



Circuitos Digitais (116351) - 5º Experimento

CIRCUITOS COMBINACIONAIS: CONVERSÃO DE CÓDIGOS

OBJETIVO: Estudo e projeto de conversores de código. O código de entrada utilizado como exemplo é o BCD e o de saída é um código de 5 *bits*. São apresentados, ainda, os códigos de Gray, excesso-3, biquinário e outros.

1. INTRODUÇÃO TEÓRICA

1.1. GENERALIDADES

O acoplamento de dois sistemas digitais entre si requer, na maioria dos casos, um sistema de interligação. A necessidade desse elemento acoplador é consequência do fato de que os sistemas a serem acoplados não utilizam os mesmos códigos. Portanto, a função básica do sistema de interligação é promover a conversão de códigos. Assim, fica possível a troca de informações entre sistemas digitais diferentes.

Consideremos, por exemplo, o “diálogo” entre um computador e uma impressora. Tal “diálogo” é realizado através de um sistema de interligação conhecido como **interface**. Além de sincronizar a velocidade com que é efetuada a troca de informações, a conversão de um código para outro é muitas vezes feita pela interface.

1.2. DESCRIÇÃO DO SISTEMA CONVERSOR

Considere-se que a informação contida numa determinada palavra binária X seja representada por $X = X(Z_1, Z_2, \dots, Z_n)$ em um código C_1 e por $X = X(Y_1, Y_2, \dots, Y_m)$ num outro código C_2 , onde, no caso geral, $n \neq m$.

O conversor de códigos recebe na sua entrada a palavra X no código C_1 e tem a incumbência de traduzi-la para o código C_2 .



Figura 1 – Conversor de códigos

A **Figura 1** mostra o diagrama de blocos de um conversor de códigos. Note que a entrada do conversor é um vetor de dimensão $(n \times 1)$ e a saída um vetor $(m \times 1)$.

É praxe reservar as palavras **codificador** e **decodificador** para os casos limites de conversores de código em que se tem na entrada ou saída do conversor, respectivamente, um código em que é ativado uma e somente uma linha de entrada ou saída por vez (em particular o código decimal).



Figura 2 – Diagrama de blocos de um codificador



Figura 3 – Diagrama de blocos de um decodificador

A **Tabela I** dá alguns exemplos de representação de números decimais em códigos bastante usados na prática.

Tabela I – Alguns códigos mais frequentes

Decimal	8 4 2 1 (BCD)	2 4 2 1	Excesso-3	2 em 5 (2 out of 5)	Biquinário
0	0000	0000	0011	00011	01 00001
1	0001	0001	0100	00101	01 00010
2	0010	0010	0101	00110	01 00100
3	0011	0011	0110	01001	01 01000
4	0100	0100	0111	01010	01 10000
5	0101	1011	1000	01100	10 00001
6	0110	1100	1001	10001	10 00010
7	0111	1101	1010	10010	10 00100
8	1000	1110	1011	10100	10 01000
9	1001	1111	1100	11000	10 10000

Os códigos das duas primeiras colunas da **Tabela I** são chamados códigos ponderados, uma vez que a cada *bit* é atribuído um peso. O código da primeira coluna é o mais comum. É chamado código 8421 ou BCD (*Binary Coded Decimal*). O valor do número decimal representado é calculado segundo a fórmula:

$$N = 8B_3 + 4B_2 + 2B_1 + B_0,$$

onde B_3 , B_2 , B_1 e B_0 são os valores dos *bits* (0 ou 1) nas posições 8, 4, 2 e 1, respectivamente. Os pesos dos 4 *bits* do código da segunda coluna são 2, 4, 2 e 1; este código tem a propriedade de que os números complementos de 9, isto é, que somam 9, têm como códigos correspondentes os 0's trocados pelos 1's, e vice-versa, entre si.

O código excesso-3 da terceira coluna não é ponderado, mas possui a mesma propriedade do código 2421 da segunda coluna. O código biquinário é derivado do ábaco. Ele também é ponderado e os pesos de seus *bits* valem 5043210. Foi usado no computador IBM 1620.

No código de Gray, ou de mudança mínima, os dígitos são obtidos por reflexões sucessivas, sendo por isso também chamado de código binário refletido. Esse código é muito útil na codificação de eixos rotativos, pois minimiza o erro devido ao não alinhamento das escovas ou fotocélulas para leitura (veja a **Tabela II**).

Dado um código de Gray de n bits ($G_{n-1}G_{n-2}...G_1G_0$) sua conversão para o código binário puro ($B_{n-1}B_{n-2}...B_1B_0$) é feito observando-se que cada *bit* foi invertido o número de vezes igual ao número de 1's a sua esquerda. Portanto, se o número de 1's à esquerda de um *bit* for ímpar ele deve ser reinvertido; caso contrário não o será. Em outras palavras, temos:

$$B_k = G_k \oplus G_{k+1} \oplus \dots \oplus G_{n-1}, k = 0, 1, \dots, n - 1$$

Por outro lado, dado um código binário puro ($B_{n-1}B_{n-2}...B_1B_0$), sua conversão para o código de Gray correspondente ($G_{n-1}G_{n-2}...G_1G_0$) é feita observando-se a lei de formação:

$$G_k = B_k \oplus B_{k+1}, k = 0, 1, \dots, n - 1$$

Em ambos os casos, sempre se considera $B_n = 0$.

