

SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO UNIVERSIDADE FEDERAL DE VIÇOSA · UFV

CAMPUS FLORESTAL

Trabalho Prático 01- CCF 251

Sistema de Codificação de 4-bits

ANA LUÍSA MOREIRA RODRIGUES – 5389

LUCAS MOREIRA DA COSTA - 5377

Florestal – MG

2023

# Sumário

[Sumário 2](#_Toc147700091)

[1. Introdução 3](#_Toc147700092)

[2. Objetivo 3](#_Toc147700093)

[3. Desenvolvimento 3](#_Toc147700094)

[4. Elaboração de circuito simplificado e verilog 9](#_Toc147700095)

[5. Vídeo de Explicação 12](#_Toc147700098)

[6. Conclusão 12](#_Toc147700102)

[7. Referências 12](#_Toc147700103)

# Introdução

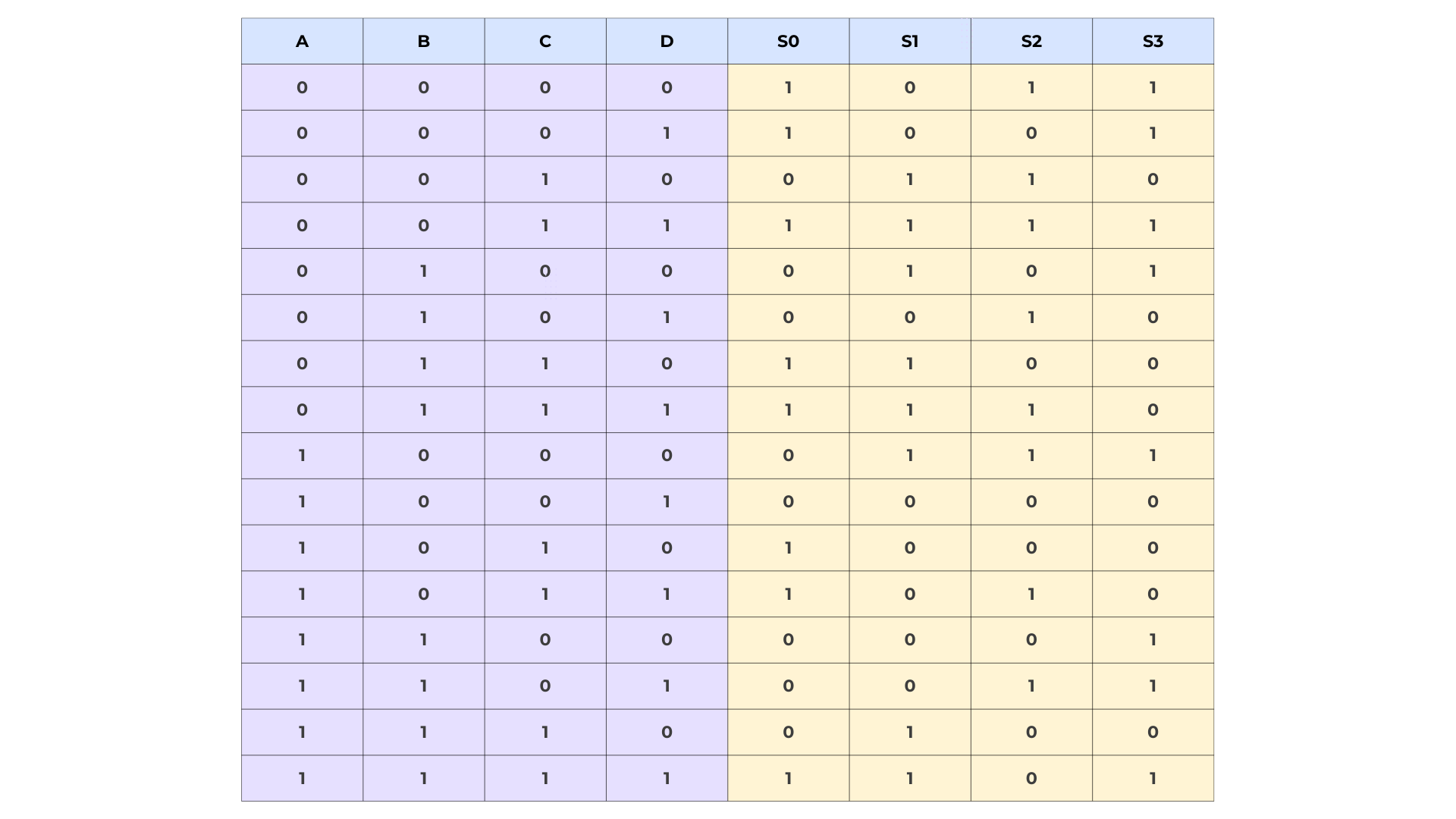
Neste trabalho da disciplina de Introdução Aos Sistemas Lógicos Digitais (CCF 251), fomos encarregados de criar um projeto, que consiste em um codificador de entrada de 4-bits. Essencialmente, trata-se de um processo de criptografia, onde um número de entrada em formato binário é convertido em outro número equivalente. Este relatório detalhará o processo de desenvolvimento, desde a concepção das expressões booleanas, elaboração do circuito simplificado com portas lógicas, implementação efetiva em verilog, até a visualização das formas de onda, destacando as soluções adotadas ao longo deste trabalho prático.

