* (https://books.google.com/books?id=8WEhBQAAQBAJ) (1 ed.). London, UK: Butterworth & Co (Publishers) Ltd. pp. 122–123. ISBN 0-408-70293-1. ISBN 1-48316335-0, 978-1-48316335-2. p. 122–123: "[…] Other forms of code are also well known. Among these are the Royal Radar Establishment code; The Excess Three decimal code; Gillham code which is recommended by ICAO for automatic height transmission for air traffic control purposes; the Petherick code, and the Leslie and Russell code of the National Engineering Laboratory. Each has its particular merits and they are offered as options by various encoder manufacturers. […]" (12+367+5 pages)
91. Phillips, Darryl (2012-07-26) [1998]. "Altitude – MODEC ASCII" (https://web.archive.org/web/20120726003224/http://www.airsport-corp.com/modecascii.txt). AirSport Avionics. Archived from the original (http://www.airsport-corp.com/modecascii.txt) on 2012-07-26.
92. Stewart, K. (2010-12-03). "Aviation Gray Code: Gillham Code Explained" (http://www.ccsinfo.com/forum/viewtopic.php?p=140960#140960). Custom Computer Services (CCS). Archived (https://web.archive.org/web/20180116184525/http://www.ccsinfo.com/forum/viewtopic.php?p=140960) from the original on 2018-01-16. Retrieved 2018-01-14.
93. Leslie, William "Bill" H. P.; Russell, A. (1964). *A cyclic progressive decimal code for simple translation to decimal and analogue outputs* (Report). East Kilbride, Glasgow, UK: National Engineering Laboratory. NEL Report 129. (17 pages)
94. Leslie, William "Bill" H. P. (1974). "The work on NC at NEL" (https://link.springer.com/chapter/10.1007/978-1-349-01921-2_30). In Koenigsberger, Franz; Tobias, Stephen Albert (eds.). *Proceedings of the Fourteenth International Machine Tool Design and Research Conference, 12–14 September 1973* (https://books.google.com/books?id=bFVdDwAAQBAJ&pg=PA215&lpg=PA215). The Macmillan Press Ltd. pp. 215–224 [215, 217]. ISBN 978-1-34901921-2. LCCN 73-16545 (https://lccn.loc.gov/73-16545). SBN 333-14913-0. ISBN 1-34901921-6.
95. Hoklas, Archibald (1989-09-06) [1988-04-29]. "Abtastvorrichtung zur digitalen Weg- oder Winkelmessung" (https://web.archive.org/web/20180118015239/https://depatisnet.dpma.de/DepatisNet/depatisnet/DD000000271603A1_all_pages.pdf?window=1&space=menu&content=download_doc_verify&action=download_doc&docid=DD000000271603A1&so=asc&sf=vn&firstdoc=0&struct=&Cl=2&Bi=1&Ab=1&De=2&Dr=5&Pts=&Pa=&We=&Sr=&Eam=&Cor=&Aa=&NrFaxPages=5&pdfpage=2&pdfmatrix=1x1&origin=pdf_window&verify_str=null) (PDF) (in German). VEB Schiffselektronik Johannes Warnke. GDR Patent DD271603A1. WP H 03 M / 315 194 8. Archived from the original (https://depatisnet.dpma.de/DepatisNet/depatisnet/DD000000271603A1_all_pages.pdf?window=1&space=menu&content=download_doc_verify&action=download_doc&docid=DD000000271603A1&so=asc&sf=vn&firstdoc=0&struct=&Cl=2&Bi=1&Ab=1&De=2&Dr=5&Pts=&Pa=&We=&Sr=&Eam=&Cor=&Aa=&NrFaxPages=5&pdfpage=2&pdfmatrix=1x1&origin=pdf_window&verify_str=null) (PDF) on 2018-01-18. Retrieved 2018-01-18 – via DEPATIS. [13] (https://depatisnet.dpma.de/DepatisNet/depatisnet?action=bibdat&docid=DD000000271603A1) [14] (https://depatisnet.dpma.de/DepatisNet/depatisnet?action=pdf&docid=DD000000271603A1)
96. Hoklas, Archibald (2005). "Gray code – Unit distance code" (http://www.ahok.de/en/hoklas-code.html). Archived (https://web.archive.org/web/20180115012854/http://www.ahok.de/en/hoklas-code.html) from the original on 2018-01-15. Retrieved 2018-01-15.
