ششمین همایش بین‌المللی افق های نویـن در
مهندسی برق، کامپیوتـر و مکانیک
6th International Conference on the New Horizons in Electrical Engineering, Computer and Mechanical
www.mhconf.ir

New Model For Implementing Real Time Binary to Gray 4-bit Decoder Based On Artificial Neural Network

Arian Taherkhani
School of Nuclear Energy, Ghazvin, Iran
aryantaherkhani2184@gmail.com

Seyed Amidedin Mousavi
Department of Electrical Engineering
Zanjan Branch, Islamic Azad University
Zanjan, Iran
a_mousavi@sbu.ac.ir

Abstract — **Nowadays, various codes are used to enhance the security of computer networks, to make it easier to work with binary codes for ordinary people, and also to facilitate the detection and correction of errors in sending and receiving information. One of the most important codes used in shaft encoder sensors is Gray code. This code is used to determine the exact angle of rotational motion. Previously, these sensors used binary code. During using these codes in encoder shafts several important faults occurred. For solving these problems scientists suggested that using gray code would be efficient. Various techniques are available to convert binary code to gray code. In this paper, an approach based on Artificial Neural Networks (ANN) is been used to convert binary code to gray. Back propagation Algorithm for feed forward ANN has been simulated using MATLAB for converting these codes. The designed ANN is trained for all possible combination of code.**

Keywords- Artificial Neural Network; gray code; binary code; real-time;decoder

---

1. Introduction

Scientists in digital systems use different codes. The most important reasons for using different codes in digital systems have been mentioned below [1]:

- Security in computer systems
- Make the binary code users friendly for ordinary peoples.
- error detection

Important codes that used in computer science are as follows [1,2]:

- BCD code: this code is used to make binary code more users friendly.
- Excess-3 code: this code is used to simplify mathematical calculations in the digital computer.
- ASCII code: this code is used in personal computers to indentify keyboard's characters to the CPU.
- Gray code: this code is used in electronic sensors to determine the position and displacement of a rotational motion.
- Error detection codes: these codes such as; biquinary, parity code and Hamming cod are used to detect and correct errors during data transmission.

2. Theory

2.1. Basic theory

Binary codes are codes which condescend the positional weight principle. Each position of the number demonstrates a specific weight which have shown in the Figure1 [1-3].

weight	…	128	64	32	16	8	4	2	1
digits	…	1	1	1	1	1	1	1	1

Figure1. Binary code as a weighted code

As shown in Figure 2 and by using the positional weight principle, users can convert decimal numbers to the binary and vice versa.

$$(1101101)_2 = 128 + 64 + 16 + 8 + 1 = (217)_{10}$$

$$(59)_{10} = 32 + 16 + 8 + 2 + 1 = (111011)_2$$

Figure 2. Decimal numbers to the Binary code converting

2.2. BCD Code

In this code, for each digit of decimal number four bit area has been considered. Then the binary equivalent of each digit is placed in these spaces. Please note the Figure 3 [4].

$$(357)_{10} = (0011\ \ 0101\ \ 0111)_{\mathrm{BCD}}$$

$$(648)_{10} = (0110\ \ 0100\ 1000)_{\mathrm{BCD}}$$

Figure3. Decimal to binary Converting

2.3. Gray code

In this code, for each digit of decimal number four bit area has been considered [3]. Then the equivalent code of each digit is placed in these spaces according to table 1.

Table1. Gray code and comparing with Binary code

Decimal digit	Gray code	Binary code
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001

Please pay attention to figure.4.

$$(452)_{10} = (0110\ \ 0111\ \ 0011)_{\text{Gray}}$$
$$(764)_{10} = (0100\ \ 0101\ 0110)_{\text{Gray}}$$

Fig.4. Decimal to binary Converting

3. Design of Binary to Gray Decoder Hardware

The As it is mentioned in the previous section, it is relatively difficult to use the table method to convert decimal numbers into gray codes. To this end, we have been trying to design a 4-bit gray code converter to BCD. The design consists of four main parts:

- Describing the problem in the correct table and determining inputs and output functions.
- Expression of output functions in the standard form.
- Simplifying output functions
- Implementation of the designed hardware.

