

Uno Raid: Prototipação em FPGA de um Vídeo Game Desenvolvido em Linguagem VHDL

Laura Quevedo Jurgina¹, Arthur Piccoli¹, Ítalo Nolasco Ramos¹,
Rafael Iankowisk Soares¹, Leomar Soares da Rosa Junior¹

¹Centro de Desenvolvimento Tecnológico (CDTec)
Universidade Federal de Pelotas (UFPel) – Pelotas, RS – Brasil

Abstract. *This work presents a VHDL video game development and its prototyping in the Altera DE2 platform. The hardware was described and validated in Quartus II tool and synthesized in Cyclone II EP2C35F672C6 FPGA. The system was implemented by Computer Science and Computer Engineering students from Federal University of Pelotas. Results show the system feasibility due to its low-cost hardware implementation, using only 3236 logical elements from the FPGA.*

Resumo. *Este trabalho apresenta o desenvolvimento de um vídeo game em linguagem VHDL e sua prototipação na plataforma Altera DE2. O hardware foi descrito e validado na ferramenta Quartus II e sintetizado para o FPGA Cyclone II EP2C35F672C6. O sistema foi implementado como uma atividade de pesquisa do Grupo PET-Computação e contou com a participação de alunos dos cursos de Ciência da Computação e de Engenharia de Computação da Universidade Federal de Pelotas. Os resultados alcançados demonstram a viabilidade do desenvolvimento do sistema com baixo custo de implementação em hardware, utilizando apenas 3236 elementos lógicos do FPGA.*

1. Introdução

A linguagem VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) foi originalmente proposta pelo Departamento de Defesa dos Estados Unidos, em meados da década de 80, com o intuito de documentar o comportamento dos circuitos integrados que faziam parte dos equipamentos das Forças Armadas americanas. Após o sucesso inicial do uso da VHDL, a sua definição foi disponibilizada para domínio público, o que levou a ser padronizada pelo IEEE (*Institute of Electrical and Electronic Engineers*) em 1987 (PEDRONI, 2010). Atualmente, a VHDL é uma linguagem obrigatória na academia e na indústria para documentar, especificar, simular, e sintetizar circuitos digitais (D'AMORE, 2012).

O desenvolvimento de circuitos digitais complexos vem se tornando popular com a utilização da VHDL. Projetistas de hardware têm explorado a linguagem para desenvolver novos produtos e aplicações, as quais podem ser sintetizadas em dispositivos FPGA (*Field Programmable Gate Array*) relativamente baratos quando comparados a outras técnicas de construção de circuitos integrados. O domínio deste conhecimento é, portanto, fundamental para projetistas que desejam se tornar qualificados em um mercado cada vez mais competitivo. Exemplos de circuitos digitais implementados em VHDL e prototipados em FPGA são os trabalhos de RANGA (2018), de DOMÉNECH-ASENSI (2018), e de SACHDEVA (2018).

Neste sentido, este trabalho consiste no desenvolvimento de um vídeo game utilizando a plataforma Altera DE2 e a linguagem VHDL. Esta plataforma foi escolhida para o desenvolvimento do trabalho em virtude de ser amplamente empregada em atividades de ensino, pesquisa e desenvolvimento de produtos. Além disto, por possuir diversas interfaces de entrada e saída, esta plataforma é ideal para futuras expansões do vídeo game. O projeto foi executado dentro das atividades de pesquisa do Grupo PET-Computação com o intuito de possibilitar o domínio das tecnologias envolvidas.

2. Metodologia

O Uno Raid é um jogo simples e intuitivo que reproduz o conceito de projetos anteriores, onde foram emulados outros jogos desenvolvidos para o Atari 2600 (MONTFORT; BOGOST, 2009) em uma placa FPGA (FLACH, 2012). Foi completamente implementado em VHDL para ser executado no kit de desenvolvimento Altera DE2 equipado com o FPGA Cyclone II. A Figura 1 ilustra a arquitetura da plataforma utilizada. Seu funcionamento é baseado em outros *games* da mesma plataforma, tais como Road Blasters e River Raid. Possui jogabilidade individual e seu objetivo é dirigir um automóvel por uma estrada - dividida em três pistas - com vários obstáculos, elevando a pontuação sempre que ultrapassá-los com sucesso. O crescimento do *score* é proporcional à velocidade da execução, aumentando a dificuldade de acordo com o êxito das metas.