97. Hoklas, Archibald (2005). "Gray-Kode – Einschrittiger Abtastkode" (http://www.ahok.de/dt/hoklas-code.html) (in German). Archived (https://web.archive.org/web/20180115012827/http://www.ahok.de/dt/hoklas-code.html) from the original on 2018-01-15. Retrieved 2018-01-15.
98. Petherick, Edward John; Hopkins, A. J. (1958). *Some Recently Developed Digital Devices for Encoding the Rotations of Shafts* (Technical Note MS21). Farnborough, UK: Royal Aircraft Establishment (RAE).
99. "Digitizer als Analog-Digital-Wandler in der Steuer-, Meß- und Regeltechnik" (https://web.archive.org/web/20200521211656/https://www.posital.com/media/posital_media/documents/TechnischeMitteilung_May1963.pdf) (PDF). *Technische Mitteilungen*. Relais, elektronische Geräte, Steuerungen (in German). No. 13. Cologne-Niehl, Germany: Franz Baumgartner (FraBa). May 1963. pp. 1–2. Archived from the original (https://www.posital.com/media/posital_media/documents/TechnischeMitteilung_May1963.pdf) (PDF) on 2020-05-21. Retrieved 2020-05-21. pp. 1–2: "[…] Die Firma Harrison Reproduction Equipment, Farnborough/England […] hat in jahrelanger Entwicklung in Zusammenarbeit mit der Britischen Luftwaffe und britischen Industriebetrieben den mechanischen Digitizer […] zu einer technischen Reife gebracht, die fast allen Anforderungen […] genügt. […] Um bei der dezimalen Entschlüsselung des verwendeten Binärcodes zu eindeutigen und bei der Übergabe von einer Dezimalstelle zur anderen in der Reihenfolge immer richtigen Ergebnissen zu kommen, wurde ein spezieller Code entwickelt, der jede Möglichkeit einer Fehlaussage durch sein Prinzip ausschließt und der außerdem durch seinen Aufbau eine relativ einfache Entschlüsselung erlaubt. Der Code basiert auf dem Petherick-Code. […]" (4 pages)
00. Charnley, C. J.; Bidgood, R. E.; Boardman, G. E. T. (October 1965). "The Design of a Pneumatic Position Encoder" (https://ac.els-cdn.com/S1474667017689559/1-s2.0-S1474667017689559-main.pdf?_tid=f0c1e48e-f95b-11e7-ad9a-00000aab0f01&acdnat=1515956073_9006e89e176c6a840b5454c38525240b) (PDF). *IFAC Proceedings Volumes*. The College of Aeronautics, Cranfield, Bedford, England. **2** (3): 75–88. doi:10.1016/S1474-6670(17)68955-9 (https://doi.org/10.1016%2FS1474-6670%2817%2968955-9). Chapter 1.5. Retrieved 2018-01-14.
01. Hollingdale, Stuart H. (1958-09-19). "Session 14. Data Processing" (http://www.chilton-computing.org.uk/acl/literature/manuals/nottingham/p014.htm). *Applications of Computers* (http://www.chilton-computing.org.uk/acl/literature/manuals/nottingham/contents.htm). *Atlas – Application of Computers, University of Nottingham 15–19 September 1958* (Conference paper). Archived (https://web.archive.org/web/20200525151358/http://www.chilton-computing.org.uk/acl/literature/manuals/nottingham/p014.htm) from the original on 2020-05-25. Retrieved 2020-05-25.
