# Unit 7 Number codes.

- Distinguish between the number system or system base and the binary representation of the digits used in the system.
- In weighted codes a weight is associated with the bit position.
- In non weighted number codes a weight can not be associated with the bit position.
- In self-complementing codes, the binary representations of numbers equidistant from the centre of the range are bit complements of each other.
- In cyclic number codes only one bit changes in going from a number to the next number in the sequence.
- In reflective codes, the leading bit changes at the midpoint and the codes in the second half repeat the first half in reverse order.
- The Gray code in the most common example of cyclic codes.

In the number systems which we discussed in the previous unit, the representation of numbers in decimal, binary and hexadecimal mode and the methods by which the operations of addition, subtraction and changing of the sign may be carried out. There are other ways in which numbers can be represented which have advantages when data is to be transmitted between systems or collected from instrumentation systems. the main advantages of these alternative representation systems is that they permit error checking to be carried out and they may give a more compact representation of the number and they help prevent errors of measurement being introduced at the data collection stage.

**Weighted codes.** A 4 bit binary code can represent $2^4 = 16$ distinct numbers. If a four bit binary number is used to represent the decimal digits from 0 to 9, satisfactory and distinct representations can be assigned in a number of ways. For instance, we could use the assignment table:

| | |
|---|---|
| 0 | 1011 |
| 1 | 1010 |
| 2 | 0110 |
| 3 | etc. |

in which the decimal digits are not represented by the equivalent binary number counted from zero. This would be an example of a non weighted code.

On the other hand, a weighted code is one where a bit, $d$, in a specific position, $w$, always has a specific weight and a decimal digit, $N$ is computed from:

$$N = d_3w_3 + d_2w_2 + d_1w_1 + d_0w_0$$

If in the weighted code, $w_3$, $w_2$, $w_1$, $w_0$, the $w_3$ is assigned the weight 8, $w_2$ is assigned the weight 4, $w_1$ is assigned the weight 2 and $w_0$ is assigned the weight 1, then the decimal number 9 would be represented by 1001 where $9 = 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$. There is only a small set of such weighted codes, the most common of which is the BCD code or binary coded decimal code. The table below shows the common codes.

| Decimal | BCD = 8421 | 7421 | 4221 | $84\overline{21}$ |
|---|---|---|---|---|
| 0 | 0000 | 0000 | 0000 | 0000 |
| 1 | 0001 | 0001 | 0001 | 0111 |
| 2 | 0010 | 0010 | 0010 | 0110 |
| 3 | 0011 | 0011 | 0011 | 0101 |
| 4 | 0100 | 0100 | 1000 | 0100 |
| 5 | 0101 | 0101 | 0111 | 1011 |
| 6 | 0110 | 0110 | 1100 | 1010 |
| 7 | 0111 | 1000 | 1101 | 1001 |
| 8 | 1000 | 1001 | 1110 | 1000 |
| 9 | 1001 | 1010 | 1111 | 1111 |

Note that in the last column, for the code $84\overline{21}$, the formula for the number is $N = d_3w_3 + d_2w_2 + d_1\overline{w_1} + d_0\overline{w_0}$ where the complement of $w_1$ and $w_0$ is used and the corresponding digits are effectively subtracted from $N$.

Also the 4221 and the $84\overline{21}$ codes are self- complementing codes, that is, codes for which the representation of a number $N$ is obtained by complementing the bits of the representation of $9 - N$. That is, if 3 is represented by 0101 then $9 - 3 = 6$ is represented by 1010.

The 4221 code has an alternative representation in which decimal 4 is represented by 0110 instead of 1000 and decimal 5 is represented by 1001 instead of 0111, etc. By convention, the assignment shown in the table is the representation which is associated with the 4221 code and which also has the self-complementing property. It is easily shown that the other possible 4221 code representation does not have the self-complementing property.

These weighted self-complementing codes were developed for us in the earlier computers as this representation made the arithmetic operation of subtraction faster to implement. However, they are now used less frequently.