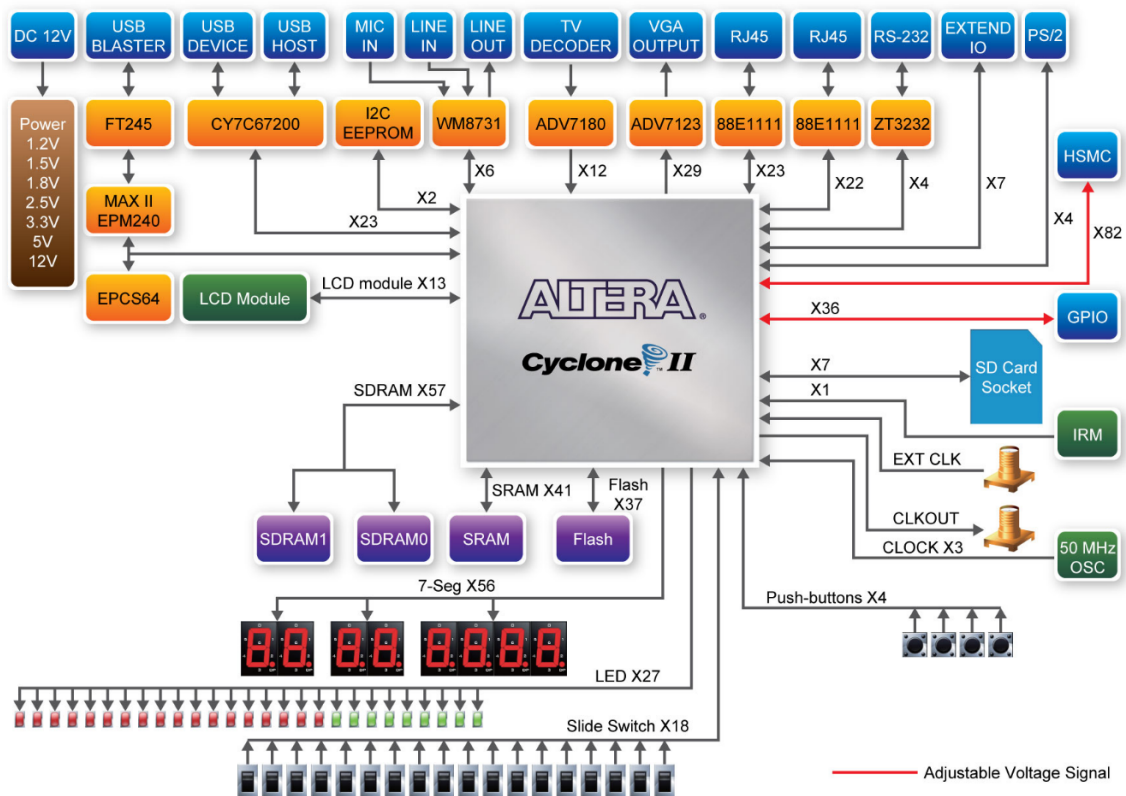


Figura 1. Plataforma Altera DE2 (MACNICA, 2019).

Quartus II (INTEL, 2019) é um software utilizado para descrição de hardware que programa dispositivos FPGA para sintetizar o seu comportamento. As entradas e saídas do sistema acontecem por meio de pinos, com configuração fornecida pelo manual da

fabricante. A plataforma de desenvolvimento Altera DE2 oferece diferentes opções de saída, como, por exemplo, LCD ou display de sete segmentos. Para o Uno Raid a porta VGA foi empregada. A partir de um monitor o jogo é apresentado com a resolução de 1280 x 1024 pixels, com taxa de atualização de 60 Hz.

