

**Centro de Informática**  
**Universidade Federal da Paraíba**

**Projeto Final**  
**Mundo do Wumpus**

Samuel Luiz Torres Angelo – 2016016023  
Stênio Ellison Pereira Ferreira – 2016068763

João Pessoa 2018

**Centro de Informática**  
**Universidade Federal da Paraíba**

## Mundo do Wumpus

Relatório elaborado para o projeto final da disciplina Circuitos Lógicos II, ministrada pelo Professor Eudisley Gomes dos Anjos no Centro de Informática da Universidade Federal da Paraíba.

## Resumo

O projeto "Mundo do Wumpus" retrata o jogo desenvolvido para a disciplina de Lógica para Ciência da Computação, porém convertido em face da linguagem de programação de hardware Verilog. Para realização do projeto foi utilizado um FPGA Cyclone II modelo DE2. O que permitiu a interação do hardware com o código criado, possibilitando controlar as ações do personagem no jogo através dos switches do FPGA.

**Palavras-chave:** FPGA, interação, swiches.

## **Lista de siglas**

**VHDL** - Hardware Description Language (Linguagem de descrição de hardware)

**FPGA** - Field Programmable Gate Array (Arranjo de portas programáveis em campo)

**IEEE** - O Instituto de Engenheiros Eletricistas e Eletrônicos

**IDE** - Integrated Development Environment (Ambiente de desenvolvimento integrado)

# **Sumário**

- 1. Introdução**
- 2. Metodologia**
- 3. Descrição do projeto**
- 4. Execução do projeto, testes e resultados**
- 5. Conclusões**
- 6. Referências**

# **Introdução**

## **Histórico**

Proposto em 1972 por Gregory Yob, o Mundo do Wumpus criado com o propósito de mudar a situação dos jogos na década de 70, Gregory cansado do padrão baseados em grandes cartesianos, criou um novo modo de jogo que consiste na exploração de outros demais ambientes, permitindo que o personagem do jogo não se locomovesse entre os campos do ambiente, mas sim pela as arestas da figura geométrica, podendo caminhar por uma espécie de "corredores" que consequentemente, o personagem do jogo chegaria em diferentes estados.

## **Enredo**

O personagem, caracterizado por ser um caçador, está em uma caverna escura à procura de ouro, o local ambientado dispõe de um ouro, buracos e o monstro Wumpus. O propósito do jogo é o caçador encontrar o ouro no ambiente e assim vencer o jogo, caso ele caia em um buraco ou seja comido pelo Wumpus, o jogador morre e o jogo termina.

## **Personagens**

Os personagens são:

- O Wumpus
- O Buraco
- O Fedor
- A Brisa
- O Ouro
- O caçador, jogador, também conhecido como agente.



# Metodologia

## Ambiente

O jogo necessita de um raciocínio e percepção para chegar no objetivo que é vencer o jogo, encontrando o "ouro" que está escondido no tabuleiro de 4x4, porém também existe outros locais que o jogador deve evitar, como o "buraco" e o "monstro", sabendo disso, para possibilitar esse feito, foi utilizado o FPGA como meio de executar o jogo e permitir o controle dele do usuário sobre o personagem, estabelecendo um contato entre o jogador e o personagem do Wumpus.