**Non weighted binary codes.** There are advantages associated with the use of codes in which the weight is not associated with a particular position and as a result the number, $N$, can not be expressed in the form

$$N = d_3w_3 + d_2w_2 + d_1w_1 + d_0w_0.$$

The advantage of using such codes are associated with the operation of measurement systems and instruments and also in facilitating some computations.

Consider the problem of measuring the angle of a shaft by using a set of brush contacts and a segmented metal disk or a set of optical sensors and a disk having transparent and opaque sectors as shown in Figure 7.1.

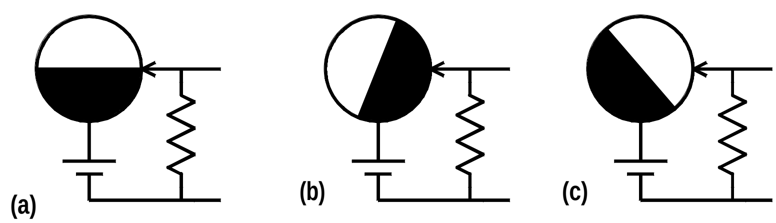


Figure 7.1: Single bit angle encoder.

The darker region on the disk represents a metal coating, the light area represents insulator. When the disk rotates to make contact between the shaft, the metal sector of the disk and the brush contact, a digital 1 signal will be present at the output as shown in Figure 7.1 (b). When the shaft is rotated so that the insulator is in contact with the brush, a digital 0 signal is present at the output as shorn in Figure 7.1 (c). The system will therefore have a single bit output which indicates the position of the shaft with an angular resolution of 180°.

Higher angular resolution is obtained by subdividing the sectors using a sectored disk such as that shown in Figure 7.2 which has four in line contacts to the sectors and which gives an angular resolution of $\frac{360°}{2^4} = 22.5°$. In this example, the sectors have been assigned clear or dark marking using the binary sequence.

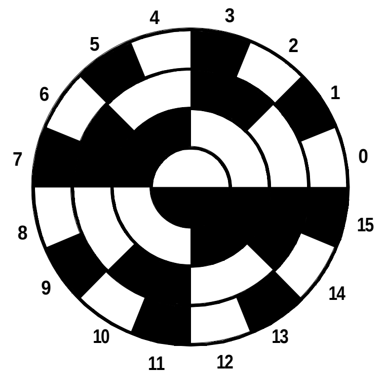


Figure 7.2: Four bit binary code angle encoder.

There is a major difficulty associated with using this binary division of the disk. Consider the output on each side of 0°, say 1° and 359°. The outputs would be 0000 and 1111. Now consider what happens at the transition through 0°. If all of the contacts are not perfectly aligned, they will not all switch state at the same angular position so that there could be an intermediate state so the sequence could be 0000 —> 0100 —> 1111 if the third bit detector is slightly misaligned. This would correspond to a reading apparently going from sector 1 to sector 8 to sector 16 or an indicated jump of angular position of 180°. This apparent angular jump could have disastrous consequences if the angle sensor is used to determine the position of a workpiece on a numerically controlled machine tool or the position of a control surface on a aircraft as the controlling computer would receive a momentarily incorrect reading and would generate an inappropriate corrective action as a result of the error. The cause of this error is that a number of the detected bits are changing simultaneously at the sector change overs.

The use of a sector which is coded using a Gray code (named after Frank Gray, a research physicist at Bell Telephone Laboratories) avoids this problem. In this coding system, only one bit changes state at each border between sectors as is shown in Figure 7.3 and therefore the maximum ambiguity is ± one (outer) sector.

In Figure 7.3, a particular angular position is marked with X and the code for this position is 1101. The most significant bit is coded by the innermost sector. If the angle changes so that the position moves into an adjacent sector it can be seen that only one bit of the position code changes so that there is a smooth progression through the position codes as the sectors rotate with no discontinuous jumps to distant position codes.