02. O'Brien, Joseph A. (May 1956) [1955-11-15, 1955-06-23]. "Cyclic Decimal Codes for Analogue to Digital Converters" (https://pdfslide.net/documents/cyclic-decimal-codes-for-analogue-to-digital-converters.html). *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*. Bell Telephone Laboratories, Whippany, New Jersey, USA. **75** (2): 120–122. doi:10.1109/TCE.1956.6372498 (https://doi.org/10.1109%2FTCE.1956.6372498). ISSN 0097-2452 (https://www.worldcat.org/issn/0097-2452). S2CID 51657314 (https://api.semanticscholar.org/CorpusID:51657314). Paper 56-21. Retrieved 2020-05-18. (3 pages) (NB. This paper was prepared for presentation at the AIEE Winter General Meeting, New York, USA, 1956-01-30 to 1956-02-03.)
03. Steinbuch, Karl W., ed. (1962). Written at Karlsruhe, Germany. *Taschenbuch der Nachrichtenverarbeitung* (in German) (1 ed.). Berlin / Göttingen / New York: Springer-Verlag OHG. pp. 71–74, 97, 761–764, 770, 1080–1081. LCCN 62-14511 (https://lccn.loc.gov/62-14511).
04. Steinbuch, Karl W.; Weber, Wolfgang; Heinemann, Traute, eds. (1974) [1967]. *Taschenbuch der Informatik – Band II – Struktur und Programmierung von EDV-Systemen*. *Taschenbuch der Nachrichtenverarbeitung* (in German). **2** (3 ed.). Berlin, Germany: Springer Verlag. pp. 98–100. ISBN 3-540-06241-6. LCCN 73-80607 (https://lccn.loc.gov/73-80607). ISBN 978-3-540-06241-7.

05. Foss, Frederic A. (1960-12-27) [1954-12-17]. "Control Systems" (https://patentimages.storage.googleapis.com/3d/a8/16/1dc616c432ca95/US2966670.pdf) (PDF). International Business Machines Corp. Fig. 7, Fig. 8, Fig. 11. U.S. Patent 2966670A (https://patents.google.com/patent/US2966670A). Serial No. 475945. Archived (https://web.archive.org/web/20200621145238/https://patentimages.storage.googleapis.com/3d/a8/16/1dc616c432ca95/US2966670.pdf) (PDF) from the original on 2020-06-21. Retrieved 2020-08-05. (14 pages) (NB. The author called his code 2*-4-2-1 (+9-±7-±3-±1) reflected decimal code.)
06. Foss, Frederic A. (December 1954). "The Use of a Reflected Code in Digital Control Systems". *IRE Transactions on Electronic Computers*. **EC-3** (4): 1–6. doi:10.1109/IREPGELC.1954.6499244 (https://doi.org/10.1109%2FIREPGELC.1954.6499244). ISSN 2168-1740 (https://www.worldcat.org/issn/2168-1740). (6 pages)
07. Evans, David Silvester (1958). "[unknown]" (https://books.google.com/books?id=mwJvieqKGFsC&q=%22Watts+code%22&dq=%22Watts+code%22&hl=en&sa=X&ved=0ahUKEwjM5-SRosrpAhVBmqQKHfulBZU4FBDoAQhqMAc). *Transactions*. Institute of Measurement and Control. 10–12: 87. Cite uses generic title (help) (NB. The Watts code was called W.R.D. code or Watts Reflected Decimal to distinguish it from other codes used at Hilger & Watts Ltd..)