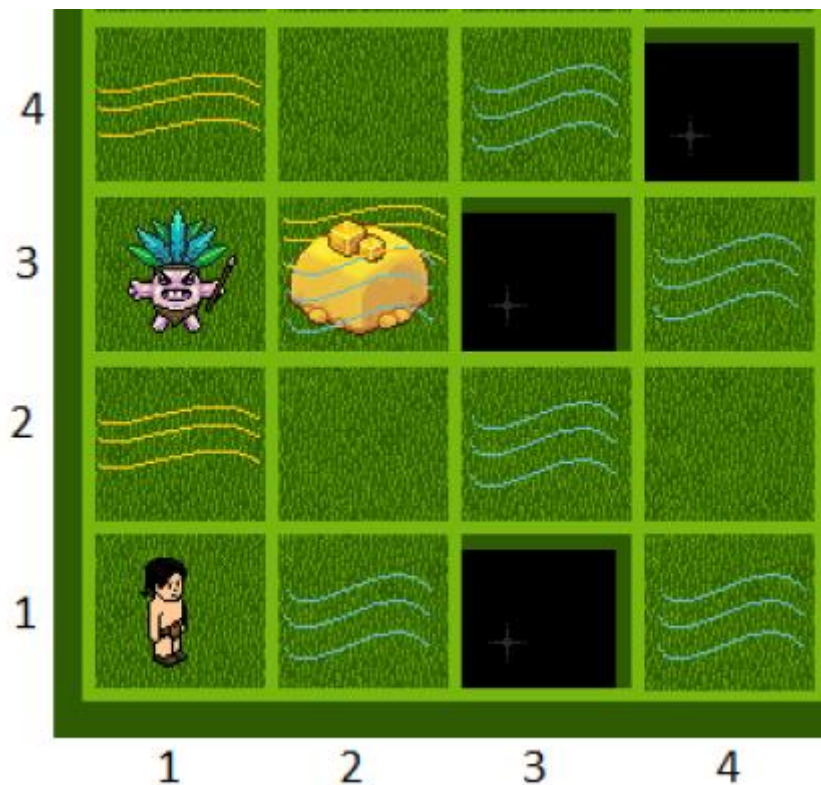
O ambiente do jogo se caracteriza por uma tabela 4x4, onde o jogador inicia na posição (1x1), sendo permitido se locomover pelas posições adjacentes. Pelo tabuleiro estão distribuídas "brisas", "fedores" que ajudarão ao agente a evitar devidas posições onde possivelmente pode haver um buraco ou um único monstro que causa o fim do jogo, pois o jogador sente o fedor do Wumpus(monstro) e percebe as "brisas" ao redor dos buracos.

As "Brisas" aparecem próximas aos "buracos" e os "fedores" indicam que o monstro está próximo dessa região do jogo. O monstro é colocado em uma determinada posição da tabela cuja qual não muda enquanto o jogo está acontecendo. Dentro da tabela há também uma posição onde se localiza o "ouro" que não se modifica no decorrer do jogo, caso o agente consiga chegar até o "ouro", ele vence o jogo e é reiniciada a partida.

Para possibilitar esse feito, foi utilizado o FPGA Cyclone II modelo DE2, através de um código na linguagem Verilog, criou-se uma máquina de estados, que avaliou cada posição do jogador como um estado, possibilitando o armazenamento na memória, da consequência do estado para onde o jogador proporcionando uma percepção dos lugares que ele não deve ir por meio das dicas que são os "fedores" e as "brisas".

## Descrição do Projeto

Para idealização do jogo foi criado uma matriz 4x4, representando a caverna, sendo cada elemento da matriz uma das possibilidades de locomoção do caçador. O monstro foi situado na posição 3x1, ao seu redor se posicionaram fedor como indica as regras do jogo, nas posições 3x2, 2x1 e 4x1. O ouro foi alocado na posição 3x2. Também foram adicionados 3 buracos no mapa, um na posição 1x3, outro na posição 3x3 e outro na posição 4x4. Seguindo as regras brisas foram adicionadas ao redor de cada buraco, nas posições: 1x2, 1x4, 2x3, 3x2, 3x4, 4x3 e 4x5. Como ilustrado na figura a seguir:



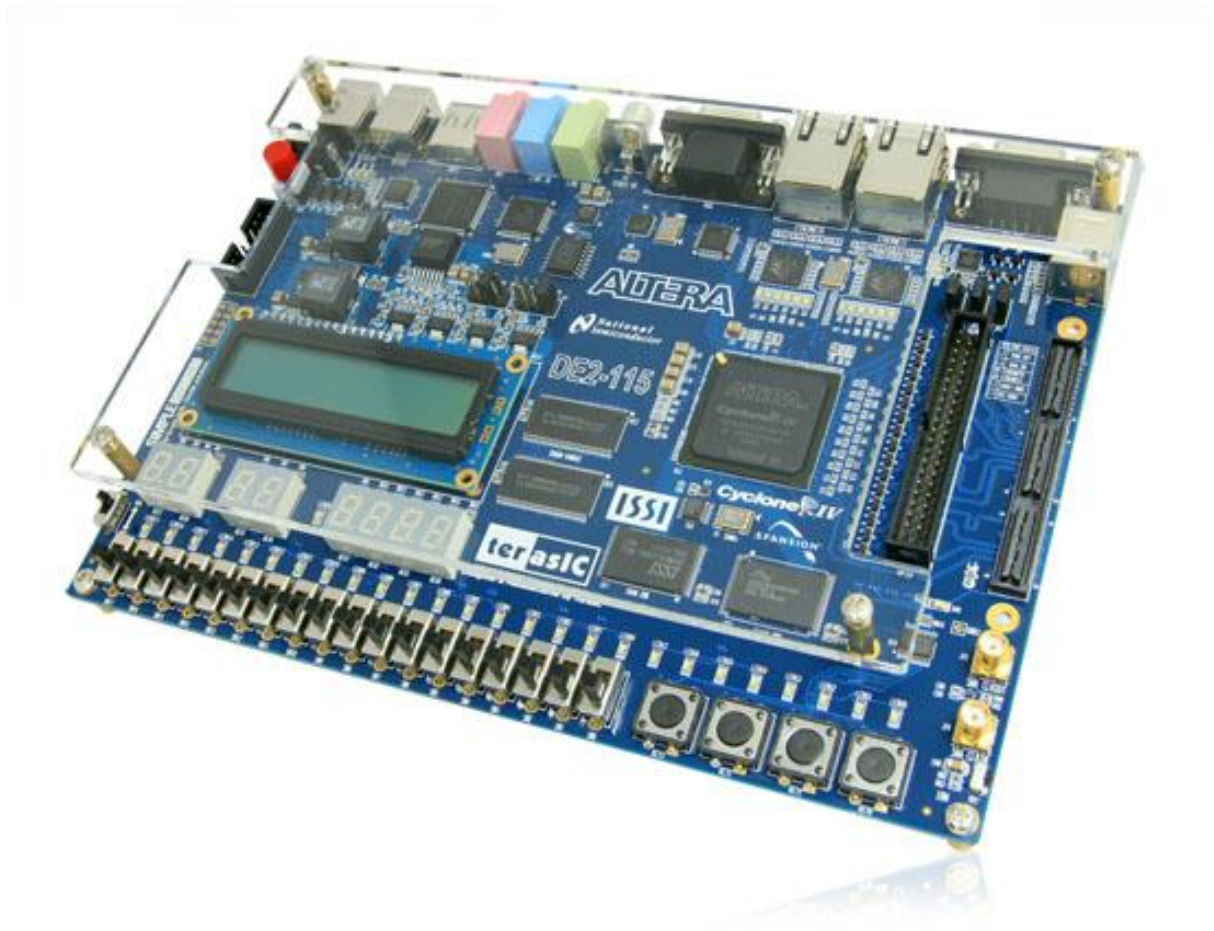
Em face do FPGA , foi utilizado a linguagem Verilog para programar uma máquina de estados de Moore. Utilizando o simulador Quartus II edição 9.7 foi simulado a máquina com 16 estados onde cada estado seria um bloco de mobilidade para o agente. Para representar o mapa foram utilizados 20 leds, 16 para representar a matriz de estados 4x4 e 4 para representar o fedor, o wumpus, a brisa e o ouro. Todos os leds foram situados em uma protoboard.

Para mover o agente pelo mapa foram utilizados 4 switches que indicam a direção desejada pelo usuário e 1 switch para execução do movimento.

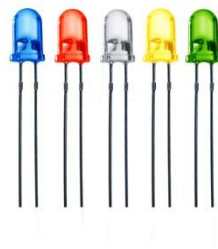
Assim, quando o jogo começar o led que representa a posição 1x1 será aceso, indicando a posição do agente, o usuário deverá indicar a direção que deseja se mover com um dos 4 primeiros switches do FPGA, em seguida deverá executar sua ação acionando o último switch do FPGA. O led que indica a nova posição do agente será aceso, também serão acesos os leds de fedor, brisa, ouro ou wumpus caso no bloco em que o agente está agora contenha um desses. Assim o agente deverá se mover pelo mapa, caso ocorra de cair em um buraco ou no wumpus, o jogador deverá ativar o switch de execução e o jogo será reiniciado. Logo, se movendo pelo mapa entre erros e acertos o usuário deverá descobrir a localização do Ouro e vencer o jogo.