Observa-se ainda que se pode construir o código de Gray a partir do código excesso-3 para se obter um código com o mínimo de mudança sem o problema do retorno de 9 para 0.

G_1	G_0	G_2	G_1	G_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	1
1	1	0	1	1	0	0	1	1
1	0	0	1	0	0	0	1	0
		1	1	0	0	1	1	0
		1	1	1	0	1	1	1
		1	0	1	0	1	0	1
		1	0	0	0	1	0	0
					1	1	0	0
					1	1	0	1
					1	1	1	1
					1	1	1	0
					1	0	1	0
					1	0	1	1
					1	0	0	1
					1	0	0	0

Note que somente um *bit* muda quando se passa de um código para o seguinte.

Tabela II – Código de Gray de 2, 3 e 4 bits.

2. PARTE EXPERIMENTAL

A experiência consiste no projeto e verificação prática de um sistema de conversão de códigos, onde os códigos de entrada e saída são indicados na **Tabela III**.

Tabela III – Códigos de entrada e saída utilizados na experiência

Código de Entrada (BCD)				Código de Saída				
A	B	C	D	V	W	X	Y	Z
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	1	1
0	1	0	0	0	1	1	1	1
0	1	0	1	1	1	1	1	1
0	1	1	0	1	1	1	1	0
0	1	1	1	1	1	1	0	0
1	0	0	0	1	1	0	0	0
1	0	0	1	1	0	0	0	0

Para a minimização das funções booleanas obtidas, considere a existência de 6 combinações de entrada que não são utilizadas. Estas combinações são chamadas de **condições irrelevantes** (*don't care*).

- 2.1. Obtenha as funções booleanas através do mapa de Karnaugh considerando os *don't cares*, marcando-os com um X.
- 2.2. Desenhe o diagrama lógico total obtido usando o número mínimo de portas lógicas (**Pré-projeto 1**).
- 2.3. Desenhe um diagrama lógico total obtido usando apenas portas NAND de 2 e 4 entradas (**Pré-projeto 2**).
- 2.4. Realize a simulação funcional, escreva a tabela verdade obtida, e realize a simulação temporal de cada um dos circuitos obtidos nos itens 2.2 e 2.3 usando o software Quartus-II (**Pós-Experimento 1**).
- 2.5. Compare os resultados obtidos nas simulações de forma de onda e comente-os.
- 2.6 Considerando a interface (chaves e LEDs) mostrada na **Figura 4**, sintetize o conversor de códigos projetado no item 2.2 no kit de desenvolvimento DE2 e teste filmando seu funcionamento para as entradas definidas na tabela verdade (**Pós-Experimento 2**).

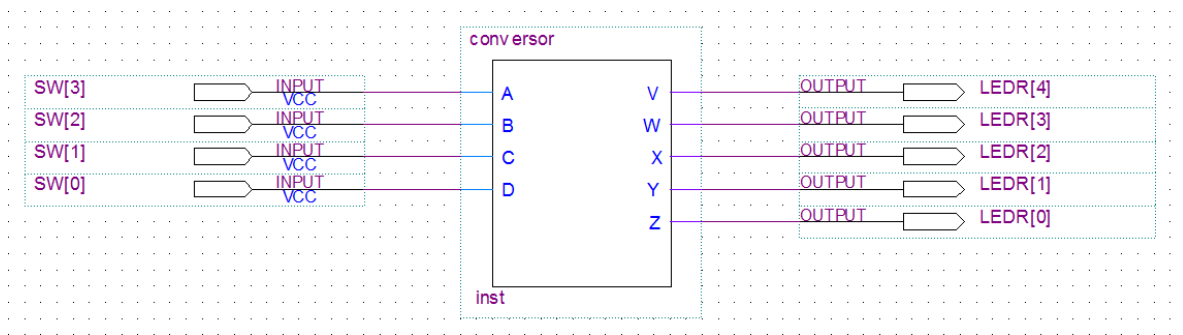


Figura 4 – Interface com a o kit de desenvolvimento DE2

3. SUMÁRIO

Um conversor de códigos promove a tradução de um código para o outro. Assim, dois sistemas digitais utilizando códigos diferentes podem ser interligados por meio de um conversor de códigos. Alguns códigos mais frequentes na prática são vistos e suas particularidades são estudadas.

4. EQUIPAMENTOS E MATERIAL

- software Quartus II versão 13.0
- kit de desenvolvimento em FPGA DE2 Altera

5. TESTE DE AUTO-AVALIAÇÃO

1. O código BCD **0110** corresponde ao seguinte número decimal:
 - a) 5
 - b) 6
 - c) 7
 - d) 8
2. O código excesso-3 **0110** corresponde ao seguinte número decimal:
 - a) 3
 - b) 6
 - c) 7
 - d) 9
3. O código binário refletido é também conhecido como:
 - a) Excesso-3
 - b) Biquinário
 - c) Gray
 - d) BCD
4. Qual dos seguintes códigos é ponderado:
 - a) Excesso-3
 - b) 2 out of 5
 - c) Gray
 - d) BCD
5. Se o código de entrada de um conversor o decimal, teremos um:
 - a) Codificador
 - b) Decodificador
 - c) As opções **a** e **b** estão corretas
 - d) NDA
6. O código de Gray **11001** corresponde ao binário:
 - a) 11110
 - b) 10001
 - c) 01110
 - d) NDA