

# Experimento 05

## Circuitos Combinacionais: Codificador de Decodificador

Giulia Moura Ferreira, 20/0018795  
Ruan Petrus Alves Leite, 21/1010459  
Grupo G8

<sup>1</sup>Dep. Ciência da Computação – Universidade de Brasília (UnB)  
CIC0231 - Laboratório de Circuitos Lógicos

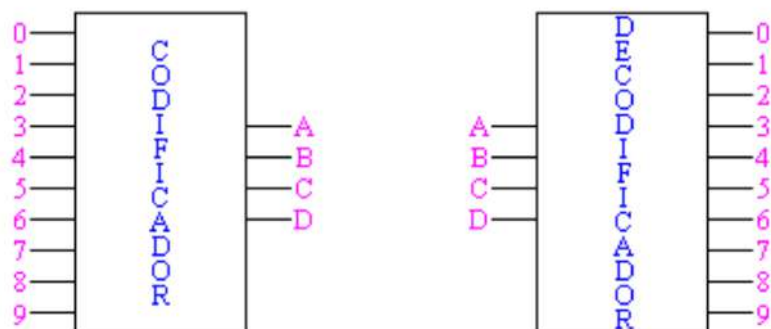
giuliamferreira01@gmail.com, pedraruan12@gmail.com

**Abstract.** *At this report it is going to be presented a description of the construction and execution of an encoder and a decoder according to the Gray code, an unweighted code where from one number to the other there is only one bit of variation. Essential concepts for undersatanding combinational circuits will also be used during the design of the experiments, such as boolean functions, considering their respective "don't cares" (optional conditions), waveforms and truth tables.*

**Resumo.** *Neste relatório será apresentada uma descrição da construção e execução de um codificador e um decodificador segundo o código de Gray, ou seja, um código não ponderado onde de um número para o outro há apenas um bit de variação. Conceitos essenciais para o entendimento de circuitos combinacionais também serão aproveitados durante a projeção dos experimentos, como funções booleanas, considerando seus respectivos "don't cares"(condições optativas), formas de onda e tabelas verdade.*

### 1. Introdução

Circuitos combinacionais são circuitos construídos com portas lógicas de forma a determinar o valor das saídas apenas em função dos valores atuais das entradas; pode-se afirmar que os circuitos combinacionais são representados por um conjunto de equações booleanas: uma equação para cada saída. Codificadores e decodificadores são componentes essenciais ao falar sobre circuitos lógicos digitais, e, apesar de parecidos, ambos possuem suas particularidades. Os codificadores são circuitos que possuem diferentes entradas e um único valor de saída para cada uma delas, sendo essa saída uma codificação da entrada. A saída é um valor binário que pode ser de 1 ou mais bits, e as entradas são independentes, tendo apenas 1 bit (1 para on, 0 para off), também é importante lembrar que apenas uma entrada poderá ser acionada por vez. Em contrapartida, os decodificadores possuem uma lógica de funcionamento contrária, ou seja, possuem diferentes saídas e um único valor de entrada, sendo essas saídas uma decodificação da entrada. Na figura 1 pode-se observar, graficamente, o que foi explicado nesta seção.



**Figura 1. Codificador e decodificador envolvendo o código decimal**

### 1.1. Objetivos

Este relatório traz como objetivo elaborar um codificador e um decodificador usando-se circuitos combinacionais e aplicando-se as técnicas de minimização de funções lógicas, assim como a verificação de possibilidade de conversão de um número decimal em um número binário de código qualquer, e logo depois, sua própria decodificação. O código binário usado será o código de Gray.

### 1.2. Materiais

Neste experimento foram utilizados os seguintes materiais e equipamentos:

- Software Deeds Simulator

## 2. Procedimentos

Utilizando o Software de Simulação Deeds, os procedimentos listados a seguir foram projetados e executados de acordo com os critérios necessários para o funcionamento e análise do codificador e decodificador, segundo o código de Gray apresentado na tabela 1. A execução dos procedimentos foi registrada e pode ser acessada a partir deste link.

**Tabela 1. Código de Gray**

Decimal	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1

É importante lembrar que o código decimal é representado por 10 símbolos diferentes, por exemplo, o símbolo '0' é codificado como '000000001', o '1' como '000000010', ... e o símbolo '9' como '100000000'.