## Execução do projeto, testes e resultados

Para execução do projeto foram utilizados o FPGA Altera DE2-115 Development and Education Board, 16 leds de cor rosa, 1 led de cor azul, 1 led de cor verde, 1 led de cor vermelho, 1 led de cor amarela, 11 resistências de 100 ohms, 9 resistências de 220ohms, 13 jumpers fêmea – fêmea, 6 jumpers macho-fêmea, 20 jumpers macho-macho, 1 protoboard de 400 furos e 1 protoboard de 830 furos.



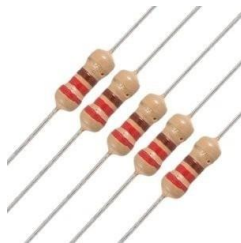
FPGA



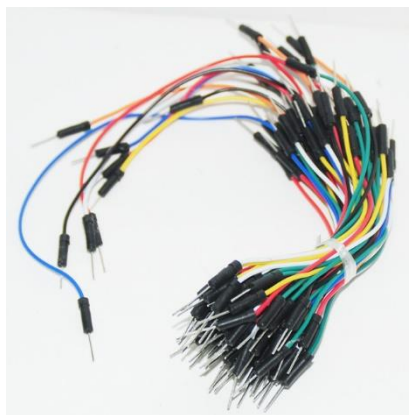
Leds



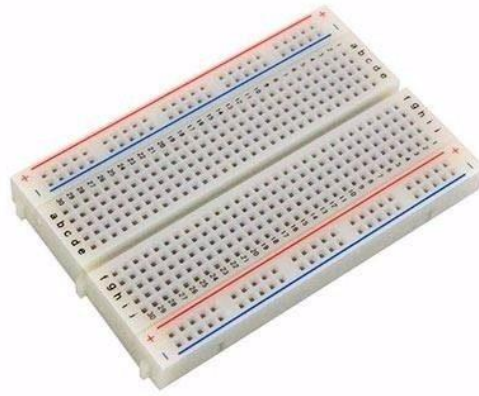
Resistores de 100 ohms



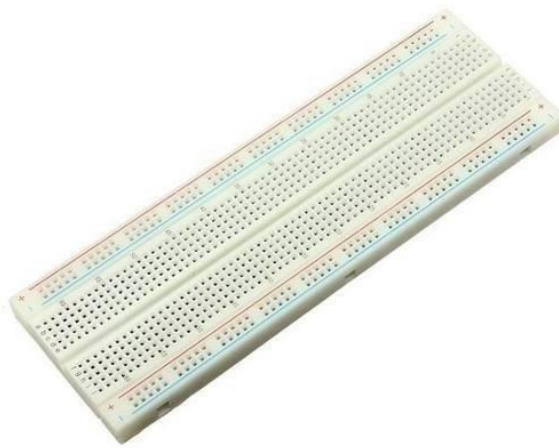
Resistores de 220 ohms



Jumpers



Protoboard 400 furos

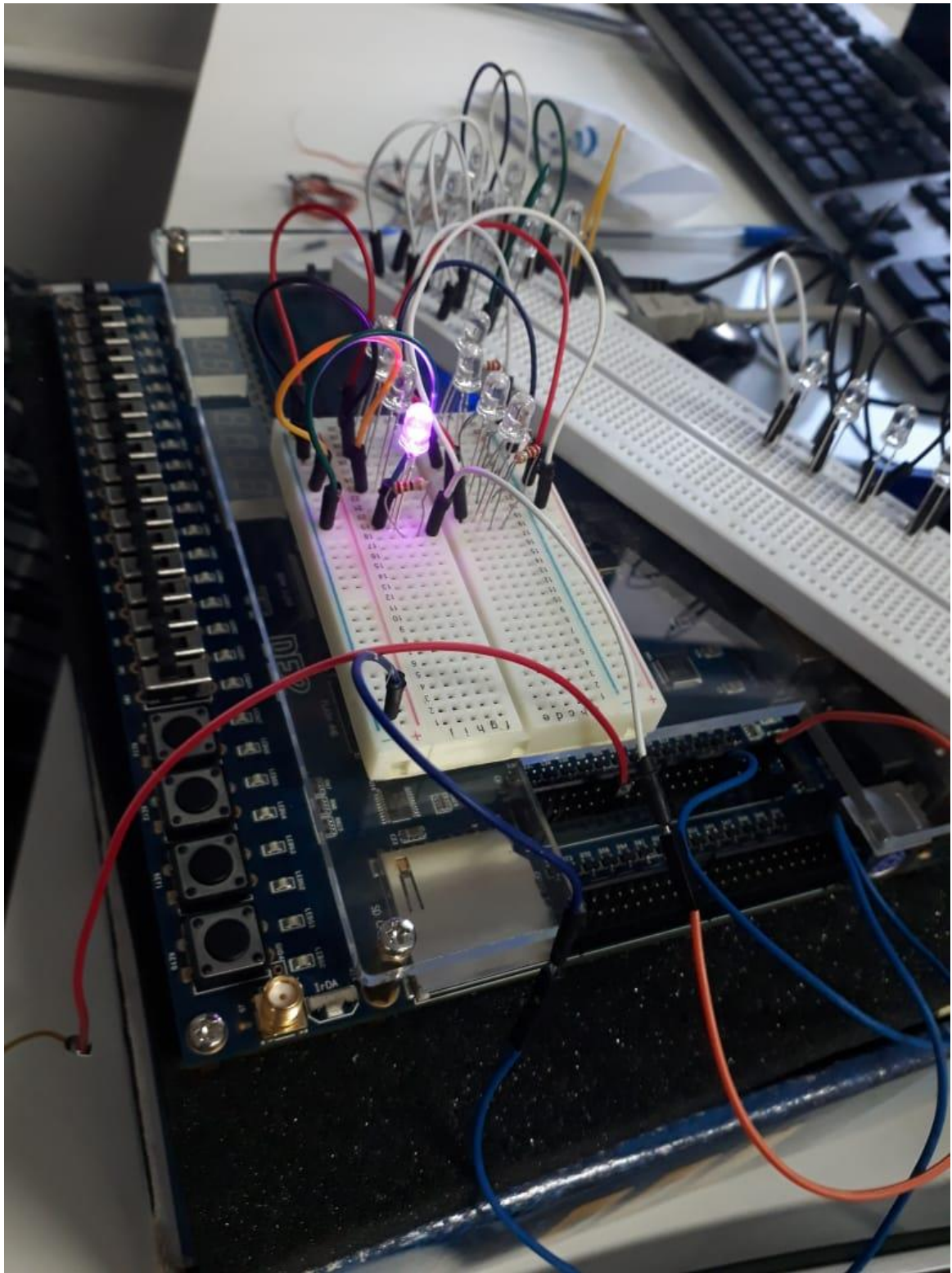