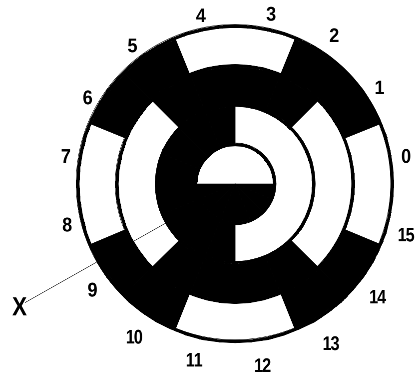

Figure 7.3: Four bit Gray code angle encoder.

This property of code systems whereby only one bit changes as the changes through successive codes is possessed by **cyclic codes**. The Gray code is a particular example of such cyclic codes which also a **reflective code**.The table shows the set of Gray codes for decimal numbers from 0 to 15. Note the reflective symmetry of the Gray codes about the center of the table, if the leading bit is disregarded.

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

This can be seen in the illustration of a linear Gray code encoder system shown in Figure 7.4. The period of the black and white stripes increases by a factor of two at each increase in resolution. Also the stripes are displaced by one period at each increase in resolution.

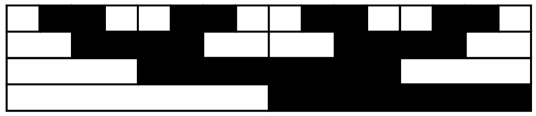

Figure 7.4: Linear Gray code position measuring system.

The relationship between the Gray code and the corresponding binary or decimal equivalent is not immediately apparent. The procedure for converting from binary to Gray is to take the msb (most significant bit) of the binary and write it as the msb of the Gray. Then carry out an eXclusive OR operation on successive bit pairs and place the result bit in the Gray code bit position. Thus to convert decimal 2 which is 010 in binary, the leading bit of the Gray code is 0, the next is 0 XOR 1 = 1 and the last bit is 1 XOR 0 = 1 to give a Gray code of 011. This is illustrated in Figure 7.5 (a).

The reverse conversion from a Gray code to a binary code can be implemented by a similar bitwise eXclusive OR operation. The most significant bit of the Gray code is again entered as the most significant bit of the equivalent binary code. This result bit is then XOR ed with the next most significant bit of the Gray code and the result entered in the next binary bit. The process is repeated until the binary code is filled in. Thus Gray code 110 has a 1 in the most significant bit of the binary. This 1 is XOR ed with the next bit of the Gray code which is a 1 to give 0. This 0 is XOR ed with the last 0 bit of the Gray to give 0 to give 0 and an equivalent binary code of 100. This process is illustrated in Figure 7.5 (b).

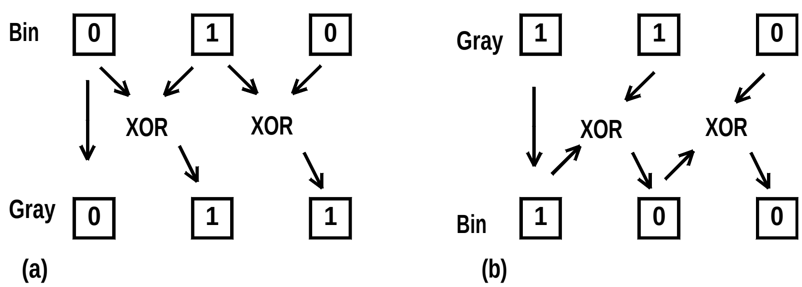


Figure 7.5: .

A circuit which will convert a Gray code to a Binary code is shown in Figure 7.6.

A single bit binary code can be represented by two points on a line as shown in Figure 7.7 (a). A two bit binary code can be represented by the

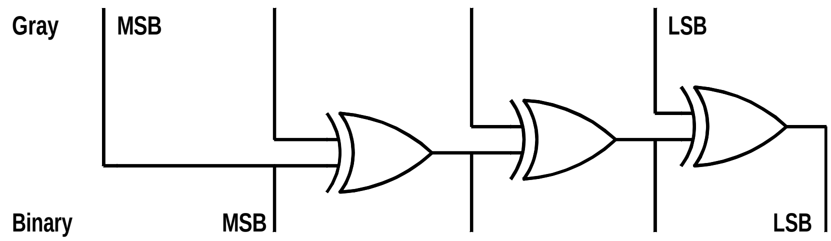


Figure 7.6: A circuit for converting a four bit Gray code to a 4 bit Binary code.

vertices of a square as shown in Figure 7.7 (b). A three bit binary code can be represented by the vertices of a cube as shown in Figure 7.7 (c).