## 2.1. Codificador

Recapitulando, os codificadores são circuitos lógicos combinacionais, no qual obtém-se um conjunto de entradas onde apenas uma delas pode ser verdadeira por vez, já as saídas são a codificação binária da entrada acionada no momento. Nesse caso, o codificador receberá 10 variáveis de entrada, as entradas possíveis seguindo os requisitos especificados e suas respectivas saídas foram representadas através da tabela 2.

**Tabela 2. Tabela Verdade Codificador**

Entradas										Saídas			
A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	A	B	C	D
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	1	1	0
0	0	0	0	1	0	0	0	0	0	0	1	1	1
0	0	0	1	0	0	0	0	0	0	0	1	0	1
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	0	0	0	0	1	1	0	1

Utilizando a tabela acima foi possível obter as funções booleanas do circuito em questão, para facilitar sua construção assim como apresentar os resultados de cada saída obtida, apresentadas a seguir. [Tocci 2011]

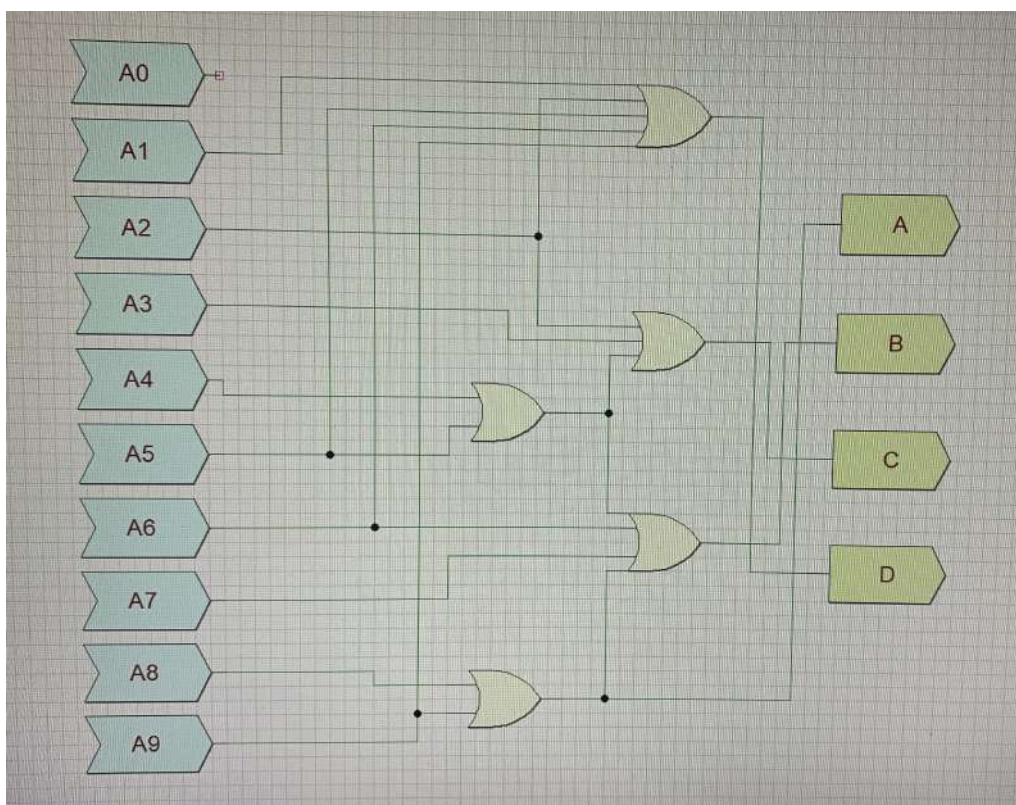
