DESIGN AND ANALYSIS OF A NEWLY INNOVATED SET OF BINARY CODE-WORDS OF UNITARY HAMMING DISTANCE

Afaq Ahmad and Ahmed Al-Maashari
Department of Electrical & Computer Engineering
College of Engineering, Sultan Qaboos University
P.O. Box 33 Al-Khod, Muscat
Sultanate of Oman 123
Téléphones: + 968 24141327, + 968 24142572 Fax: + 968 24413454
E-mails: afaq@squ.edu.om, maashari@squ.edu.om

Abstract:

A newly innovated set of binary code-words of unitary Hamming distance is proposed. The design of generating the code using VHDL, hardware and algorithmic implementation procedures are embodied in this paper. Further, a comparison study is carried-out to demonstrate the edge-over improvement and added advantages over the binary Gray code-words.

Keywords:

Hamming distance, binary code-words, Gray code-words, Transition counts, IC 74LS175

1. Introduction

The binary code-words with unitary Hamming distance has special place in those situations where sharp transitioning is required. Also, higher speed, minimum power dissipation, and reduction of noise spikes are the other added advantages of such codes. Furthermore, these types of code are also relatively free from errors. One of such an important binary coding scheme is the Gray coding [1]. Gray coding is a simple binary coding of with place values *8-4-2-1* for BCD then counting from *7 (0111)* to *8 (1000)* requires *4* bits to be changed simultaneously. If this does not happen then various numbers could be momentarily generated during the transition so creating spurious numbers which could be read wrongly. The Gray coding avoids this since only one bit changes between subsequent numbers. And, therefore, Gray codes are extensively used in several kinds of applications. Some of the examples are like solving puzzles such as the Tower of Hanoi and the Brain, analog-digital-converters (goniometers), Hamiltonian circuits in hyper-cubes and Cayley graphs of Coxeter groups, capanology (the study of bell-ringing), continuous space-filling curves, classification of Venn diagrams, design of communication codes, enhancements of the resolution of spectrometer, in Genetic Algorithms, in labelling the axes of Karnaugh maps, and digital system testing [2] – [8].

Some industrial automation applications require control systems which know the rotational position of a shaft. Similar devices are also used for digital volume controls, etc, on domestic appliances. A shaft encoder is used to record the rotation typically in the form of a binary Gray code. One of a common use for the Gray-code is in positioning or angle-encoding. The use is because the position (rotation) error due to misalignment is at most one bit. There is no such guarantee for a binary coded sensor. Figure 1 [6] shows such an application where a *5*-bit rotation encoder is designed, using a Gray-code pattern in front of an optical sensor [6].

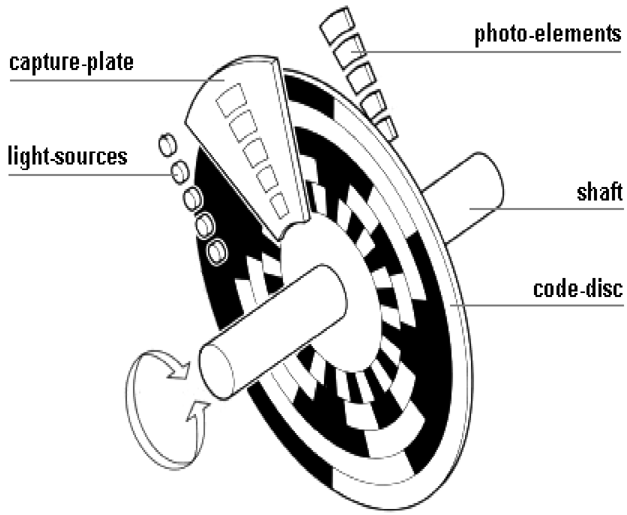

Figure 1: A 5-bit disc rotation encoder [6]

2. Our Investigation

Maintaining the unitary Hamming distance between each of the predecessor and successor binary code-words *C(i)* and *C(i+1)* means differing at exactly only one bit position. For example let's consider a *2*-bit size such binary code-words *[C(1), C(2), C(3), C(4)]* are *[00, 01, 11, 10]*. The Gray code which is weighted and cyclic in nature came into existence at the time when digital logic circuits were built from vacuum tubes and electromechanical relays. The time at which counters dissipating enormous power and generating noise spikes when many bits changed at once. Using binary code-words with unitary Hamming distance (Gray code counter), where the binary counter increment or decrement in the value changes only one bit, regardless of the size of the number, thereby minimizing the effect of noise and power dissipation as well. However, we further investigated the property of Gray codes and found that the code-words it self can be improved while maintaining the unitary Hamming distance between the predecessor and successor code words. Our proposed code-words have constant transition counts, reducing the power dissipation and also cutting short the noise spikes.