Protoboard 830 furos

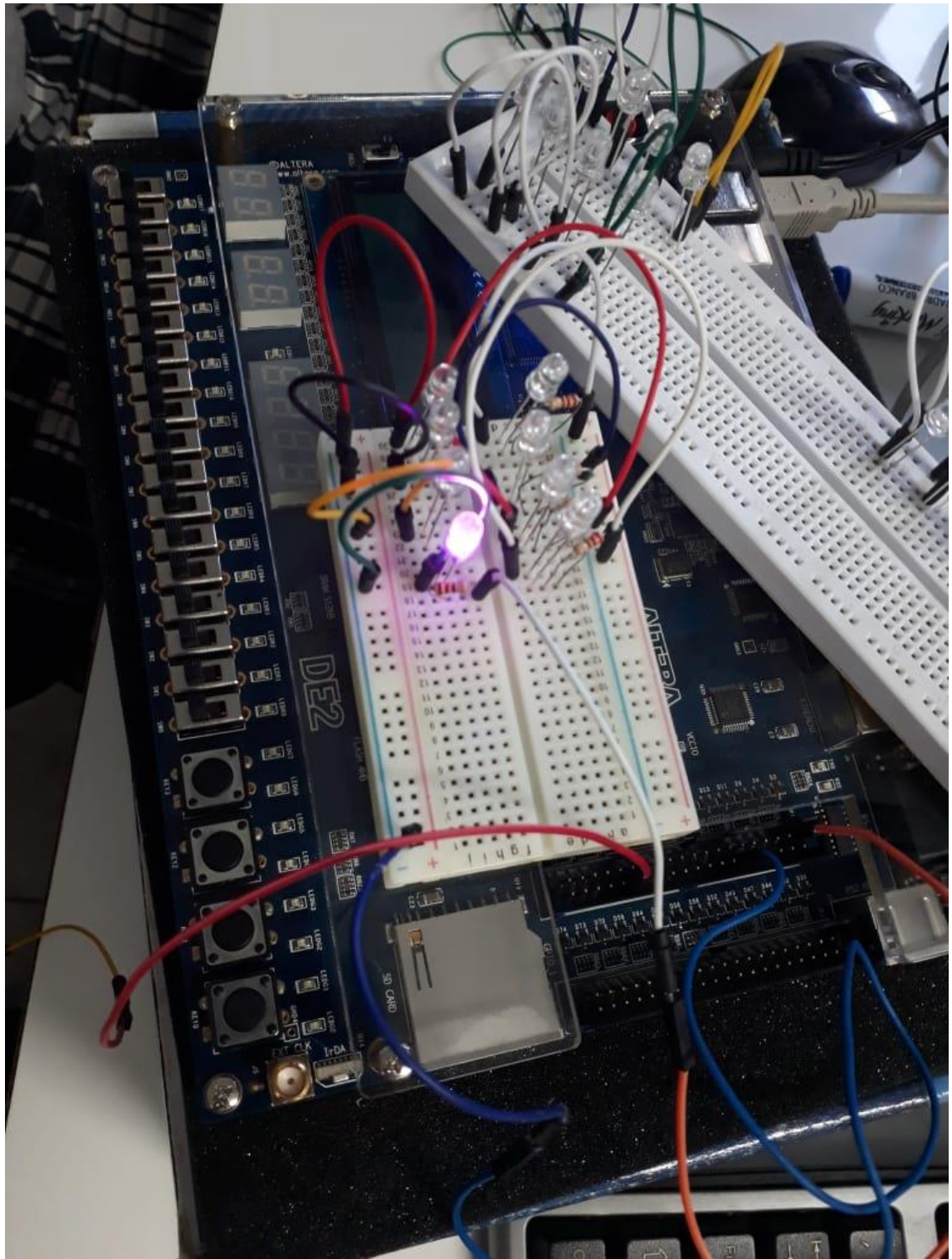
## Testes

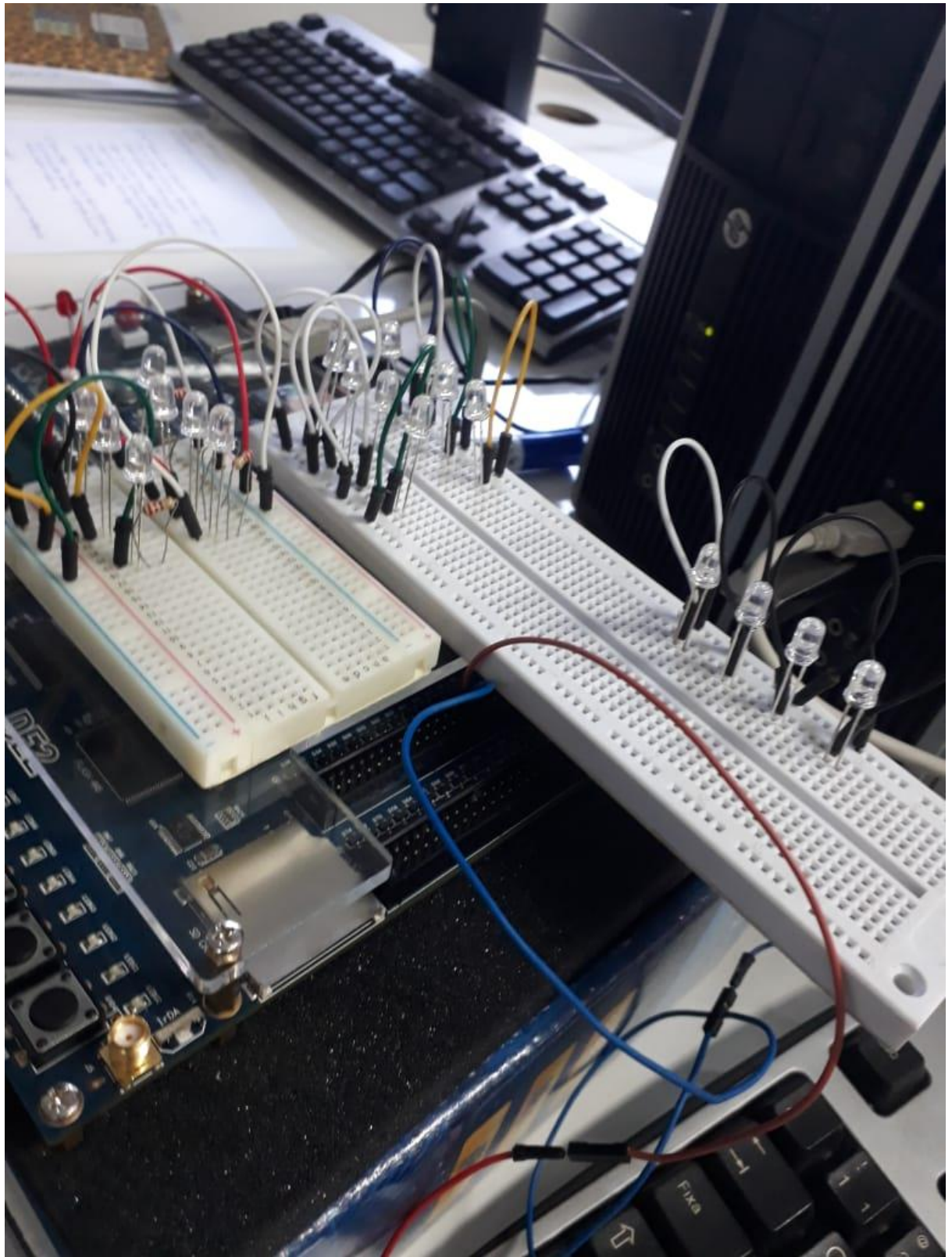
Primeiro foram utilizados os testes utilizando os leds acoplados no próprio FPGA, quando concluído a máquina de estados, os leds externos foram montados na protoboard e após estudado o funcionamento do gpio foram colocados os leds externos na protoboard e ligados os mesmos no gpio. Como ilustrado a seguir, nos testes em que apenas o primeiro estado estava ligado no gpio, em seguida teste com todos os leds ligados ao gpio.











## Conclusões

Através da realização do projeto obtivemos uma melhora significativa no que tange aos conhecimentos relacionados ao uso do FPGA modelo DE2 