Trabalho Prático 01 – Circuitos Combinacionais
CCF 251 – Introdução aos Sistemas Lógicos
Ciência da Computação – Campus UFV-Florestal
Prof. José Augusto Miranda Nacif

Muitos dos progressos feitos pela humanidade foram em sua grande parte graças à habilidade de comunicação desenvolvida pelos seres humanos. Desdes os primórdios da humanidade existe a necessidade de um indivíduo se comunicar com um grupo seleto de pessoas, e daí surge a motivação para a criação de um método que possibilite transformar uma mensagem em um cifra que, se interceptada, torne difícil a absorção da informação contida na mensagem por parte do interceptador.

A criptoanálise é uma das principais forças de uma nação em uma Guerra. Ela foi muito utilizada nas duas Guerras Mundiais que esse mundo vivenciou. Ela é responsável pela derrocada de líderes militares, assim como pela promoção de outros. Antigamente essa função de criptoanálise de mensagens era feita por poucos engenheiros e matemáticos. Hoje em dia, a quebra de códigos é realizada por inúmeras pessoas com conhecimentos nas mais diversificadas áreas.

Em criptografia, a **Cifra de César**, é uma das mais simples e conhecidas técnicas de criptografia, foi nomeada através de Júlio César, o general Romano que foi o principal responsável por transformar Roma em um império aproximadamente em 50 a.C. Essa cifra era usada no Império Romano para fins de proteção de informações relevantes do ponto de vista militar. É um tipo de cifra de substituição na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes que é determinado por uma chave. O processo de criptografia de uma cifra de César é frequentemente incorporado como parte de esquemas mais complexos, e continua tendo aplicações modernas.

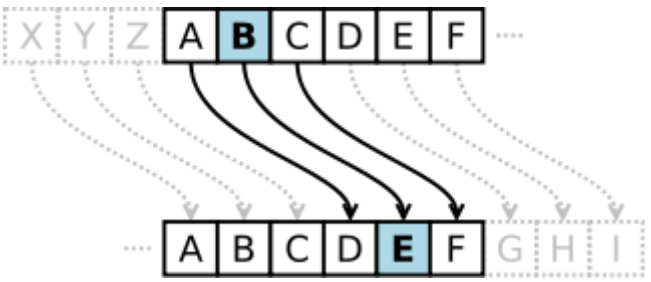

Fig. 1 - A ação de uma cifra de César. Este exemplo está com uma chave de valor 3, então o B no texto normal se torna E no texto cifrado.

Neste trabalho, vocês devem mascarar números, e para isso vocês devem utilizar a ideia da técnica de substituição da Cifra de César. Primeiramente, cada equipe precisa obter a sua **chave** por meio deste link:

https://docs.google.com/spreadsheets/d/1eRVg0VHEILWRntpmOmxxSSLmjn9ZvV7ETHgNNjnvtV4/edit?usp=sharing

Note que a representação binária de cada número da nossa coleção de números pode variar de equipe por causa da chave.

Números	Representação binária
1	1 0 0 0 0
2	1 1 0 0 0
3	1 1 1 0 0
4	1 1 1 1 0
5	1 1 1 1 1
6	0 1 1 1 1
7	0 0 1 1 1
8	0 0 0 1 1
9	0 0 0 0 1
0	0 0 0 0 0

Exemplo utilizando a **chave 1**:

Entrada	Saída
0 0 0 1	1 1 0 0 0

Exemplo utilizando a **chave 2**:

Entrada	Saída
0 0 0 1	1 1 1 0 0

1. Descrição do trabalho

Em posse das informações acima, é necessário que vocês utilizem a linguagem de descrição de *hardware* **Verilog** para descrever o circuito tanto em ambiente virtual (**simulação**) quanto em ambiente físico (**FPGA**) para exibir os números criptografados em um **display de 7 segmentos** (fig. 2). *Vocês podem considerar uma exibição em binário ou decimal nos displays. Exemplo: 11111 pode ser exibido como 11111 ou 31.*

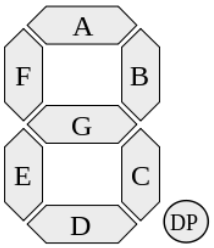

Fig. 2: Display de sete segmentos.

Displays de sete segmentos são comumente usados em eletrônica como forma de exibir uma informação alfanumérica (binário, octadecimal, decimal ou hexadecimal) que possa ser prontamente compreendida pelo usuário sobre as operações internas de um dispositivo e os quais podem ser ligados ou desligados individualmente (fig. 3).

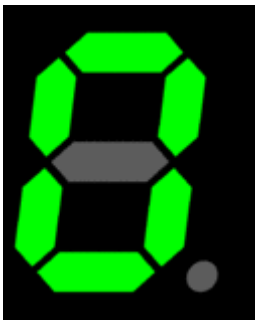

Fig. 3: Simulação de um display LED de sete segmentos exibindo os 16 dígitos.