3. Proposed Designs

Through analyzing the bit-patterns of Gray code we realized that why not to write the code-words in the following patterns to minimize the transition while retaining the property of unitary Hamming distance. Let's consider the bit size of the code-word as *n* then the proposed pattern generation scheme can be expressed in the form of a theorem (Theorem 1) as given below.

Theorem 1:

For an *n*-bit size of code there can be 2n binary code-words where each of the predecessor and successor binary code-words has unitary Hamming distance.

Figure 2 below constructs the code-word vector *C* which will be of size *2n x n*. The code-word pattern can be represented by the each row vector of the following matrix.

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & . & . & . & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & . & . & . & 1 & 1 & 1 \\ . & . & . & . & . & . & . & . & . & . \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{2nxn}$$

Figure 2: n-bit size code-words generation matrix

In general, to script all the *2n* code-words, the following equations are suggested.

The i^{th} code-word (where, i varies from *1* to *n*) scripting is governed by Equation (1) whereas; to script the rest (*n-1*) code-words Equation (2) is to be used. The Equations (1) and (2) are as given below.

$(2^{i-1} - 1)_{base2}$; where i varies from *1* to *n*. (1)

$(2^{n} - 2^{j})_{base2}$; where j varies from *1* to *n-1*. (2)

Example 1:

Let's consider *n* = 4, then the first five code words can be written using the Equation (1) by inserting the subsequent values of *i* from *1, 2, 3, 4,* and *5*.

$C(1) = [0\ 0\ 0\ 0]_{1x4}$
$C(2) = [0\ 0\ 0\ 1]_{1x4}$
$C(3) = [0\ 0\ 1\ 1]_{1x4}$
$C(4) = [0\ 1\ 1\ 1]_{1x4}$
$C(5) = [1\ 1\ 1\ 1]_{1x4}$

Using Equation (2) and subsequently inserting values of *j* from *1, 2,* and *3* we get

$C(6) = [1\ 1\ 1\ 0]_{1x4}$
$C(7) = [1\ 1\ 0\ 0]_{1x4}$
$C(8) = [1\ 0\ 0\ 0]_{1x4}$

Simulation and Hardware Realization:

The implementation of the proposed code generation can be done either using the simulation or the hardware. The following VHDL code (see Figure 3) simulates the *n*-bit code of the proposed scheme.

The VHDL code:

```
entity My_Counter_n is
  port(clock:in bit;   q:out bit_vector(n downto 1));
end;

architecture RTL of My_Counter_n is
  signal My_Counter: bit_vector(n downto 1);
begin
  process(clock)
  begin
    if clock'EVENT and clock='1' then
      My_Counter (1) <= not(My_Counter (n));
      LFSR(n downto 2) <= My_Counter (n-1 downto 1);
    end if;
    q <= My_Counter;
  end process;
end;
```

Figure 3: VHDL script for generating *n*-bit size code-words

The proposed hardware implementation for generating the code is much simpler than the Gray code generation schemes. To illustrate the design of a *4*-bit code generator is considered. A quad D type positive-edge-triggered flip-flops chip *74LS175* is used which has propagation delay of *20ns* (see Figure 4). The first code bits $[b_4\ b_3\ b_2\ b_1]$ are generated by applying the active low signal at pin 1. For rest of the code-words generation the pin 1 should be kept at logic *1* (V_{cc}). Using the clocks the outputs from pins *15, 10, 7*, and *2* will be available b_4, b_3, b_2, and b_1 respectively.