# Objetivo

O objetivo desta prática é desenvolver expressões lógicas com base na tabela-verdade de cada grupo e utilizar Mapas de Karnaugh para simplificá-las,. Uma vez que as expressões lógicas estejam definidas, o segundo objetivo é elaborar módulos e o testbench em verilog, uma linguagem de descrição de hardware, para que o programa possa realizar a criptografia de números no intervalo de 0 a 15. E por fim, simular e visualizar as formas de onda resultantes.

# Desenvolvimento

Em um primeiro momento, foi elaborada a tabela verdade de acordo com as saídas fornecidas para o nosso grupo (30) (Figura 1), em seguida os mapas de Karnaugh para cada saída S0 (Figura 2), S1 (Figura 2), S2 (Figura 3), S3 (Figura4), juntamente com equações booleanas minimizadas, reduzidas, canônicas, mintermo e maxtermo. Figuras a seguir são referentes a tabela verdade e aos mapas das saídas:

****

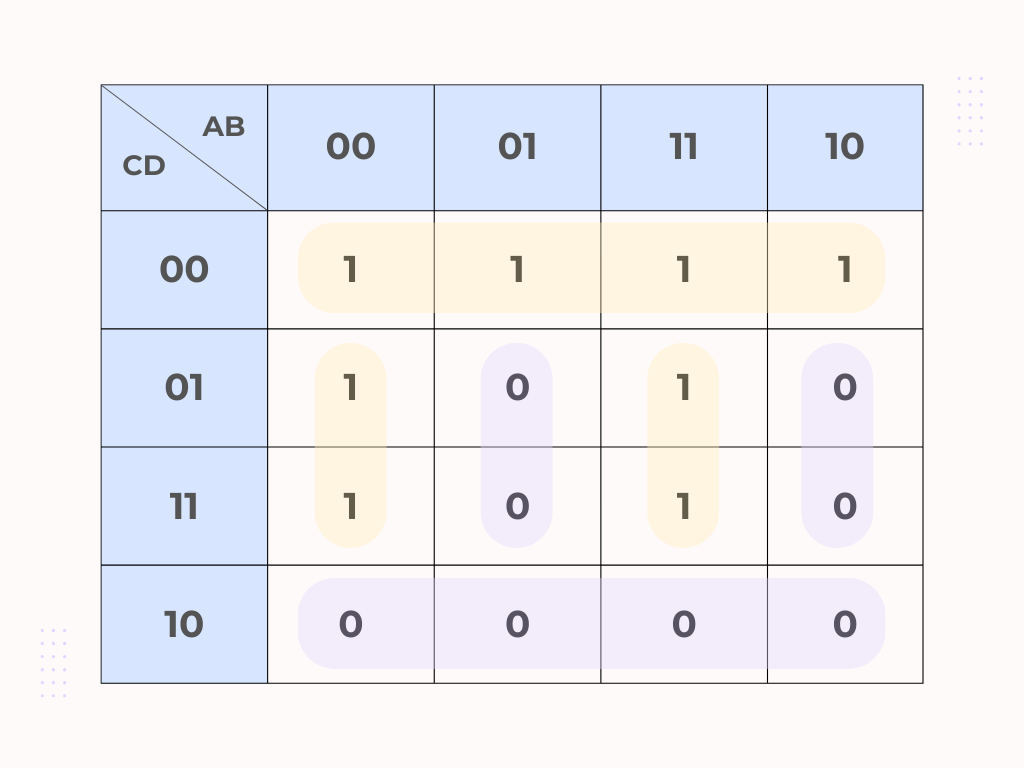
**Figura 1: Tabela verdade**

**Saída S0:**

**Produto de Somas:**

f(A,B,C,D) = (C' + D) (A + B' + D') (A' + B + D')

**Soma de Produtos:**

f(A,B,C,D) = C'D' + A'B'D + ABD

**Figura 2: Mapa de Karnaugh referente a saída S0**

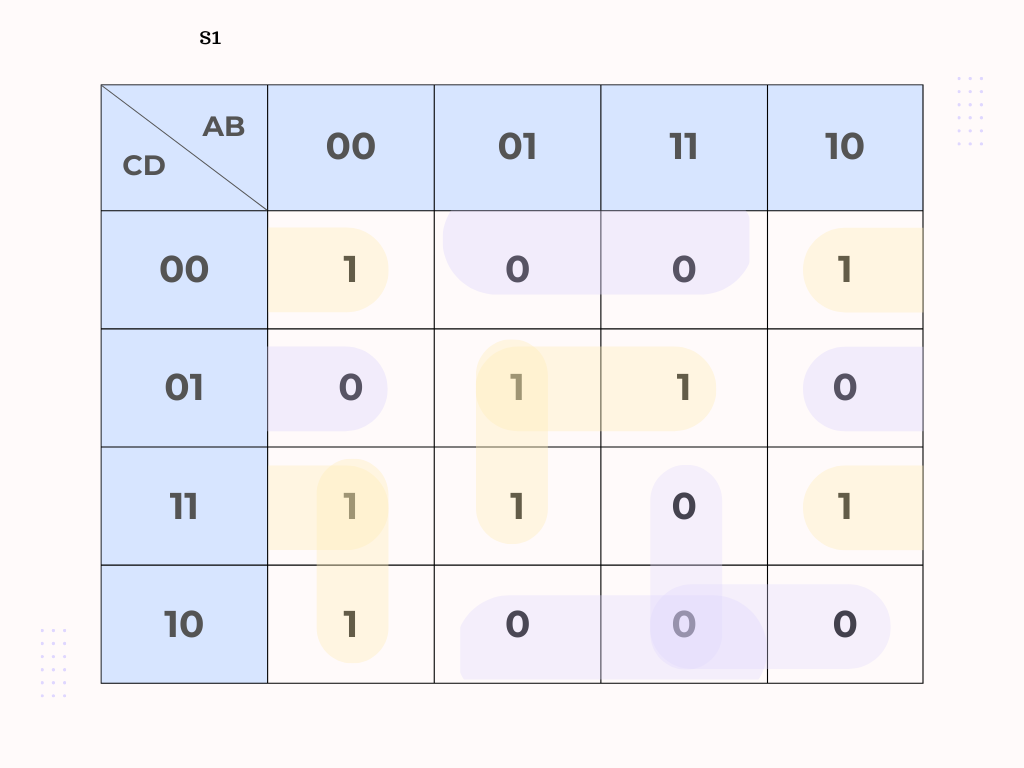
**Saída S1:**

**Produto de Somas:**

f(A,B,C,D) = (B' + D) (B + C + D') (A' + C' + D) (A' + B' + C')