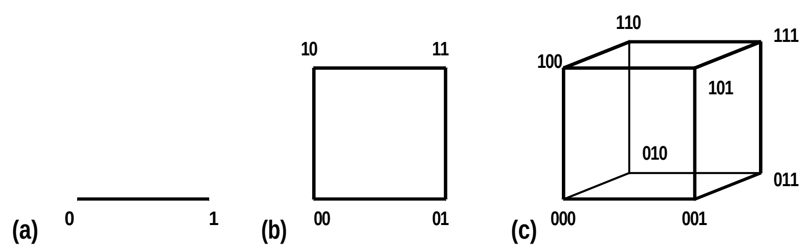


Figure 7.7: Graphs for 1, 2 and 3 bit binary codes.

An $n$ bit binary code can be represented by the vertices of an $n$ dimensional graph (not illustrated). A Gray code sequence is then obtained by moving along the edges of the graph, passing through each and all of the vertices in turn and finally returning to the origin. Such a path is called a Hamiltonian path after the Irish mathematician, William Rowan Hamilton. Strictly speaking, a Gray code sequence is any such path, of which there are many. The convention, however, is that only paths which are Hamiltonian and reflective are called Gray code sequences. This reflective requirement gives us a very quick way of writing down the Gray code sequence which is illustrated in the table. For 1 bit the codes are 0 and 1. For 2 bits, reflect the 1 bit code and put a 0 and a 1 in front of each half. For 3 bits reflect the two bit code and put 0 and 1 in front of each half.

| No. of bits | 1 | 2 | 3 |
|---|---|---|---|
| Codes | 0 | 00 | 000 |
| | 1 | 01 | 001 |
| | | 11 | 011 |
| | | 10 | 010 |
| | | | 110 |
| | | | 111 |
| | | | 101 |
| | | | 100 |

And finally, to provide a plain language illustration of Gray coding, we suggest this example of a cyclic but nonreflective Gray verse.

Gray
Cray
Clay
Clap
Clip
Flip
Flop

## 7.1 References

Floyd, T. L. (1994), *Digital Fundamentals, 5th ed.*, Merrill Maxwell.

Lala, P.K. (1996), *Practical Digital Logic Design and Testing*, Prentice Hall.

Gardner, M. (1986), *Knotted Doughnets and Other Mathematical Entertainments*, W.H.Freeman and Company.

## 7.2 Problems

7.1 Convert the hexadecimal number $31B_H$ to the following formats:
a) BCD 8421
b) 7421
c) 4221
d) $84\overline{21}$

7.2 Design a circuit which will implement the algorithm for converting a Binary code to a Gray code which is shown in Figure 7.5 (a). The circuit should use XOR gates similar to the circuit shown in Figure 7.6.

7.3 A Gray code optical shaft position encoder has a resolution of 1 part in 128. What is the range of each sector in degrees? If the shaft is in position 51, calculate the signals which should be observed on each of the 7 output signal lines.

7.4 A linear position encoder using a track divided using a Gray code system is 1.1 meter long. The encoder is to resolve the position to 1mm. How many divisions must there be on the track segment with the most divisions?

7.5 How many Hamiltonian paths are there around the 3 dimensional graph (cube) shown in Figure 7.7? How many of these Hamiltonian paths will also be reflective? Obtain an expression for the number of possible $n$ bit Gray codes as a function of $n$. (Warning: The general solution to this problem is not known—yet! For 4 bits, it is 48384; for 5 bits, it is $\approx 5.8 \times 10^{10}$; for 6 bits, it is $\approx 2.4 \times 10^{25}$.)