A frequência necessária para a atualização do pixel na imagem é de 108 MHz. Contudo, o valor fornecido pela Altera DE2 é de 27 MHz. Sendo assim, um multiplicador de frequência foi construído para elevar a frequência original. No desenvolvimento deste conversor fora empregado um plug-in ALTPLL que eleva o *clock* padrão da placa por meio de uma malha de captura de fase (PLL). Após a compilação da descrição do hardware para o FPGA Cyclone II EP2C35F672C6 presente na Altera DE2, 3236 elementos lógicos, 198 registradores, 41 pinos e 1 PLL foram necessários para a implementação total do vídeo game. A Figura 2 apresenta a interface do software Quartus II com parte da descrição em VHDL do código implementado. A Figura 3 apresenta o mapeamento dos elementos lógicos e a pinagem de entrada e saída realizada para a prototipação do sistema completo.

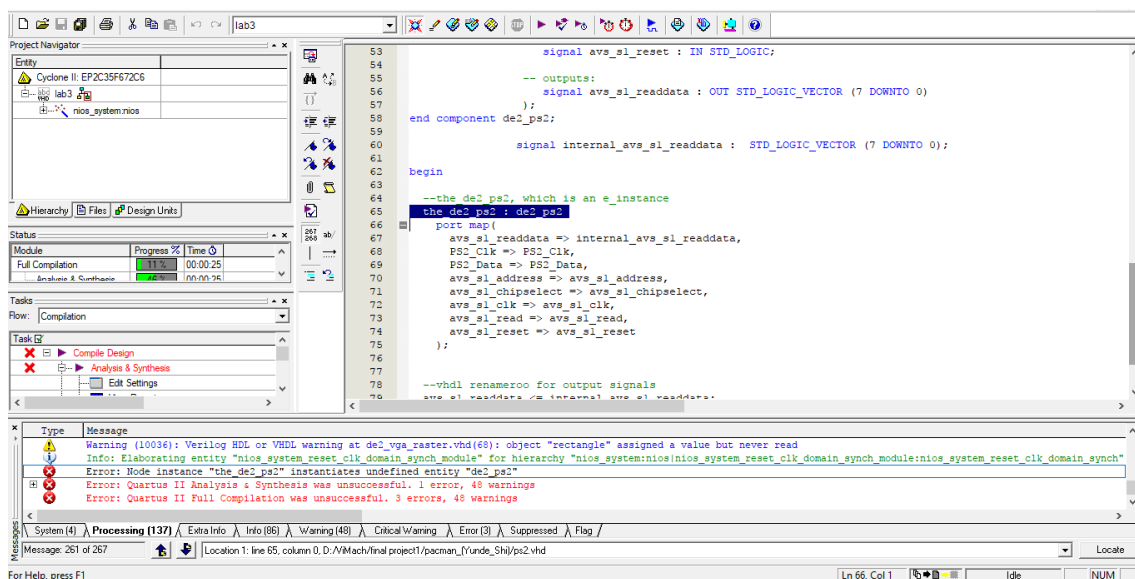


Figura 2. Interface do software Quartus II.

A parte de controle dos elementos do jogo foi desenvolvida utilizando uma máquina de estados finita (SEHN, 2018), com cinco estados, sendo atualizada a cada quadro que compõe a imagem, ou seja, em uma frequência de 60 Hz. A Figura 4 ilustra a máquina de estados finitos do vídeo game.

No primeiro estado a pontuação do jogo é zerada e somente o carro do jogador é exibido na pista central da estrada, com as linhas e o gramado movendo-se lentamente na vertical criando a ilusão de movimento conforme os quadros avançam. Os objetos mencionados são deslocados 4 pixels para baixo a cada quadro, enquanto o estado aguarda que o botão que começa o jogo seja pressionado, ocasionando a mudança para o próximo estado da máquina.

O segundo estado introduz o primeiro adversário na estrada, movendo-se a 4 pixels por quadro, enquanto o gramado e as linhas começam a se mover a 8 pixels por quadro,

