

UNIVERSIDADE DE SÃO PAULO
Escola Politécnica



PCS3635 – Laboratório Digital I

JDR
Jogo Desafio do Ritmo

Manual de usuário

Professor Dr. Paulo Cugnasca

Bancada B3

João Pedro Arroyo
Rafael de Almeida Innecco

Nº USP: 12550991
Nº USP: 12550535

SÃO PAULO, 07/04/2023

1. Especificação de requisitos:

| | |
|---|---|
| Código: 01 | ▪ Funcional ▫ Não Funcional |
| Requisito: Mostrar jogadas seguintes ao jogador. | Requisitos associados: 03 |
| Descrição: Permitir que o jogador veja as quatro próximas jogadas em uma matriz de leds 4x4. Cada linha corresponde a uma jogada e a linha mais abaixo é a jogada atual. As jogadas mostradas devem ser atualizadas conforme o andar da partida. | |
| Prioridade: ▪ Alta ▫ Média ▫ Baixa | Estabilidade: ▪ Alta ▫ Média ▫ Baixa |
| Rationale: Há duas partes no desenvolvimento do requisito. Primeiro, deve-se alterar o código VHDL do jogo base da memória para que jogadas futuras sejam mostradas de forma contínua ao longo da partida. Os testes da alteração no VHDL podem ser feitos pelo <i>software</i> ModelSim. Em seguida, a programação na placa FPGA e junção de outro componente eventualmente relevante ao projeto deve ser feita para análise do real funcionamento do circuito. | |

Tabela 1: Requisito para mostrar quatro jogadas seguintes em matriz de leds.

| | |
|---|---|
| Código: 02 | ▪ Funcional ▫ Não Funcional |
| Requisito: Mostrar pontuação atual da partida. | Requisitos associados: 03 |
| Descrição: O jogador deve poder ver quantos pontos já marcou ao longo da partida. Ao final, a pontuação atingida deve permanecer à mostra até que algum novo jogo se inicie ou seja dado <i>reset</i> . Devido ao valor da pontuação máxima exceder 99 (decimal) são requeridos três <i>displays</i> de sete segmentos. | |
| Prioridade: ▪ Alta ▫ Média ▫ Baixa | Estabilidade: ▪ Alta ▫ Média ▫ Baixa |
| Rationale: A cada acerto a pontuação deve ser incrementada do valor correspondente ao tempo levado para clicar no botão desde o início daquela jogada. A análise da implementação do requisito ocorre em duas etapas: alteração do código VHDL e simulação no Modelsim; teste com a placa FPGA, <i>displays</i> requeridos e eventuais componentes externos previamente implementados. | |

Tabela 2: Requisito para mostrar pontuação em tempo real.

| | |
|--|---|
| Código: 03 | ▫ Funcional ▪ Não Funcional |
| Requisito: Tempo limite por jogada. | Requisitos associados: 04 |
| Descrição: Cada jogada é contabilizada como correta se é feita até um tempo limite. Após este tempo, uma nova jogada é proposta ou há uma temporização até a jogada seguinte, a depender da música gravada. | |
| Prioridade: ▪ Alta ▫ Média ▫ Baixa | Estabilidade: ▪ Alta ▫ Média ▫ Baixa |
| Rationale: Junto deste requisito é implementado o sistema que contabiliza uma pontuação diferente a depender do ponto em que houve acerto. O teste do requisito consiste inicialmente da simulação com o Modelsim; em seguida deve-se testar na placa, para chegar se o funcionamento é mantido como o esperado na escala de tempo real (aproximadamente 1 segundo por jogada) do jogo. | |

Tabela 3: Requisito para implementar tempo limite por jogada.

| Código: 04 | <input type="checkbox"/> Funcional <input checked="" type="checkbox"/> Não Funcional |
|---|--|
| Requisito: Permitir jogada simultânea (dupla). | Requisitos associados: Não há. |
| Descrição: Deve ser possível fazer até duas jogadas por vez. | |
| Prioridade: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa | Estabilidade: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa |
| Rationale: A implementação deste requisito exige alteração no componente <i>edge detector</i> ou na forma de usá-lo. Portanto, deve-se ter especial cuidado com o funcionamento do circuito quando os botões são de fato pressionados por um humano, visto que o tempo de duração excede em muito o do <i>clock</i> . Testes com a FPGA são, assim, de grande importância. | |