The description of the problem is presented in Fig. 5. In this figure, each BCD number is converted to the Gray code, but since the BCD digits are up to the 9, the output that related to 10 to 15 is considered to be don't care. These states are shown in the table with the X symbol.

BCD				Gray			
W	X	Y	Z	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Figure 5. Truth table that related to BCD to gray converting

In the correct table, the output functions are represented by the symbols F0, F1, F2 and F3, respectively, and should be written in the standard form as the sum of minterms. The standard form of these functions is given in relations (1) to (4).

$$\begin{cases} F_0 = \sum m(1,2,5,6,9) \\ d = \sum m(10,11,12,13,14,15) \end{cases} \quad (1)$$

$$\begin{cases} F_1 = \sum m(2,3,4,5) \\ d = \sum m(10,11,12,13,14,15) \end{cases} \tag{2}$$

$$\begin{cases} F_2 = \sum m(4,5,6,7,8,9) \\ d = \sum m(10,11,12,13,14,15) \end{cases} \tag{3}$$

$$\begin{cases} F_3 = \sum m(8,9) \\ d = \sum m(10,11,12,13,14,15) \end{cases} \tag{4}$$

To achieve optimal design, the output functions should be simplified with Karnaugh map. The simplification procedure is shown in Figure 6.

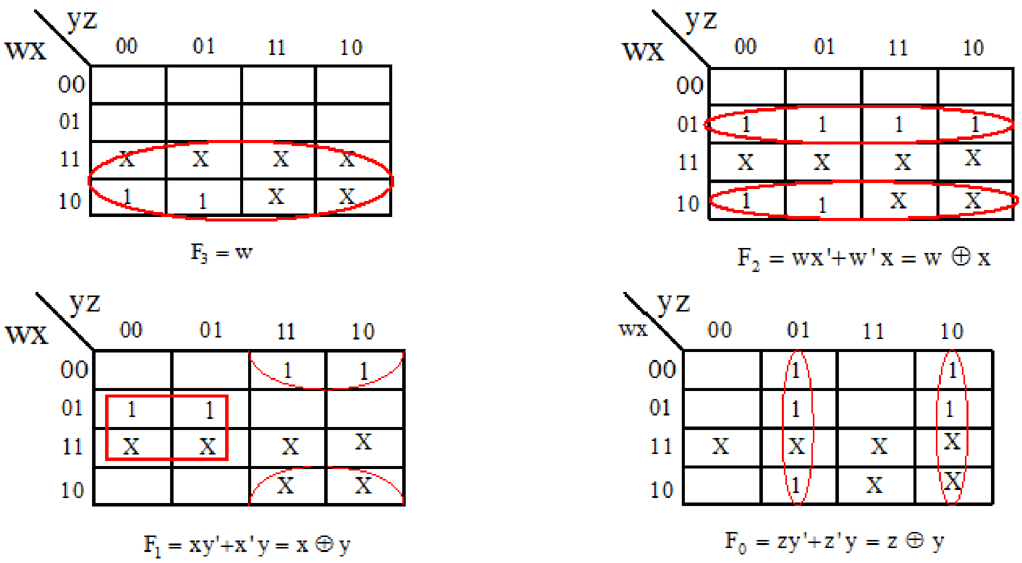

Fig.6. Simplify output functions with Karnaugh map

After simplification, the hardware-logic diagram of the designed circuit must be drawn up. This diagram is shown in Figure 7.

Simplification, the hardware-logic diagram of the designed circuit must be drawn up. This diagram is shown in Figure

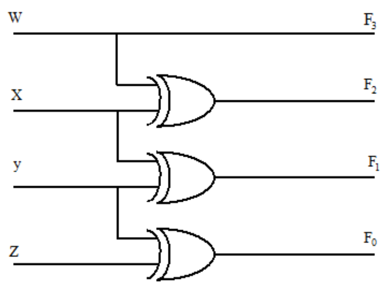

Figure7. 4-bit converter circuit for converting BCD to gray code

By interpreting the hardware circuit, a simpler pattern can be obtained for converting decimals into gray code. Figure 7 shows that to convert the BCD code to the gray code it is enough to hold the most significant bit, and to get the next bits in the BCD code, we must execute XOR operation on another digits from left to right. So, it's suitable to convert decimal numbers into BCDs and then convert decimal numbers to gray code.

