

O Código Morse é um sistema binário de representação e comunicação composto por todos os números e letras do alfabeto, além de sinais de pontuação. Os caracteres são representados através de uma combinação específica de pontos e traços, conforme exposto na Figura 1. Para formar palavras, basta realizar a combinação correta dos símbolos.

A	• —	U	• • —
B	— • • •	V	• • — —
C	— • — •	W	• — — •
D	— • •	X	— • • —
E	•	Y	— • — —
F	• • — •	Z	— — • •
G	— — •		
H	• • • •		
I	• •		
J	• — — —		
K	— • — —	1	• — — — —
L	• — • •	2	• • — — —
M	— —	3	• • • — —
N	— •	4	• • • • —
O	— — —	5	• • • • •
P	• — — •	6	— • • • •
Q	— — • —	7	— — • • •
R	• — •	8	— — — • •
S	• • •	9	— — — — •
T	—	0	— — — — —

Figura 1. Padrão internacional do Código Morse

Este trabalho prático consiste na implementação de um codificador Morse capaz de converter um número (de 0 a 9) em seu sinal correspondente em código Morse. A linguagem de descrição de *hardware* **Verilog** deverá ser utilizada para descrever o circuito tanto em ambiente virtual (simulação) quanto em ambiente físico (FPGA).

1. Descrição do trabalho

O objetivo primordial é a implementação de dois módulos: **número** e **transmissão**. O primeiro módulo, representado na figura 2, será responsável por receber uma sequência binária (de 0 a 9) e retornar a sequência de *bits* equivalente ao Código Morse, conforme o padrão apresentado na figura 1. A descrição dos sinais é apresentada na Tabela 1.

O segundo módulo será 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).

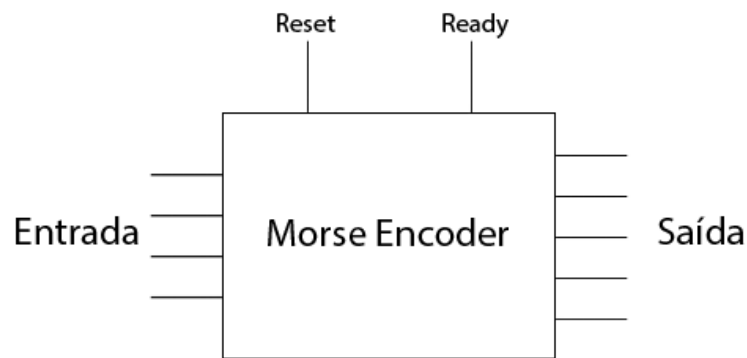


Figura 2. Esboço do módulo número, responsável pela codificação da entrada

Tabela 1 – Descrição dos sinais

Nome	Tamanho	E/S	Descrição
Número	4 bits	Entrada	Número a ser convertido para código Morse.
Reset	1 bit	Entrada	Inicializa o sistema.
Ready	1 bit	Entrada	Indica que a entrada do número já foi finalizada. Conversão pode ser inicializada.
Código	5 bits	Saída	Número em código Morse no qual pontos e traços representados por 1 e 0, respectivamente.

Ao final, seu *design* deverá ser capaz de receber um número em binário e exibir seu valor equivalente em código Morse.

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**:

- Levantamento das equações booleanas para cada saída;
- Simplificação das equações booleanas utilizando lógica booleana e mapas de Karnaugh;
- Apresentação das formas canônicas;
- Apresentação do mintermo e do maxtermo das saídas;
- Elaboração do circuito simplificado com portas lógicas no software [Logisim](#).

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

- Elaboração e simulação dos módulos em Verilog através da ferramenta [Icarus Verilog](#);
- Visualização das formas de onda resultantes através da ferramenta [GTKWave](#).

3. Dicas úteis

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. Trabalhando com um FPGA: Utilize os recursos físicos da placa para receber e transmitir informações. Os LEDs ou os displays de 7 segmentos podem ser utilizados para transmitir a mensagem final em código Morse, por exemplo.

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

4. Observações

O trabalho pode ser feito em grupos de até 4 alunos. Entretanto, a avaliação é individual e dependente do resultado da entrevista à ser realizada. Todas as etapas devem ser documentadas no relatório que deverá ser entregue. O PVANet 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 em formato .pdf, os arquivos em Verilog dos módulos implementados e do módulo de teste (*testbench*) e o arquivo de simulação de ondas (.vcd).

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: 15/10/2018

Implementação em FPGA: 31/10/2018

6. Dúvidas

As dúvidas relacionadas deverão ser direcionadas ao monitor da disciplina, Lucas Duarte (lucas.f.duarte@ufv.br). Os horários das monitorias estão disponíveis no PVANet.