08. Benjamin, P. W.; Nicholls, G. S. (1963). "3.2.2 Electromechanical Digitizers". *Measurement of Neutron Spectra by Semi-Automatic Scanning of Recoil Protons in Photographic Emulsions* (https://books.google.com/books?id=tD5YAAAAYAAJ&pg=PT4). United Kingdom Atomic Energy Authority, Atomic Weapons Research Establishment, UK: U.S. Department of Energy. pp. 8–10, 19. AWRE Report No. NR 5/63. [15] (https://web.archive.org/web/20200525193325/https://books.googleusercontent.com/books/content?req=AKW5QadxAC8d1K6TXnjLMOKRt0wxLI2XEnyvs5b7wXlxCaZ4XdYW6qcmagghi7O1TLA4DL5nybOmiY0ue2ljEH9L2NvnP6Z6vN54dNvpxHnCIK79h1xoamtO5-waq-mDh2BPHJE9eJkmkoUl7uyPvrbWd7gNM8gpQKbx82WI_3gYiZHy-aHdQGQAFOkacvI1Kf52lu4i8_cuMNYztfa--t2eGiIk5ku1bP7T7n378Q2rNZRCMz3JCXeT0EzYNMkJzlpYSiP2FiQ8) (23 pages)
09. Klinkowski, James J. (1967-03-14) [1964-03-23]. "Electronic Diode Matrix Decoder Circuits" (https://patentimages.storage.googleapis.com/71/6a/f4/044b43b93097e7/US3309695.pdf) (PDF). Detroit, Michigan, USA: Burroughs Corporation. U.S. Patent 3309695A (https://patents.google.com/patent/US3309695A). Serial No. 353845. Archived (https://web.archive.org/web/20200523144819/https://patentimages.storage.googleapis.com/71/6a/f4/044b43b93097e7/US3309695.pdf) (PDF) from the original on 2020-05-23. Retrieved 2020-05-23. (5 pages) [16] (https://web.archive.org/web/20200805103732/https://patentimages.storage.googleapis.com/97/33/74/aec52610026e59/USRE26780.pdf)
10. Klinkowski, James J. (1970-03-31) [1966-12-22]. "Binary-coded decimal signal converter" (https://patentimages.storage.googleapis.com/65/d6/45/f7837c143fa3da/US3504363.pdf) (PDF). Detroit, Michigan, USA: Burroughs Corporation. U.S. Patent 3504363A (https://patents.google.com/patent/US3504363A). Serial No. 603926. Archived (https://web.archive.org/web/20200523145655/https://patentimages.storage.googleapis.com/65/d6/45/f7837c143fa3da/US3504363.pdf) (PDF) from the original on 2020-05-23. Retrieved 2020-05-23. (7 pages)
11. "[unknown]". *Electrical Design News*. Rogers Publishing Company. **12**. 1967. ISSN 0012-7515 (https://www.worldcat.org/issn/0012-7515). Cite uses generic title (help) [17] (https://books.google.com/books?id=MThAAQAAIAAJ&q=%22Watts+code%22+%22Excess-3+Gray+Code%22&dq=%22Watts+code%22+%22Excess-3+Gray+Code%22&hl=en&sa=X&ved=0ahUKEwjzpIbVgMrpAhUBNOwKHd9CB1YQ6AEIMDAB)[18] (https://books.google.com/books?id=MThAAQAAIAAJ&q=%22Watts+code%22+%22Excess-3+Gray+Code%22&dq=%22Watts+code%22+%22Excess-3+Gray+Code%22&hl=en&sa=X&ved=0ahUKEwjzpIbVgMrpAhUBNOwKHd9CB1YQ6AEIMDAB)
12. Tóth-Zentai, Györgyi (1979-10-05). "Some Problems Of Angular Rotational Digital Converters" (https://pp.bme.hu/ee/article/view/4838). *Periodica Polytechnica Electrical Engineering*. Department of Electronics Technology, Technical University, Budapest, Hungary. **23** (3–4): 265–270 [266]. Retrieved 2020-05-23. [19] (https://pp.bme.hu/ee/article/download/4838/3943/) (6 pages) (NB. Shows a 6-digit Watts code.)