4. Artificial Neural Network Model

In this section, an artificial neural network is designed and structured to convert the binary code to Gray code. The 4-bit binary codes are clustered into input and 4-Bit Gray codes clustered into target. The structure of input and target of the ANNs is as follows [5]:

- The variable predicted by ANN: each bit of 4-Bit Gray codes.
- Inputs: 4-Bit Binary Codes
- Target: 4-Bit Gray Codes

Twenty-five percent of the data are used for validation, Twenty-five percent of the data are used for testing and fifty percent of the data are used for the training. Multilayer Perceptron Artificial Neural Networks (MLP_ANNs) are set up to predict the 4-Bit Gray Codes. The structure of MLP-ANNs is obtained by minimizing the MSE performance. The structure of the MLP-ANN is as follows:

- The variable predicted by ANN: 4-bit Gray Code.
- Number of inputs: 48
- Number of outputs: 48
- Number of hidden layers: 3
- Number of neurons in hidden layers: Layer1=Layer2=Layer3=5

Figures 8-9 illustrate the execution of MLP-ANN in terms of MSE and coefficient of correlation (R) for predicting the 4-bit Gray codes.

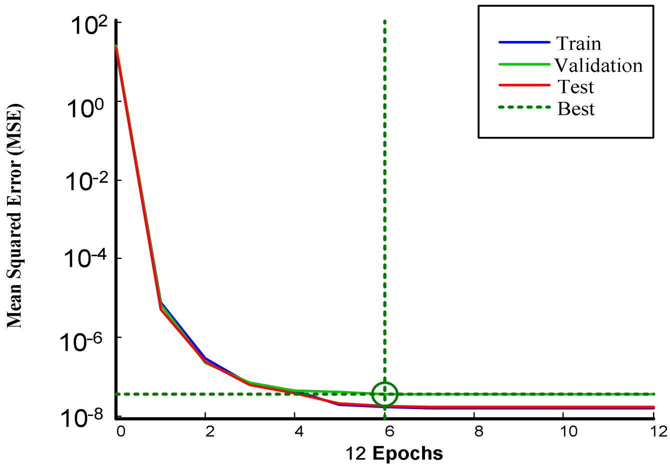

Figure 12. Performance of MLP-ANN in terms of MSE for predicting the 4-Bit Gray Codes.

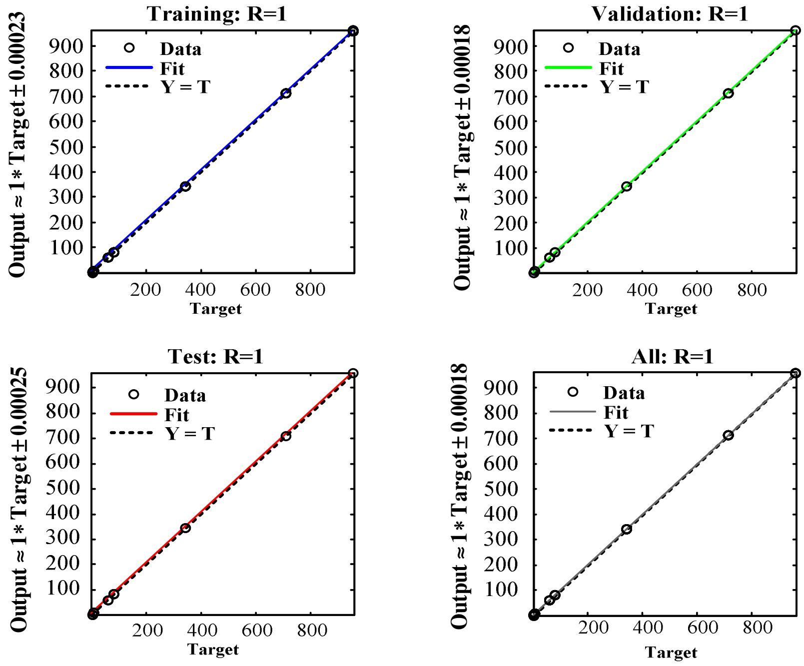

Figure 8. The results of regression for predicting the 4-bit Gray codes.