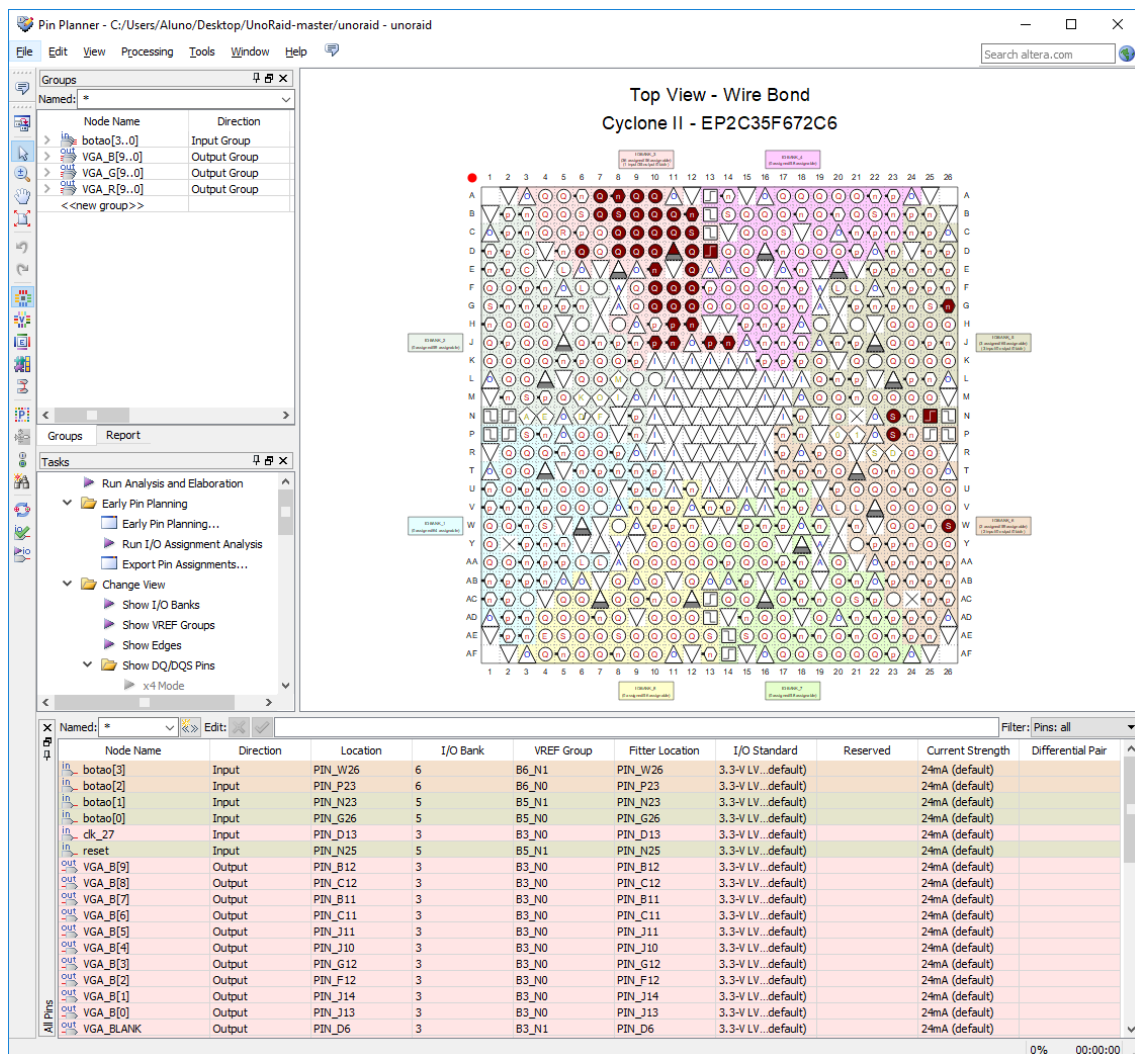


Figura 3. Mapeamento dos elementos lógicos e pinos de entrada e saída para o Cyclone II.

criando um efeito de velocidade maior no carro controlado pelo jogador. Quando a parte superior do carro do adversário alcança a linha 384 da parte visível da tela, o jogo avança mudando para o terceiro estado da máquina.

No terceiro estado, o segundo adversário é introduzido na tela em uma pista diferente da primeira, e, quando atinge a mesma altura que a do anterior, o jogo vai para o quarto estado. No caso de uma colisão dos carros, a máquina vai para o quinto estado.