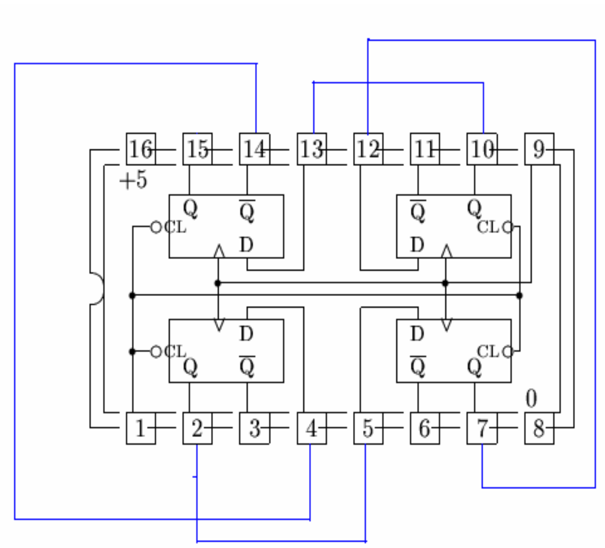

Figure 4: Hardware design of proposed code-words generator *4*-bit size

4. Analysis of the Proposed Code-words

We simulated the binary Gray-code words and the proposed binary code-words for large values of n. The Tables 1 and 2 show respectively the binary Gray code-words and proposed binary code-words for n = 2 to 6.

Transition count is one of the measures which reflect the noise spike, power dissipation and the propagation delay in the code-word signal itself. The Transition count is referred to number of times a signal changes from 0 to 1 or from 1 to 0 during the propagation of the signal.

For a sequence vector, R = $[r_1, r_2, \ldots, r_m]$, the associated Transition count can be given by

$$TC(R) = \sum_{i=1}^{m-1} (r_i \oplus r_{i-1}) \tag{3}$$

Where $\oplus$ denotes modulo-2 addition.

Since, the Transition detector needs a counter with $[\log_2 m]$ stages, and thus the maximum transition count can be *(n-1)*. The Transition counts using the Equation (3) are also computed and shown in Tables 1 and 2 for the binary Gray code-words and proposed binary code-words respectively.

Table 1: Binary Gray code-words (*n = 2:6*)

S.no.	6-bit		5-bit		4-bit		3-bit		2-bit	
	Code	TC	Code	TC	Code	TC	Code	TC	Code	TC
1	001011	3	00010	2	0000	0	000	0	00	0
2	001001	3	00110	2	0001	1	001	1	01	1
3	001000	2	00111	1	0011	1	011	1	11	0
4	011000	2	00101	3	0010	2	010	2	10	1
5	011001	3	00100	2	0110	2	110	1		
6	011011	3	01100	2	0111	1	111	1		
7	011010	4	01101	3	0101	3				
8	011110	2	01111	1	0100	2				
9	011111	1	01110	2						
10	011101	3	01010	4						
11	011100	2								
12	010100	5								

Table 2: Proposed binary code-words (*n = 2:6*)

S.no.	6-bit		5-bit		4-bit		3-bit		2-bit	
	Code	TC	Code	TC	Code	TC	Code	TC	Code	TC
1	000000	1	00000	1	0000	1	000	1	00	1
2	000001	1	00001	1	0001	1	001	1	01	1
3	000011	1	00011	1	0011	1	011	1	11	1
4	000111	1	00111	1	0111	1	111	1	10	1
5	001111	1	01111	1	1111	1	110	1		
6	011111	1	11111	1	1110	1	100	1		
7	111111	1	11110	1	1100	1				
8	111110	1	11100	1	1000	1				
9	111100	1	11000	1						
10	111000	1	10000	1						
11	110000	1								
12	100000	1								

The Transition counts variations of both types of the code-words are shown in Figure 5.

5. Result:

To prevent multiple bits transitioning simultaneously, one can use a gray code to ensure that only a single bit transitions at a time. In the physical world, there is no way to ensure that all the bits will transition at exactly the same time, so a system may actually pass through a sequence of *000, 001, 011,……*

To demonstrate the practical approach of our proposed binary code-words we consider an example of an angular rotational disk of *60°* slots. Therefore, six slots are required to rotate completely to *360°*. Thus, $n = 3$ code generation scheme will be required to implement the system. On some parts of a track on a disk have metal contacts (contacts *1, 2, and 3*), corresponding to a *"1"* (*on*), and other parts have insulator, corresponding to a "*0*" (*off*). The switching transitioning of the proposed code-words and the binary Gray code-words are shown in Tables 3 and 4 respectively. It can be seen from the Tables 4 and 5 that the proposed scheme has better transitions in comparison to binary Gray codes while switching from sector *4* to *5*. The proposed scheme switches from ***on → on → on*** to ***on → on → off*** while the Gray code switches from ***off → on → off*** to ***on → on → off***.

