

Codificador Binário para Código Gray

Versão 2

Profs. Bruno Albertini e Edson Gomi

1 Objetivo do Projeto

Este projeto tem como objetivo exercitar e avaliar o conhecimento para desenvolver projetos de circuitos combinatórios, especificamente com o uso de Mapa de Karnaugh e implementando em linguagem Verilog.

2 Breve descrição do Código Gray

Não se sabe a origem dos códigos Gray, mas há relatos de suas aplicações desde o Século XIX [1]. Mas esta classe de códigos recebeu este nome em homenagem ao pesquisador Frank Gray, que criou e descreveu um método sistemático para a construção de códigos binários refletidos (*Binary Reflected Gray Code*) na patente US 2,632,058 [2], que foi submetida ao US PTO em 13 de novembro de 1947. Na época, o Frank Gray era pesquisador do Bell Laboratories. A patente não reivindica o código Gray, mas protege a invenção de um método de modulação digital para TV a cores.

A característica essencial de um código Gray é que a diferença entre duas palavras consecutivas é de apenas um bit. Por exemplo, no código Gray de 3 bits formado pelas palavras 000, 001, 011, 010, 110, 111, 101 e 100, ocorre mudança em apenas 1 bit entre duas palavras consecutivas. O mesmo ocorre entre a última e a primeira palavra da sequência, o que permite que o código seja usado de forma cíclica. Esta é uma característica interessante em aplicações de contagem. Um aspecto importante a ressaltar é que, para existem mais de um código Gray para um código de mesmo comprimento. Por exemplo, o código 011, 111, 101, 001, 010, 110, 100 e 000 também é um código Gray de 3 bits [3].

A construção de um código Gray binário refletido ocorre da seguinte maneira:

- Comece com o caso base, de 1 bit:

0 1

- Acrescente bits 0 e 1 na posição anterior (préfixa) ou posterior (pósfixa). Neste exemplo acrescentaremos na posição anterior, formando dois grupos de palavras:

00 01 | 10 11

No passo seguinte, refletimos o segundo grupo, para formar um código Gray de 2 bits:

00 01 11 10

- Agora acrescentamos novo bit em posição anterior:

000 001 011 010 | 100 101 111 110

- E refletimos o segundo grupo, formando um código Gray de 3 bits:

000 001 011 010 110 111 101 100

- Finalmente podemos formar um código Gray de 4 bits:

0000 0001 0011 0010 0110 0111 0101 0100 |
1000 1001 1011 1010 1110 1111 1101 1100

- Refletindo o segundo grupo, temos:

0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000

Códigos Gray são usados em codificadores de posição, em códigos de correção de erros, em comunicação entre circuitos digitais com domínios de clock diferentes, e também nos Mapas de Karnaugh.

Para saber mais sobre o código Gray, leia a seção 2.11 do livro do Wakerly [4].

3 Atividades a realizar

Este exercício tem duas atividades:

1. Projete um circuito combinatório que converte um inteiro sem sinal de 4 bits para a representação em Código Gray, conforme apresentado na Tabela 1. Utilize o método de minimização do Mapa de Karnaugh e implemente a solução contendo os implicantes primos essenciais. Desenhe um diagrama lógico correspondente à solução projetada.

Descreva o Mapa de Karnaugh, a solução com os implicantes primos essenciais e o diagrama lógico num arquivo PDF. Não se esqueça de colocar o seu nome e o seu número USP no cabeçalho do documento.

Submeta na Tarefa correspondente no Moodle o arquivo PDF e o arquivo Verilog da solução do item a seguir.

2. Descreva o circuito projetado em Verilog. Espera-se que o circuito em Verilog corresponda ao projeto feito no item anterior. O módulo do circuito a ser submetido ao Juiz deverá ter a seguinte declaração:

Binário (Nbin)	Gray (Ngray)
0000	1000
0001	0000
0010	0001
0011	0011
0100	0010
0101	0110
0110	0111
0111	0101
1000	0100
1001	1100
1010	1101
1011	1111
1100	1110
1101	1010
1110	1011
1111	1001

Tabela 1: Tabela Verdade do circuito a ser projetado

```
module grayenc (
    input [3:0] Nbin,
    output [3:0] Ngray
);
```

onde Nbin é a palavra binária a ser convertida e Ngray é a palavra código Gray resultante.

O Juiz está programado para permitir no máximo 5 tentativas. Assim, é altamente recomendado que preparem um testbench para testar o circuito projetado antes de submetê-lo ao Juiz. Para aprender a preparar testbenches, recomendamos testar os exemplos da Seção 5.13 do livro do Wakerly [4].

4 Referências

- [1] Wikipedia, “Gray code,” Online. [Online]. Available: https://en.wikipedia.org/wiki/Gray_code
- [2] F. Gray, “Pulse code communication,” Mar. 1953. [Online]. Available: <https://patents.google.com/patent/US2632058A/en>
- [3] R. W. Doran, “The gray code,” *J. Univers. Comput. Sci.*, vol. 13, pp. 1573–1597, 2007. [Online]. Available: <https://api.semanticscholar.org/CorpusID:1770171>
- [4] J. F. Wakerly, *Digital Design: Principles and Practices (5th Edition)*, 5th ed. Pearson, 2017.