**Soma de Produtos:**

f(A,B,C,D) = A'B'D' + B'C'D' + A'CD + B'CD + BC'D



**Figura 3: Mapa de Karnaugh referente a saída S1**

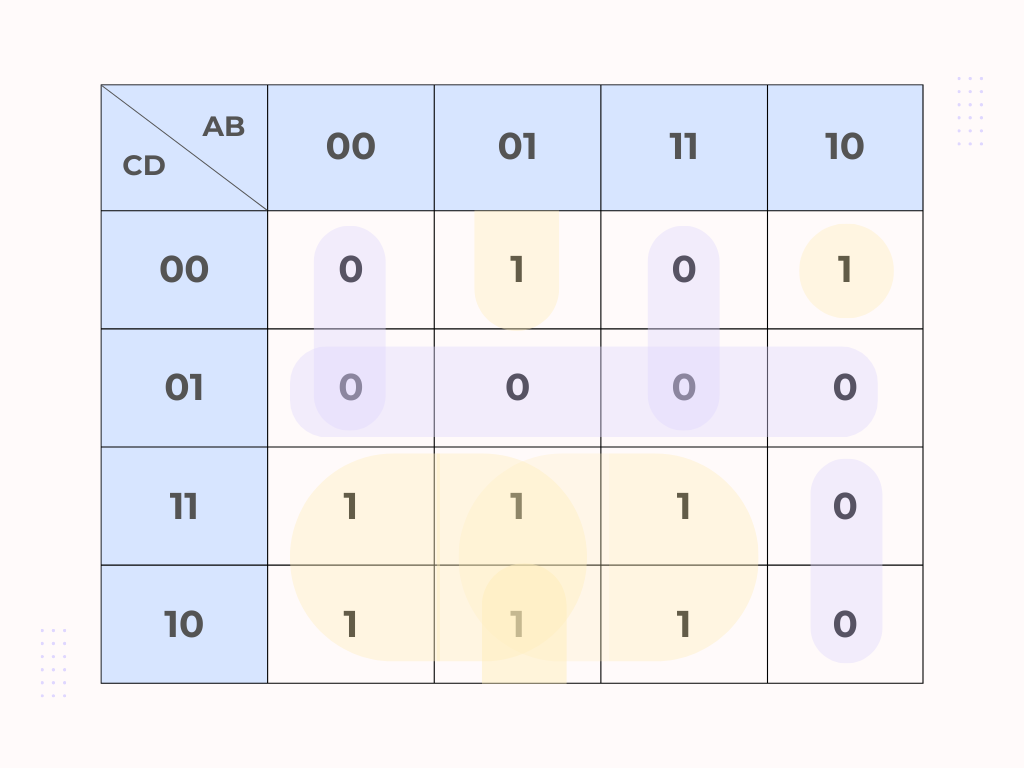
**Saída S2:**

**Produto de Somas:**

f(A,B,C,D) = (C + D') (A + B + C) (A' + B + C') (A' + B' + C)

**Soma de Produtos:**

f(A,B,C,D) = A'C + BC + A'BD' + AB'C'D'