No quarto estado, o terceiro e último oponente surge na tela deixando todos os três adversários igualmente separados. Assim que um carro ultrapassado deixa o campo da tela, a pontuação do jogo é incrementada em um ponto e sua posição na estrada é alterada, fazendo com que reapareça na parte superior da tela, em uma pista diferente da anterior. A velocidade do jogo é acrescida em 2 unidades de velocidade para o deslocamento do oponente e em 4 para as linhas de limite da pista e cenário toda vez que o jogador alcança as pontuações 5, 6, 15, 16, 50 e 51. Estes aumentos subsequentes existem para criar o efeito de aceleração. O Uno Raid permanece nesse estado até que o jogador cause uma

colisão e, portanto, mude para o quinto estado.

O quinto e último estado, congela a pontuação alcançada pelo jogador e os objetos na tela, esperando que o botão seja pressionado para retornar ao estado inicial. O sistema de colisão funciona quando os sinais de desenhar o carro controlado pelo jogador e o de desenhar um carro adversário estão ativos, avisando a colisão entre os objetos.

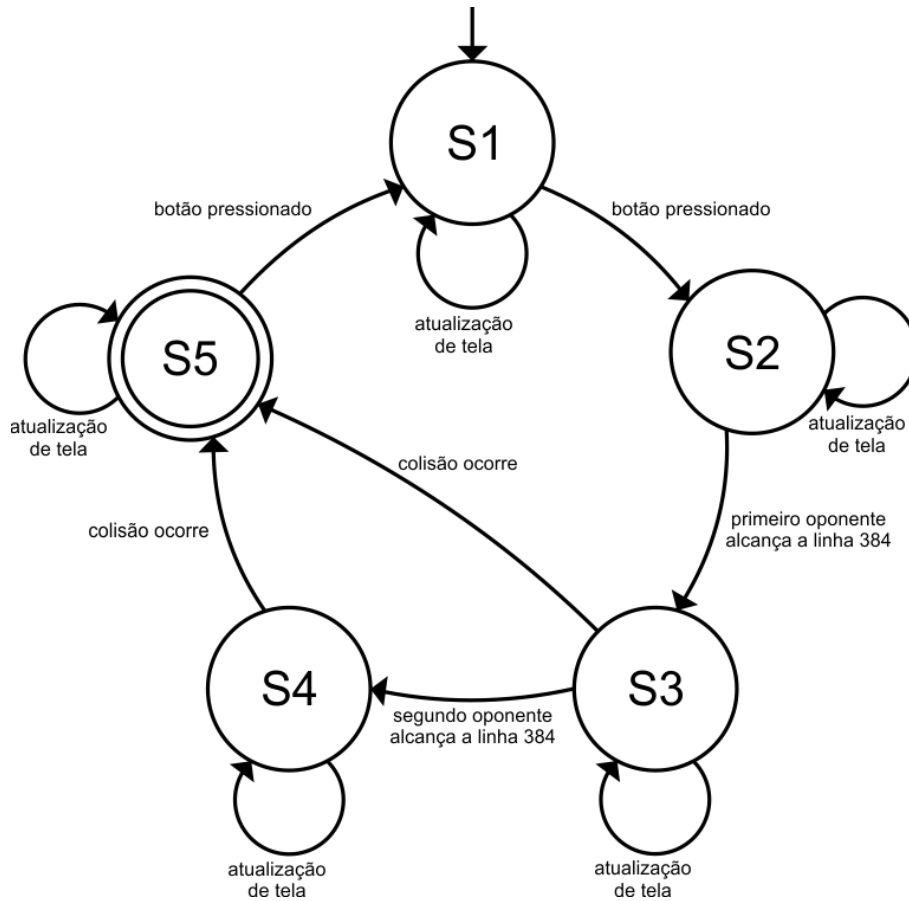


Figura 4. Máquina de estados finita que controla o jogo.

A posição dos oponentes em uma nova partida é definida de acordo com a última assumida na tela, de modo que os novos obstáculos que aparecerão sejam diferentes do instante final do desafio anterior. A localização é gerada a partir de uma pseudo-aleatoriedade a cada recomeço.

A saída VGA funciona como em telas analógicas, onde cada pixel é impresso na tela por meio de uma varredura pixel-a-pixel (SKLIAROVA, 2005). Além dos pixels visíveis, há os de *front porch* e *back porch* e os de sincronização horizontais e verticais. Portanto a resolução total, incluindo os elementos não visíveis, é de 1688 pixels horizontais e 1066 pixels verticais. Para atingir a taxa de atualização de 60 Hz, cada um precisaria ser renovado em uma frequência de 107.964.480 Hz, o que é suficientemente aproximado dos 108 MHz possibilitados pelo PLL.