$$A = A_9 + A_8 \quad (1)$$

$$B = A_9 + A_8 + A_7 + A_6 + A_5 + A_4 \quad (2)$$

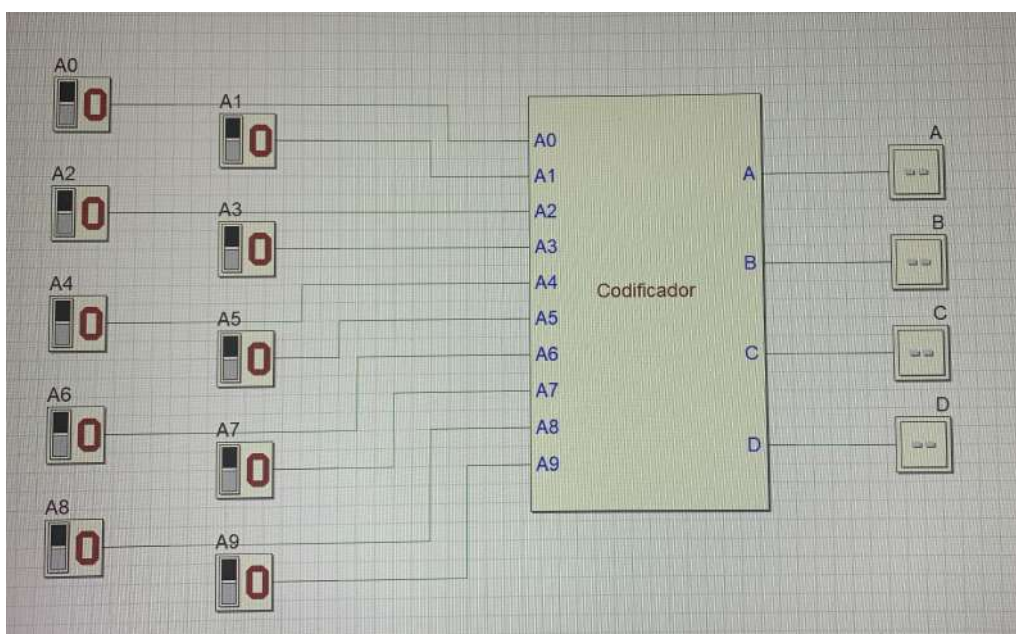
$$C = A_5 + A_4 + A_3 + A_2 \quad (3)$$

$$D = A_9 + A_6 + A_5 + A_2 + A_1 \quad (4)$$

Com as funções booleanas encontradas, foi possível construir o circuito lógico que possibilita a codificação das entradas fornecidas, apresentado nas figuras a seguir, onde a figura 2 representa seu subcircuito, também chamado de Block, ou seja, um diagrama esquemático do codificador projetado, e a figura 3 representa o circuito completo, no qual as entradas serão ligadas, ou desligadas, dependendo do decimal a ser codificado.



**Figura 2. Esquema do circuito codificador**



**Figura 3. Circuito codificador**

A execução do circuito codificador construído foi registrada e pode ser encontrada a partir deste link, assim como a verificação dos resultados obtidos na tabela 2. Ao executar o circuito através do Simulador de Diagrama de Tempo, fornecido pelo Software