**Figura 4: Mapa de Karnaugh referente a saída S2**

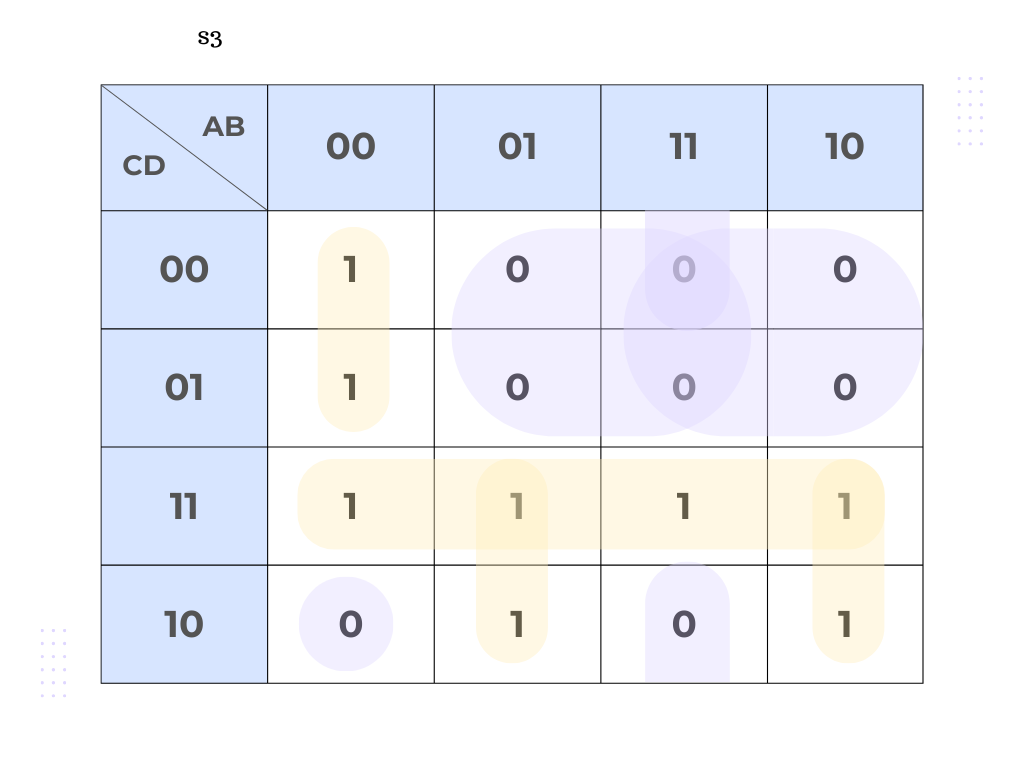
**Saída S3:**

**Produto de Somas:**

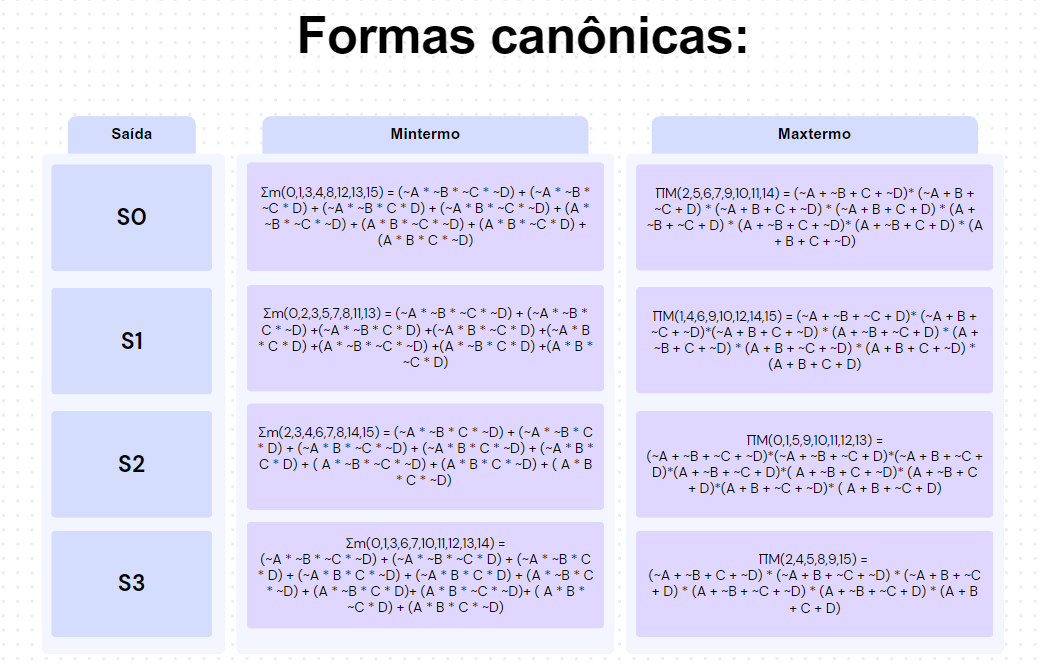
f(A,B,C,D) = (B' + C) (A' + C) (A' + B' + D) (A + B + C' + D)