desenvolvido pela Intel, conhecimentos de leds e seu manuseio, protoboards, calcular e utilizar resistências para o manuseio de resistores, entender correntes e aplicação de tensão. Bem como o aumento no entendimento e realização de códigos na linguagem Verilog. Também a modular utilizando o software desenvolvido pela Altera para simular códigos e baixá-los no fpga.

O jogo se demonstrou de forma satisfazível, pois foi possível a interação do hardware FPGA e o código em Verilog, proporcionando uma experiência ao jogador de controle e de raciocínio para criar estratégias a fim de ganhar o jogo.

Além disso, o projeto permitiu conhecer o GPIO, que foi peça fundamental para a execução do "Mundo do Wumpus" permitindo transmitir a tensão para a protoboard, permitindo que cada estado tenha sua saída individual transferida para cada led representante, fazendo com o que eles ligassem.

# Referências

1. <https://www.badprog.com/electronics-verilog-blinking-a-led-with-gpios>
2. <https://www.ime.usp.br/~leliane/LabVIA/historia.htm>
3. [ftp://ftp.altera.com/up/pub/Altera\\_Material/12.1/Boards/DE2-115/DE2\\_115\\_User\\_Manual.pdf](ftp://ftp.altera.com/up/pub/Altera_Material/12.1/Boards/DE2-115/DE2_115_User_Manual.pdf)
4. <http://www.comofazerascoisas.com.br/como-calcular-o-resistor-adequado-para-um-led.html>
5. [http://www-ug.eecg.toronto.edu/msl/manuals/tutorials\\_v5.pdf](http://www-ug.eecg.toronto.edu/msl/manuals/tutorials_v5.pdf)