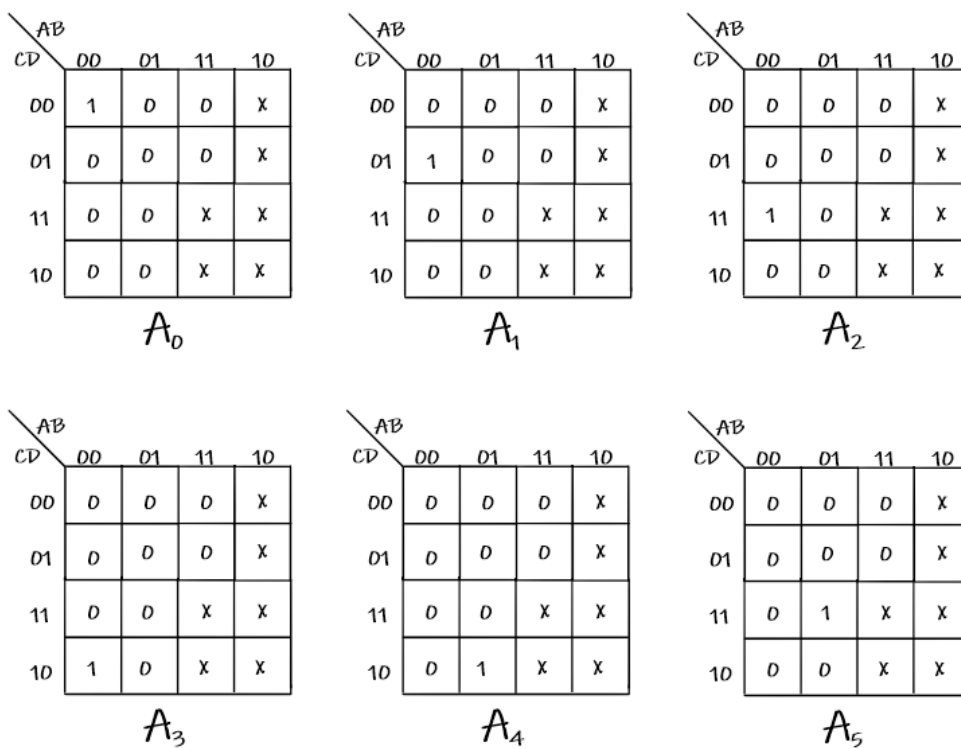


Figura 5. Mapas Karnaugh da Tabela 3 parte 1

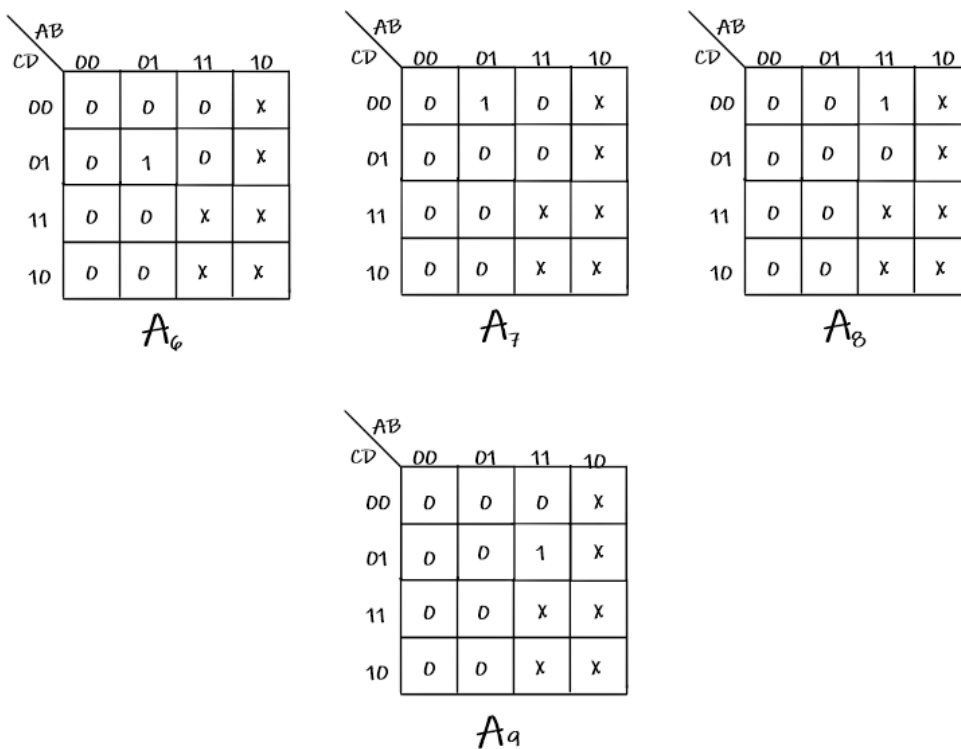


Figura 6. Mapas Karnaugh da Tabela 3 parte 2

Utilizando-se as informações obtidas da tabela 3, os mapas de Karnaugh de cada saída foram construídos e, a partir destes, suas respectivas funções booleanas foram encontradas e representadas nas equações a seguir. [Floyd 2007]

$$A_0 = \overline{B}.\overline{C}.\overline{D} \quad (5)$$

$$A_1 = \overline{B}.\overline{C}.D \quad (6)$$

$$A_2 = \overline{B}.C.\overline{D} \quad (7)$$

$$A_3 = \overline{B}.C.D \quad (8)$$

$$A_4 = B.C.\overline{D} \quad (9)$$

$$A_5 = B.C.D \quad (10)$$

$$A_6 = \overline{A}.B.\overline{C}.D \quad (11)$$

$$A_7 = \overline{A}.B.\overline{C}.\overline{D} \quad (12)$$

$$A_8 = A.\overline{D} \quad (13)$$

$$A_9 = A.D \quad (14)$$

Tendo conhecimento das funções booleanas listadas acima, torna-se possível construir o circuito lógico que decodifica as entradas fornecidas, o subcircuito do decodificador pode ser observado na figura 7 e o circuito completo na figura 8, onde é possível obter a decodificação dos códigos de Gray das entradas em decimal. A execução do circuito construído e a verificação dos resultados obtidos na tabela 3 foi registrada e pode ser acessada neste link.