13. Savard, John J. G. (2018) [2006]. "Decimal Representations" (http://www.quadibloc.com/comp/cp0203.htm). *quadibloc*. Archived (https://web.archive.org/web/20180716101321/http://www.quadibloc.com/comp/cp0203.htm) from the original on 2018-07-16. Retrieved 2018-07-16.
14. Turvey, Jr., Frank P. (1958-07-29) [1956-05-17]. "Pulse-Count Coder" (https://patentimages.storage.googleapis.com/14/07/c6/404dafe27dc829/US2845617.pdf) (PDF). Nutley, New Jersey, USA: International Telephone and Telegraph Corporation. U.S. Patent 2845617A (https://patents.google.com/patent/US2845617A). Serial No. 585494. Archived (https://web.archive.org/web/20200523115811/https://patentimages.storage.googleapis.com/14/07/c6/404dafe27dc829/US2845617.pdf) (PDF) from the original on 2020-05-23. Retrieved 2020-05-23. (5 pages)
15. Glixon, Harry Robert (March 1957). "Can You Take Advantage of the Cyclic Binary-Decimal Code?" (https://books.google.com/books?id=-_5IAQAAIAAJ). *Control Engineering*. Technical Publishing Company, a division of Dun-Donnelley Publishing Corporation, Dun & Bradstreet Corp. **4** (3): 87–91. ISSN 0010-8049 (https://www.worldcat.org/issn/0010-8049). (5 pages)
16. Borucki, Lorenz; Dittmann, Joachim (1971) [July 1970, 1966, Autumn 1965]. "2.3 Gebräuchliche Codes in der digitalen Meßtechnik". Written at Krefeld / Karlsruhe, Germany. *Digitale Meßtechnik: Eine Einführung* (in German) (2 ed.). Berlin / Heidelberg, Germany: Springer-Verlag. pp. 10–23 [12–14]. doi:10.1007/978-3-642-80560-8 (https://doi.org/10.1007%2F978-3-642-80560-8). ISBN 3-540-05058-2. LCCN 75-131547 (https://lccn.loc.gov/75-131547). ISBN 978-3-642-80561-5. (viii+252 pages) 1st edition (https://books.google.com/books?id=NwyCBwAAQBAJ&pg=PA15&lpg=PA15) (NB. Like Kämmerer, the authors describe a 6-bit 20-cyclic Glixon code.)
17. Kämmerer, Wilhelm (May 1969). "II.15. Struktur: Informationsdarstellung im Automaten". Written at Jena, Germany. In Frühauf, Hans; Kämmerer, Wilhelm; Schröder, Kurz; Winkler, Helmut (eds.). *Digitale Automaten – Theorie, Struktur, Technik, Programmieren*. Elektronisches Rechnen und Regeln (in German). **5** (1 ed.). Berlin, Germany: Akademie-Verlag GmbH. p. 173. License no. 202-100/416/69. Order no. 4666 ES 20 K 3. (NB. A second edition 1973 exists as well. Like Borucki and Dittmann, but without naming it Glixon code, the author creates a 20-cyclic tetradic code from Glixon code and a Glixon code variant with inverted high-order bit.)
18. Paul, Matthias R. (1995-08-10) [1994]. "Unterbrechungsfreier Schleifencode" (http://www.uni-bonn.de/~uzs180/download/mpbcd102.zip) [Continuous loop code]. 1.02 (in German). Retrieved 2008-02-11. (NB. The author called this code *Schleifencode* (English: "loop code"). It differs from Gray BCD code only in the encoding of state 0 to make it a cyclic unit-distance code for full-circle rotatory applications. Avoiding the all-zero code pattern allows for loop self-testing and to use the data lines for uninterrupted power distribution.)