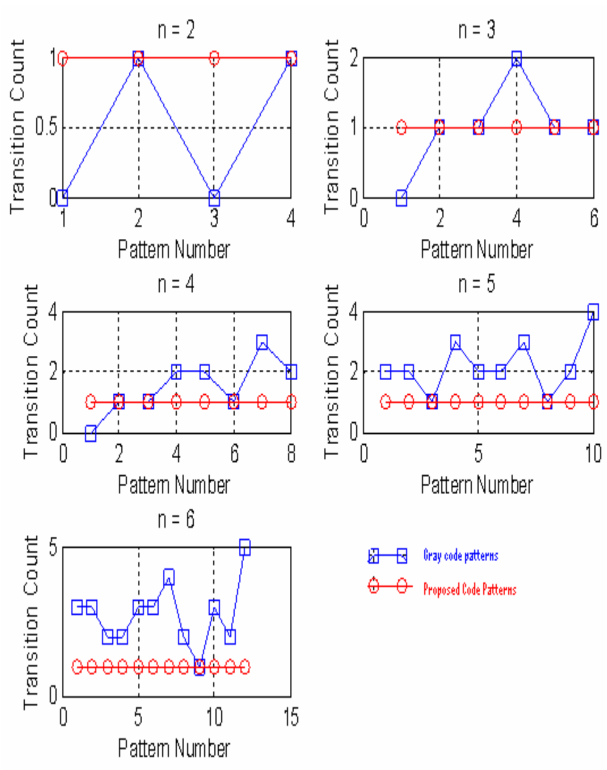

Figure 5: Transition counts variation graph

Table 3: A *60°* angular rotational scheme using the proposed binary code-words

Proposed binary code-words				
Sector	**Contact 1**	**Contact 2**	**Contact 3**	**Angle**
1	off	off	off	0° to 60°
2	off	off	on	60° to 120°
3	off	on	on	120° to 180°
4	on	on	on	180° to 240°
5	on	on	off	240° to 300°
6	on	off	off	300° to 360°

Table 4: A *60°* angular rotational scheme using binary Gray code-words

Binary Gray code-words				
Sector	**Contact 1**	**Contact 2**	**Contact 3**	**Angle**
1	off	off	off	0° to 60°
2	off	off	on	60° to 120°
3	off	on	on	120° to 180°
4	off	on	off	180° to 240°
5	on	on	off	240° to 300°
6	on	on	on	300° to 360°

6. Conclusion

A novel technique for generating a subset of binary code-words with unitary Hamming distance is proposed. The generated code-words have better transitioning than the binary Gray code-words. The proposed scheme requires less hardware and reduces the propagation delays because it does not require binary code converter stage. Thus, it's a cost and time effective scheme of generating the code-words of unitary Hamming distance. The propagation delays further reduces due to the better transitioning attributes of the proposed codes. Also, less power dissipation and noise spike cuts are provided due to the constant transitioning.

7. References

[1] Gray, F. 'Pulse Code Communication'. United States Patent Number 2,632,058, March 17, 1953.
[2] Gilbert, E.N. 'Gray Codes and Paths on the n-Cube'. Bell System Tech. Journal, 37 (1958), pp. 815-826.
[3] F. Ruskey, 'A Survey of Venn Diagrams', Elec. J. of Comb. (1997).
[4] Goddyn, L., Gvozdjak, P. 'Binary gray codes with long bit runs', Electron. Journal Combin 10 (2003), pp. 1-10.
[5] Moshe Schwartz and Tuvi Etzion, 'The Structure of Single-Track Gray Codes', IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 45, NO. 7, NOVEMBER 1999, pp. 2383 - 2396
[6] http://tams-www.informatik.uni-hamburg.de/ applets/ hades/webdemos/10-gates/15-graycode/ dual2gray.html (December, 2006)
[7] http://computingdictionary.thefreedictionary.com/gray+code (November, 2006)
[8] http://www.theory.csc.uvic.ca/~cos/inf/comb/SubsetInfo.html. (November, 2006)