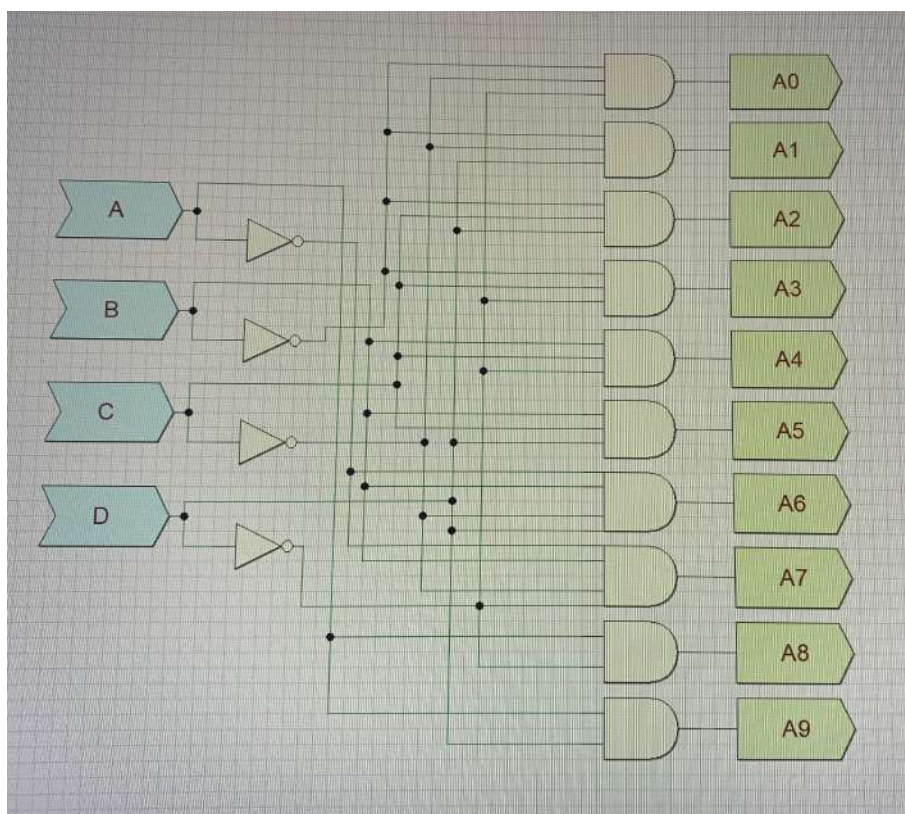


Figura 7. Esquema do circuito decodificador

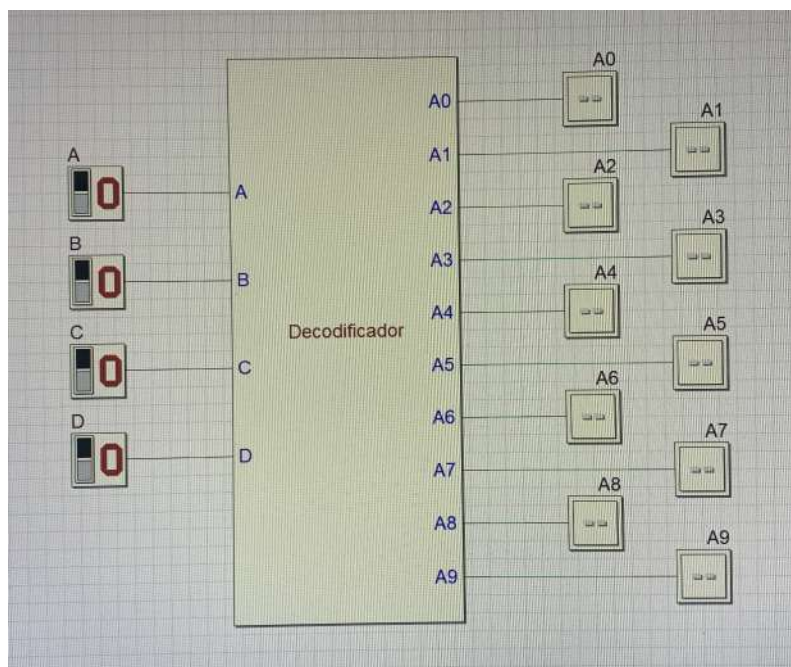
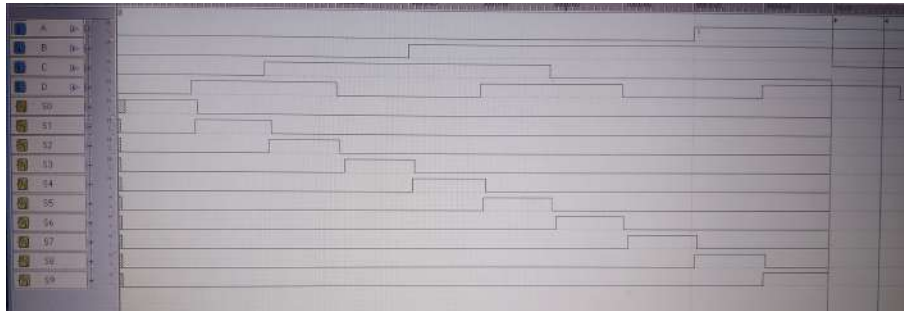