19. Klar, Rainer (1970-02-01). *Digitale Rechenautomaten – Eine Einführung* (https://books.google.com/books?id=QnqVDwAAQBAJ&pg=PA17) [*Digital Computers – An Introduction*]. Sammlung Göschen (in German). 1241/1241a (1 ed.). Berlin, Germany: Walter de Gruyter & Co. / G. J. Göschen'sche Verlagsbuchhandlung. p. 17. ISBN 3-11-083160-0. ISBN 978-3-11-083160-3. Archiv-Nr. 7990709. Archived (https://web.archive.org/web/20200601100645/https://books.google.de/books?id=QnqVDwAAQBAJ&pg=PA17&lpg=PA17&redir_esc=y#v=onepage&q&f=false) from the original on 2020-06-01. Retrieved 2020-04-13. (205 pages) (NB. A 2019 reprint of the first edition is available under ISBN 3-11002793-3, 978-3-11002793-8. A reworked and expanded 4th edition exists as well.)
20. Klar, Rainer (1989) [1988-10-01]. *Digitale Rechenautomaten – Eine Einführung in die Struktur von Computerhardware* [*Digital Computers – An Introduction into the structure of computer hardware*]. Sammlung Göschen (in German). **2050** (4th reworked ed.). Berlin, Germany: Walter de Gruyter & Co. p. 28. ISBN 3-11011700-2. ISBN 978-3-11011700-4. (320 pages) (NB. The author called this code *Einheitsabstandscode* (English: "unit-distance code"). By swapping two bit rows and inverting one of them, it can be transferred into the O'Brien code II, whereas by swapping and inverting two bit rows, it can be transferred into the Petherick code.)

Further reading

- Richards, Richard Kohler (1955). *Arithmetic Operations in Digital Computers* (https://books.google.com/books?id=BI5QAAAAMAAJ) (5 ed.). New York, USA: D. Van Nostrand Co., Inc.
- Richards, Richard Kohler (1967). *Electronic Digital Components and Circuits*. D. Van Nostrand Co., Inc. pp. 490, 500–504, 510–511.
- Black, Paul E. (2004-02-25). "Gray code" (https://xlinux.nist.gov/dads/HTML/graycode.html). NIST.
- Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian P. (2007). "Section 22.3. Gray Codes" (http://apps.nrbook.com/empanel/index.html#pg=1166). *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York, USA: Cambridge University Press. ISBN 978-0-521-88068-8.
- Savage, Carla Diane (1997). "A Survey of Combinatorial Gray Codes" (http://www4.ncsu.edu/~savage/AVAILABLE_FOR_MAILING/survey.ps). *SIAM Review*. Society for Industrial and Applied Mathematics (SIAM). **39** (4): 605–629. CiteSeerX 10.1.1.39.1924 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.1924). doi:10.1137/S0036144595295272 (https://doi.org/10.1137%2FS0036144595295272). JSTOR 2132693 (https://www.jstor.org/stable/2132693).
- Wilf, Herbert Saul (1989). "Chapters 1–3". *Combinatorial algorithms: An update* (https://archive.org/details/combinatorialalg0000wilf). Society for Industrial and Applied Mathematics (SIAM). ISBN 0-89871-231-9. ISBN 978-0-89871-231-5.
- Dewar, Megan; Stevens, Brett (2012-08-29). *Ordering Block Designs – Gray Codes, Universal Cycles and Configuration*. CMS Books in Mathematics (1 ed.). New York, USA: Springer Science+Business Media. doi:10.1007/978-1-4614-4325-4 (https://doi.org/10.1007%2F978-1-4614-4325-4). ISBN 978-1-46144324-7. ISSN 1613-5237 (https://www.worldcat.org/issn/1613-5237). ISBN 1-46144324-5.