O objetivo deste trabalho prático consiste na implementação de um encriptador capaz de converter um número (de 0 a 9) em seu número correspondente pela chave. Vocês precisam dos seguintes módulos:

- **número (fig.4):** recebe a sequência binária de 0 a 9 e retorna a um novo valor equivalente. Observe que a descrição dos sinais é apresentada na tabela 1.
- **transmissão**: responsável por receber a sequência de bits proveniente do primeiro e transmiti-la ao meio de saída desejado (ao testbench em simulação e ao FPGA em ambiente real).

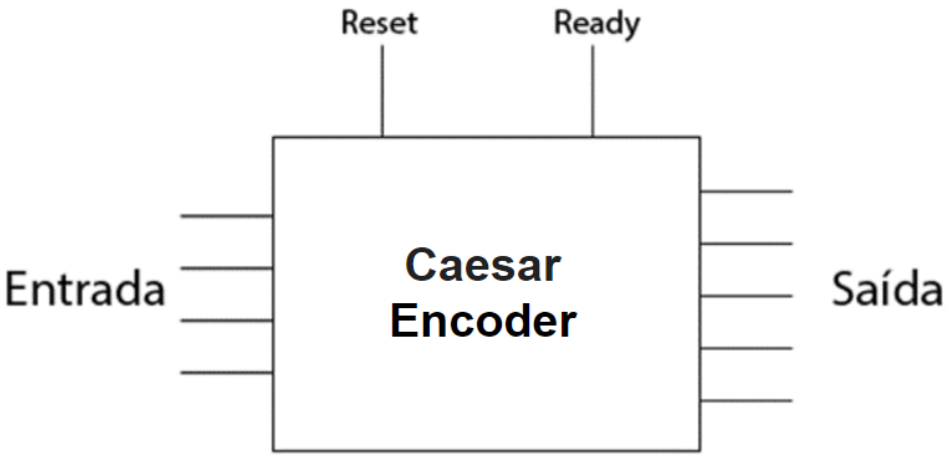

Fig. 4. Esboço do módulo número, responsável pela codificação da entrada

Tabela 1 – Descrição dos sinais

Nome	**Tamanho**	**E/S**	**Descrição**
Número	4 bits	Entrada	Número a ser mascarado
Reset	1 bit	Entrada	Inicializa o sistema.
Ready	1 bit	Entrada	Indica que a entrada do número já foi finalizada. A conversão pode ser iniciada.
Novo número	5 bits	Saída	Número mascarado

Ao final, sua implementação deverá ser capaz de receber um número em binário e exibir seu valor equivalente em relação a sua chave.

Atenção: O módulo ***número*** deve ter sua saída gerada através de **lógica combinacional**. Para isso, você deve propor e simplificar uma tabela-verdade a fim de obter as saídas, de acordo com os passos descritos abaixo na seção 2.

2. Etapas do desenvolvimento

Os passos seguintes devem ser seguidos no processo de desenvolvimento do módulo **número**:

1. Levantamento das **equações booleanas** para cada saída com base na chave da sua equipe;
2. Simplificação das equações booleanas utilizando lógica booleana e **mapas de Karnaugh**;
3. Apresentação das **formas canônicas e reduzidas**;
4. Apresentação do **mintermo** e do **maxtermo** das saídas;

5. Elaboração do **circuito simplificado** com portas lógicas no software Logisim.

Os passos seguintes devem ser seguidos no processo de desenvolvimento de **todos os módulos:**

1. Elaboração e simulação dos módulos em Verilog através da ferramenta Icarus Verilog;
2. Visualização das formas de onda resultantes através da ferramenta GTKWave.

3. Dicas

a. Utilize sinais de controle: Como os módulos saberão se é ou não a hora de transmitir a mensagem final? Através de sinais de controle. Crie tantos quanto forem necessários, desde que sejam coerentes;

b. Reset: Não existe nenhum tipo de circuito que não precise, em algum momento, retornar a um estado inicial. Portanto, não se esqueça de implementar o sinal de controle em questão.

c. Entrada inválida: Utilizem uma forma para representar uma entrada inválida.

4. Observações

Todas as etapas devem ser documentadas no relatório que deverá ser entregue. O moodle será o ambiente de submissão do trabalho, e é necessário que apenas um aluno do grupo o faça. Deverá ser submetido um arquivo compactado (*.zip*) contendo:

- A documentação detalhada em formato *.pdf*,
- Os arquivos em Verilog dos módulos implementados e do módulo de teste (*testbench*)
- Arquivo de simulação de ondas (*.vcd*).
- Um vídeo de até 5 minutos explicando o código e demonstrando o funcionamento.

Atenção: Cópias de trabalhos práticos de outros grupos ou da internet serão exemplarmente punidos. Em caso de cópias entre grupos, a punição será a mesma tanto para quem copiar quanto para quem fornecer o trabalho.

5. Data das Entregas

Simulação: 23/10/2022

Implementação em FPGA: 31/10/2022

6. Dúvidas

As dúvidas relacionadas deverão ser direcionadas ao Matheus Freitas (matheus.f.martins@ufv.br)