Figura 8. Circuito decodificador



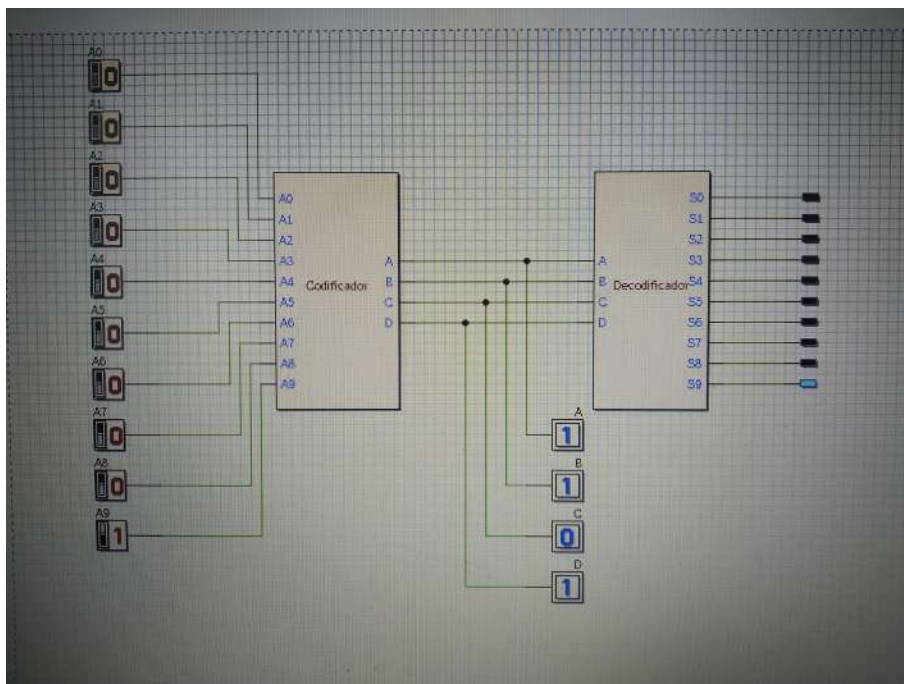
Após executar o circuito, foi possível analisar as formas de onda obtidas pelas saídas, registradas na figura 9, assim como no circuito codificador, as saídas se comportaram como o esperado, não foram presenciadas situações indesejadas ou picos inesperados, seguindo a lógica de suas respectivas funções booleanas



**Figura 9. Formas de onda do circuito decodificador**

### 2.3. Codificador e Decodificador

Nesta seção, os dois circuitos projetados nas seções anteriores serão usados para construir um circuito maior, que codifica as entradas, e logo depois, as decodifica novamente. A figura 10 apresenta os blocos das imagens 2 e 7 inseridos no circuito final. Entre estes blocos, pode-se observar as saídas, ou seja, as entradas codificadas, e, após o bloco do circuito decodificador, os LEDs representam as saídas decodificadas novamente, voltando para seu estado inicial. A simulação interativa foi registrada e pode ser encontrada a partir deste link.