O jogo tem arquivos de design VHDL auxiliares que ajudam a informar como desenhar os carros e as faixas na tela. Como a única diferença entre o carro do jogador e os carros dos adversários é a ilustração de um acessório anexado ao topo, seus arquivos

são muito semelhantes, com o do jogador tendo mais dois sinais de cor. Ambos recebem a coordenada do canto superior esquerdo do carro da parte visível da tela e compara se o pixel atual está dentro da área de 144 por 256 pixels que o objeto deve ocupar, enviando dois sinais no caso positivo, sendo um o de desenhar e outro informando qual cor deve ser impressa. Os outros objetos não possuem sinal de cor porque são mais simples.

O contador da pontuação tem uma entrada específica além daquelas comuns aos outros objetos cuja é responsável pelo valor que deve exibir. O cálculo é realizado pela máquina de estados. Os objetos, das faixas móveis e do gramado, funcionam como os mencionados anteriormente, porém são mais simples pois seu valor horizontal é fixo.

Todos os sinais de controle entram em estruturas condicionais encadeadas, onde são convertidos em valores RGB que estão pinados à saída VGA. Os elementos restantes, que são fixos e não têm arquivos separados - como o asfalto, as faixas fixas e o fundo do gramado - são codificados diretamente na cadeia condicional. Se todos os sinais de desenhar estiverem desligados e o pixel não fizer parte dos elementos fixos, significa que o mesmo está fora da parte visível da tela, de modo que preto puro é enviado para a saída VGA. Depois que cada pixel recebeu um valor RGB, um quadro é formado e a sequência de quadros forma a imagem do jogo, neste caso, em um monitor que a saída VGA está conectada, como ilustrado na Figura 5.

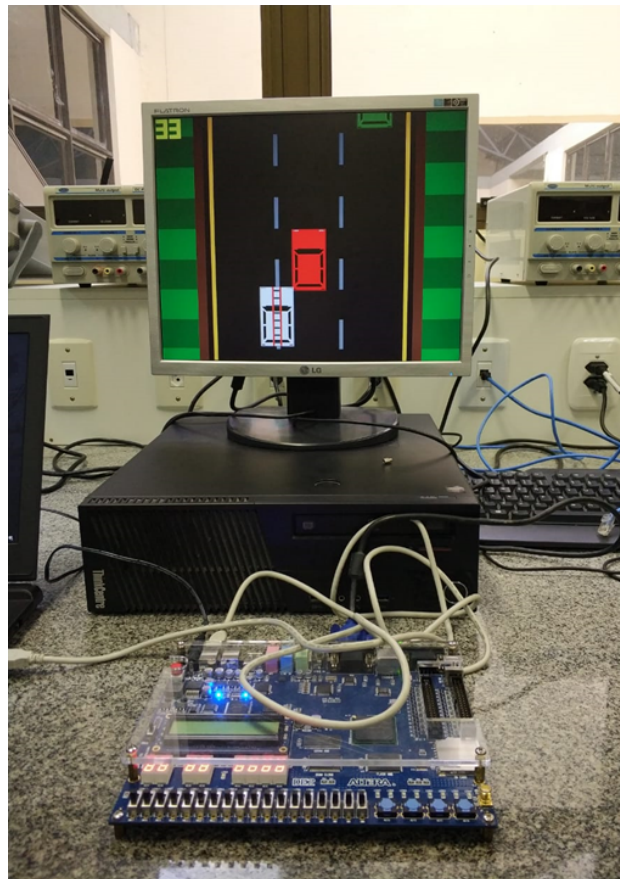


Figura 5. Uno Raid sendo executado na Altera DE2.

3. Resultados e Discussão

Para que seja simples e intuitivo, o carro do jogador pode mover-se com diferentes velocidades. O começo é mais lento, e segue assim até a quinta ultrapassagem, quando ocorre a primeira aceleração. É possível considerar a velocidade inicial como o nível mais fácil do jogo, em virtude do tempo de resposta que o jogador possui para ultrapassar os adversários. A segunda velocidade é duas vezes mais rápida que a anterior e é a dificuldade média do jogo, durando até a décima quinta ultrapassagem, quando ela triplica tornando-se mais desafiante. Ao superar o quinquagésimo obstáculo a velocidade do jogo é quadruplicada e segue assim até que haja uma colisão.