**Soma de Produtos:**

f(A,B,C,D) = CD + A'B'C' + A'BC + AB'C

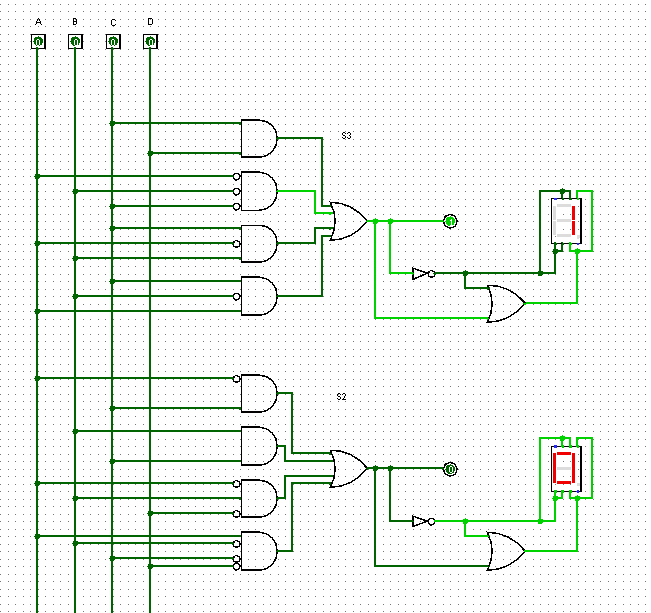


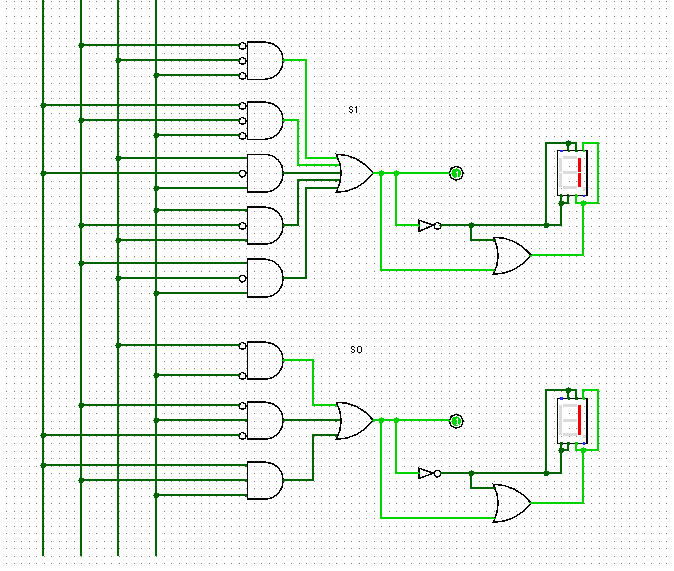
**Figura 5: Mapa de Karnaugh referente a saída S3**

Depois da realização do mapa de Karnaugh, fizemos o levantamento das equações em formas canônicas, mintermo, maxtermo e sua forma reduzida:

# Elaboração de circuito simplificado e verilog

Utilizando as equações dos mintermos, foi montado o circuito lógico no software Logisim, no qual torna possível visualizar o que sai no final de forma interativa. Os pinos no final representam a saída que pode ser 1 ou 0 **.**