- Maxfield, Clive "Max" (2012-10-01) [2011-05-28]. "Gray Code Fundamentals" (https://www.eetimes.com/document.asp?doc_id=1278809). *Design How-To*. EETimes. Part 1. Archived (https://web.archive.org/web/20171030135842/https://www.eetimes.com/document.asp?doc_id=1278809) from the original on 2017-10-30. Retrieved 2017-10-30. Part 2 (https://www.eetimes.com/document.asp?doc_id=1278827) Part 3 (https://www.eetimes.com/document.asp?doc_id=1278853)
- Warren, Jr., Henry S. (2013). "Chapter 13: Gray Code". *Hacker's Delight* (2 ed.). Addison Wesley – Pearson Education, Inc. pp. 311–317. ISBN 978-0-321-84268-8. (7 pages)
- Zinovik, Igor; Kroening, Daniel; Chebiryak, Yury (2008-03-21). "Computing Binary Combinatorial Gray Codes Via Exhaustive Search With SAT Solvers" (http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=4475352&arnumber=4475394&count=44&index=39). *IEEE Transactions on Information Theory*. IEEE. **54** (4): 1819–1823. doi:10.1109/TIT.2008.917695 (https://doi.org/10.1109%2FTIT.2008.917695). hdl:20.500.11850/11304 (https://hdl.handle.net/20.500.11850%2F11304). (5 pages)
- O'Brien, Joseph A. (June 1957). "Unit-Distance Binary-Decimal Code Translators" (https://www.researchgate.net/publication/250929745_Unit-Distance_Binary-Decimal_Code_Translators). *IRE Transactions on Electronic Computers*. **EC-6** (2): 122–123. doi:10.1109/TEC.1957.5221585 (https://doi.org/10.1109%2FTEC.1957.5221585). ISSN 0367-9950 (https://www.worldcat.org/issn/0367-9950). Retrieved 2020-05-25. (2 pages)
- Barr, K. G. (March 1981). "A decimal Gray code – Easily converted for shaft position coding" (https://worldradiohistory.com/hd2/IDX-Site-Early-Radio/Archive-Wireless-World-IDX/80s/Wireless-World-1981-03-OCR-Page-0046.pdf) (PDF). *Wireless World*. Vol. 87 no. 1542. Faculty of Natural Sciences, University of the West Indies. pp. 86–87. Archived (https://web.archive.org/web/20200728093447/https://worldradiohistory.com/hd2/IDX-Site-Early-Radio/Archive-Wireless-World-IDX/80s/Wireless-World-1981-03-OCR-Page-0046.pdf) (PDF) from the original on 2020-07-28. Retrieved 2020-07-28.

External links

- "Gray Code" demonstration (http://demonstrations.wolfram.com/BinaryGrayCode/) by Michael Schreiber, Wolfram Demonstrations Project (with Mathematica implementation). 2007.
- NIST Dictionary of Algorithms and Data Structures: Gray code (https://xlinux.nist.gov/dads/HTML/graycode.html).
- Hitch Hiker's Guide to Evolutionary Computation, Q21: What are Gray codes, and why are they used? (https://web.archive.org/web/20151026021510/http://www.aip.de/~ast/EvolCompFAQ/Q21.htm), including C code to convert between binary and BRGC.
- Dragos A. Harabor uses Gray codes in a 3D digitizer (https://web.archive.org/web/20151122120754/http://www.ugcs.caltech.edu/~dragos/3DP/coord.html).
- Single-track gray codes, binary chain codes (Lancaster 1994 (http://tinaja.com/text/chain01.html)), and linear feedback shift registers are all useful in finding one's absolute position on a single-track rotary encoder (or other position sensor).
- AMS Column: Gray codes (https://www.ams.org/featurecolumn/archive/gray.html)
- Optical Encoder Wheel Generator (http://www.bushytails.net/~randyg/encoder/encoderwheel.html)

- ProtoTalk.net – Understanding Quadrature Encoding (https://web.archive.org/web/20110724021700/http://prototalk.net/forums/showthread.php?t=78) – Covers quadrature encoding in more detail with a focus on robotic applications

Retrieved from "https://en.wikipedia.org/w/index.php?title=Gray_code&oldid=1023725478"

This page was last edited on 18 May 2021, at 00:31 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.