Depois de carregar o jogo na memória interna da Altera DE2, o jogador terá quatro botões para utilizar, as imagens aparecerão em seu monitor, a tela inicial é composta do carro controlável em movimento na estrada do jogo, até que seja pressionado o quarto botão, recebendo o comando do automóvel. O jogador poderá usar os dois primeiros botões para mover o carro horizontalmente e assim ultrapassar os obstáculos e evitar colisões.

A pontuação do jogo é incrementada a cada ultrapassagem realizada. Um placar, de dois dígitos, localizado no canto superior esquerdo da tela, informa ao jogador quantos obstáculos foram ultrapassados. O jogo chega ao fim quando o jogador colidir seu carro com outro veículo.

4. Conclusões

A partir de um conceito simples, Uno Raid cumpre o propósito de ser uma aplicação funcional em uma placa FPGA. Possui jogabilidade desafiadora que mantém o usuário interessado, fazendo-o tentar superar seus próprios resultados. Em seu desenvolvimento, o *game* instiga o *designer* a compreender o funcionamento da sincronização e impressão de imagens, além do uso de técnicas avançadas de VHDL e organização de etapas para construção de um jogo, dividindo o projeto em objetos, mapa, máquina de estados, entre outros. Uno Raid também abre espaço para novos recursos, como o controle através de um teclado e efeitos sonoros para acompanhar os carros.

Com a implementação do Uno Raid e com o domínio das tecnologias envolvidas, o Grupo PET-Computação pretende, como trabalhos futuros, dar continuidade ao projeto oferecendo cursos de VHDL com foco na criação de jogos. Estes cursos serão dedicados às comunidades acadêmica e externa, através de ações de ensino, pesquisa e extensão contidas no planejamento anual do grupo.

5. Agradecimentos

Os autores agradecem o Programa de Educação Tutorial do Ministério da Educação e as agências de fomento CNPq e FAPERGS pelo apoio no desenvolvimento deste trabalho.

6. Referências

PEDRONI, V. Eletrônica Digital Moderna e VHDL. Editora Campus, 2010.

D'AMORE, R. VHDL - Descrição e Síntese de Circuitos Digitais. Editora LTC, 2012.

RANGA, S.; et. al. An Equivalent Circuit for Flying-Adder Frequency Synthesizer and Its VHDL Implementation. In: INTERNATIONAL SYMPOSIUM ELMAR, Croacia, 2018.

DOMÉNECH-ASENSI, G.; et. al. All-hardware SIFT implementation for real-time VGA images feature extraction. Journal of Real-Time Image Processing, 2018.

SACHDEVA, S.; et. al. Design and Implementation of a Green Traffic Light Controller on FPGA Using VHDL. In: Muttoo S. (eds) System and Architecture. Advances in Intelligent Systems and Computing, vol 732. Springer, Singapore, 2018.

MACNICA. Altera DE2 Kit. Macnica DHW. Acessado em 13 apr. 2019. Online. Disponível em: <http://www.macnicadhw.com.br/produtos/kits/altera-de2115>

MONTFORT, N.; BOGOST, I. Racing the Beam: The Atari Video Computer System. Cambridge, London: The MIT Press, 2009.

FLACH, G.; et. al. Revisiting Atari 2600 on an FPGA. In: VIII SOUTHERN CONFERENCE ON PROGRAMMABLE LOGIC, Bento Gonçalves, 2012. Proceedings... Piscataway: IEEE, 2012.

INTEL. Quartus II Web Edition. Acessado em 13 apr. 2019. Online. Disponível em: <https://www.intel.com/content/www/us/en/programmable/downloads/software/quartus-ii-we/121.html>

SEHN, L. R. Computadores e Sistemas Digitais: Conceitos Básicos. Editora Simplíssimo, 2018.

SKLIAROVA, I. Desenvolvimento de circuitos reconfiguráveis que interagem com um monitor VGA. Revista do DETUA, vol. 4, n. 5, p.626-631, 2005.