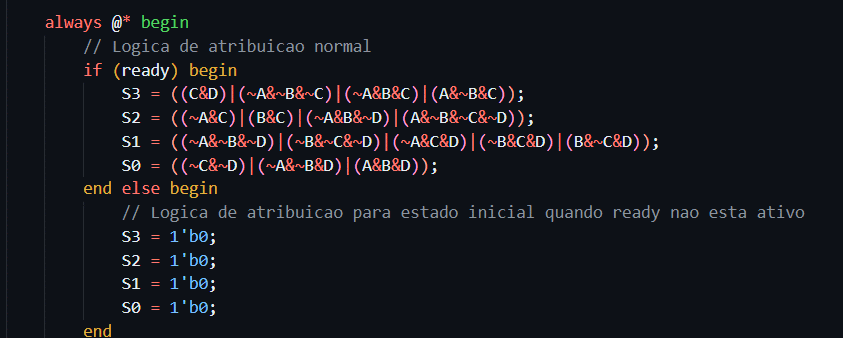




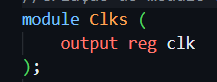
**Figura 6: Circuito simplificado no Software Logisim. (As fotos estão cortadas, mas fazem parte do mesmo circuito).**

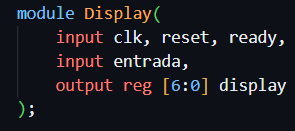
E assim como no Logisim foram usados os mintermos para montagem da lógica.

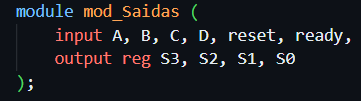
**Figura 7: Equações booleanas minimizadas.**



Os principais módulos na codificação em verilog foram:







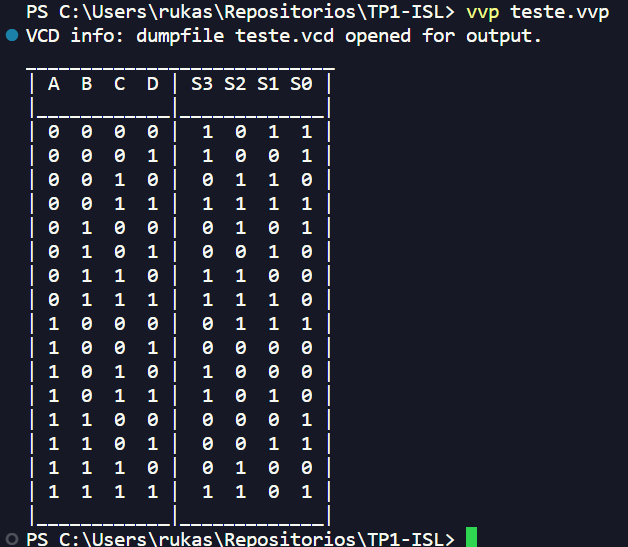
**Figura 8: Declaração das entradas e saídas dos principais módulos implementados.**

Comandos utilizados para compilar o código em verilog e realizar a simulação:





**Figura 9: Comandos utilizados.**



**Figura 10: Saída do algoritmo.**

# 

# 

**Figura 11: Diagrama no Software GTKwave**

# Vídeo de Explicação

No momento do anexo do arquivo .zip no pva net moodle, tivemos um problema de tamanho excedente em relação ao tamanho permitido na plataforma, e portanto, diminuímos a qualidade do vídeo para que pudéssemos enviá-lo.

Como uma outra alternativa, anexamos o link do drive, com o mesmo vídeo em uma qualidade melhor, basta clicar nele e será possível acessá-lo.

https://drive.google.com/file/d/1DveZi-JV3ZeB07CV5HQAPMTb04sSc1u\_/view?usp=sharing

# Conclusão

No desfecho do projeto, é possível observar que os resultados alcançados, através da abordagem que adotamos, estão em conformidade com as expectativas estabelecidas. E, dessa forma, todas as etapas durante este trabalho prático, desde o levantamento das equações booleanas para cada saída até a visualização das formas de onda tiveram significativa relevância no aprendizado e fixação dos alunos envolvidos. É importante frisar que em todo desenvolvimento do projeto houve participação dos integrantes.

# Referências

# 1. HDL Bits: <https://hdlbits.01xz.net/wiki/Main_Page>

# 2. R. Katz, G. Borriello, Contemporary Logic Design, 2ª edição, Prentice Hall,2004.

# 3. ChipVerify: https://chipverify.com