The results of simulation by using ANN model have been illustrative in Table2.

Table 2. Gray codes Prediction by using ANN from Binary codes

Decimal	Binary	Real Gray Codes				Simulated Gray Codes			
		G3	G2	G1	G0	G3	G2	G1	G0
0	0000	0	0	0	0	0	0	0	0
1	0001	0	0	0	1	2.3e-27	0.0001	1.4e-12	0.9999
2	0010	0	0	1	1	2.3e-27	2.3e-27	0.9887	0.9779
3	0011	0	0	1	0	2.3e-27	2.3e-27	0.9888	2.3e-27
4	0100	0	1	1	0	2.3e-27	0.9888	1	2.3e-27
5	0101	0	1	1	1	2.3e-27	0.9999	0.9999	0.9988
6	0110	0	1	0	1	2.3e-27	1	2.3e-27	1
7	0111	0	1	0	0	2.3e-27	0.9888	2.3e-27	2.3e-27
8	1000	1	1	0	0	1	1	2.3e-27	2.3e-27
9	1001	1	1	0	1	0.9998	0.988	2.3e-27	0.988
10	1010	1	1	1	1	1	0.9999	0.9998	1
11	1011	1	1	1	0	1	0.9999	0.9998	2.3e-27
12	1100	1	0	1	0	0.9999	2.3e-27	0.9999	2.3e-27
13	1101	1	0	1	1	1	2.3e-27	1	1
14	1110	1	0	0	1	0.9999	2.3e-27	2.3e-27	1
15	1111	1	0	0	0	1	2.3e-27	2.3e-27	2.3e-27

In this paper, an ANN based Binary to gray codes decoder is presented. A Multi-layer perceptron ANN is selected for training. The selection of the number of input layer neurons and output layer neurons required can be exactly specified because they depend upon the codes word. The ANN is trained using back propagation algorithm such that it is able to detect and correct 1-bit error. As the number of hidden layer neurons increases, the convergence time for the error to fall below a threshold decreases. Also, as the learning rate increases, the convergence time for the error to fall below a threshold decreases. After the training, the designed ANN decoder is tested.

References

[1] Mousavi, Seyed Amidedin and Moghimi, Nikta and Ghafri, Maedeh,1395,The Application Of Boolean Algebra In Design Of Digital Circuits; A Case Study Of Sample Circuit, The First National Conference on Modelling Mathematics and Static in Applied Studies,Chalus,https://civilica.com/doc/750578

[2] Nouhi, Shima and Mousavi, Seyed Amidedin, 1398, Gray code and shaft encoder design based on it, Fifth National Conference on Electrical and Mechatronics Engineering of Iran,Tehran,https://civilica.com/doc/988540

[3] Mansouri, Arnika and Mousavi, Seyed Amidedin,1398,Intelligence Error Detection Code in Digital Data Transmission Systems, Fifth National Conference on Electrical and Mechatronics Engineering of Iran,Tehran,https://civilica.com/doc/988541.

[4] Seyed Amidedin Mousavi, Shima Nouhi, Fateme Sadat Hoseini,"Gray Code and its Application in Rotary Encoder Sensors", Electrical Machines Letters, Vol.1, Iss.1, 2019.

[5] Seyed Amidedin Mousavi, Mostafa Sedighzadeh, Arsalan Hekmati, Mehdi Bigdeli, "Artificial neural network based method for temperature correction in FDS measurement of transformer insulation", Journal of Physics D: Applied Physics, Vol. 53, Iss.14., IOP publishing, 2020.