**Figura 10. Circuito (de)codificador**

### 3. Análise dos Resultados

#### 3.1. Análise do Codificador - Procedimento 2.1

Ao observar o circuito codificador projetado na seção 2.1, é importante pontuar que a entrada  $A_0$  não foi conectada a nenhuma porta ou saída lógica, isso acontece porque as saídas permanecem desligadas independentemente do seu valor de entrada, por conta disso, a entrada  $A_0$  pode ser considerada uma condição optativa, ou "don't care", portanto a tabela 2 pode ser apresentada também, de forma em que todos os valores de  $A_0$  sejam representados pela letra X.

Outro ponto importante é compreender que, como um passo essencial para a construção de uma tabela-verdade, o número de linhas da tabela dependerá da quantidade de sentenças que compõem a proposição, portanto, com 10 variáveis de entrada, existe a possibilidade de obter-se 1024 ( $2^{10}$ ) diferentes valores de saída, entretanto, para este procedimento em específico, foram consideradas apenas 10 valores de saída, que são as situações em que cada entrada será acionada apenas uma única vez durante todo o processo, como explicado anteriormente.

#### 3.2. Análise do Decodificador - Procedimento 2.2

Em relação ao circuito decodificador projetado na seção 2.2, pode-se observar que foi possível montar o mapa de Karnaugh de cada uma das saídas, como foram fornecidas quatro entradas, a quantidade de saída possíveis foi 16 ( $2^4$ ), entretanto, todas as saídas que obtinham mais de um bit verdadeiro, foram consideradas condições optativas, ou "don't care", no momento da construção do mapa karnaugh e da tabela 3. É possível observar, também, através dos mapas de Karnaugh das figuras 5 e 6, a "movimentação" do bit acionado ao decorrer das variações de entrada, registrado na figura abaixo.

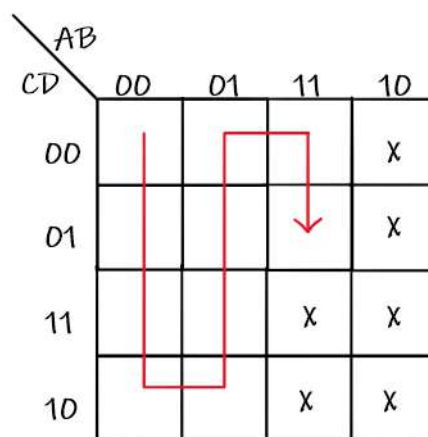


Figura 11. Movimentação do bit verdadeiro nas saída decodificadas

#### 3.3. Análise do (De)Codificador - Procedimento 2.3

O circuito final contém um codificador e um decodificador, ligados à quatro saídas de 1 bit e 10 LEDs de teste. Pode-se observar que cada LED representa um dos bits de entrada, ou seja, quando um bit de entrada é acionado, seu LED correspondente também é, enquanto isso, a saída obtida é a codificação do código decimal inserido inicialmente.

#### **4. Conclusão**

Este relatório foi composto por dois procedimentos essenciais, sendo eles a construção de um codificador e de um decodificador, em cada processo de construção foi possível observar suas similaridades e diferenças, em tese, o resultado desejado seria a codificação de um número binário em decimal, e, logo em seguida, sua decodificação de volta para o número binário.

Na construção do codificador, foram usadas portas OR para que a saída desejada fosse possível, de acordo com suas respectivas funções booleanas. A partir das formas de ondas obtidas na figura 4 pode-se observar o comportamento coerente com o resultado da tabela-verdade criada para a realização do circuito lógico

Já na construção do decodificador, foram utilizadas portas NOT para a inversão das entradas e portas NAND de duas, três e quatro entradas para seus respectivos cálculos de saída, como observado no codificador, as formas de onda do decodificador também não apresentaram nenhum comportamento inesperado, sendo originadas ondas de acordo com a tabela verdade do circuito.

Portanto, é compreensível que a utilização das técnicas de minimização de funções booleanas facilitou todo o procedimento de construção dos circuitos, desde a descoberta das tabelas-verdade até a análise das ondas originadas pelas saídas, tornando-se possível a codificação e decodificação das entradas fornecidas.

#### **Referências**

- [Floyd 2007] Floyd, T. L. (2007). Portas lógicas. In Nowaczyk, D. W., editor, *Sistemas Digitais*, pages 128–176. Bookman.
- [Tocci 2011] Tocci, R. J. (2011). Sistemas digitais. In Levensteinas, S., editor, *Princípios e Aplicações*, pages 101–173. Pearson.

### **Auto-Avaliação**

1. C
2. C
3. A
4. B
5. C
6. E
7. B
8. F