Tabela 4: Requisito para permitir jogada com mais de um botão.

| Código: 05 | <input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Não Funcional |
|---|--|
| Requisito: Gerar som correspondente à tecla apertada. | Requisitos associados: Não há. |
| Descrição: Quando o jogador pressiona certa tecla, o som correspondente deve ser gerado. Será usado um <i>buzzer</i> para isso. | |
| Prioridade: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa | Estabilidade: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa |
| Rationale: Um dispositivo externo deve ser usado para produzir o som correspondente quando uma tecla é apertada. Cada botão deve emitir um som de frequência diferente quando pressionado. O teste consiste principalmente de simulações com o Modelsim. Com a FPGA, deve-se checar se as frequências escolhidas são de fato agradáveis. | |

Tabela 5: Requisito para gerar som correspondente à tecla apertada.

| Código: 06 | <input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Não Funcional |
|--|--|
| Requisito: Modo de jogo em que a sequência é criada. | Requisitos associados: Não há. |
| Descrição: Segundo modo de jogo, em que a sequência pode ser gravada e jogada pelo jogador. | |
| Prioridade: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa | Estabilidade: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa |
| Rationale: Quando a gravação é feita, esta música fica disponível até que outra seja guardada em seu lugar. É usada uma segunda memória, então o modo <i>default</i> segue sempre disponível e inalterável. O teste consiste, neste caso, principalmente da simulação com o Modelsim, visto que não há componentes internos e que, devido ao tipo de requisito, dificilmente haveria manifestação de algum problema na placa FPGA que não aparecesse no Modelsim. | |

Tabela 6: Requisito para criar modo de jogo em que sequência é gravada e pode ser jogada.

| | |
|--|---|
| Código: 07 | ▪ Funcional ▫ Não Funcional |
| Requisito: Adiciona seletor de velocidade, aplicável a todos os modos de jogo. | Requisitos associados: Não há. |
| Descrição: Antes da partida, o jogador pode escolher entre dois modos de velocidade: normal e desafiador. | |
| Prioridade: ▪ Alta ▫ Média ▫ Baixa | Estabilidade: ▪ Alta ▫ Média ▫ Baixa |
| Rationale: É um requisito de dificuldade de implementação supostamente menor, mas com impacto possivelmente bastante significativo na experiência do usuário. Devido ao caráter do requisito, deve ser testado extensivamente na placa, posto que o sistema de geração de <i>clock</i> não é o mesmo. | |

Tabela 7: Requisito para criação de um segundo modo de dificuldade.

| | |
|---|---|
| Código: 08 | ▪ Funcional ▫ Não Funcional |
| Requisito: Converter resultado do <i>display</i> para decimal. | Requisitos associados: 02 |
| Descrição: O resultado mostrado nos <i>displays</i> para a pontuação deve estar em decimal. | |
| Prioridade: ▫ Alta ▫ Média ▪ Baixa | Estabilidade: ▪ Alta ▫ Média ▫ Baixa |
| Rationale: Requisito importante para interface homem-máquina. Teste feito principalmente com o Modelsim e <i>testbench</i> específico para o componente. | |

Tabela 8: Requisito para conversão do placar de hexadecimal para decimal.

| | |
|--|---|
| Código: 09 | ▪ Funcional ▫ Não Funcional |
| Requisito: Capturar jogadas com botões externos. | Requisitos associados: Não há |
| Descrição: Implementação de componentes externos para captura de jogadas. | |
| Estabilidade: ▪ Alta ▫ Média ▫ Baixa | Prioridade: ▫ Alta ▪ Média ▫ Baixa |
| Rationale: Requisito importante para interface homem-máquina. Teste feito inteiramente em laboratório com circuito programado na placa, posto que se trata de conexão a componentes externos. | |

Tabela 9: Requisito para capturar jogadas com botões externos.

2. Funcionamento do projeto:

O projeto JDR consiste de um jogo no qual o participante deve tocar as notas musicais indicadas (a partir de botões externos) dentro de intervalos de tempo especificados, mantendo o ritmo da música. Para iniciar a partida, o jogador deve, primeiro, selecionar o modo de jogo e a dificuldade desejados. Em seguida, deve clicar no botão correspondente a “jogar” e será dado início. Caso queira parar a partida atual pode clicar no botão de *reset*. Abaixo tem-se uma foto indicando a posição destes botões na placa FPGA e uma tabela com o significado de cada um.

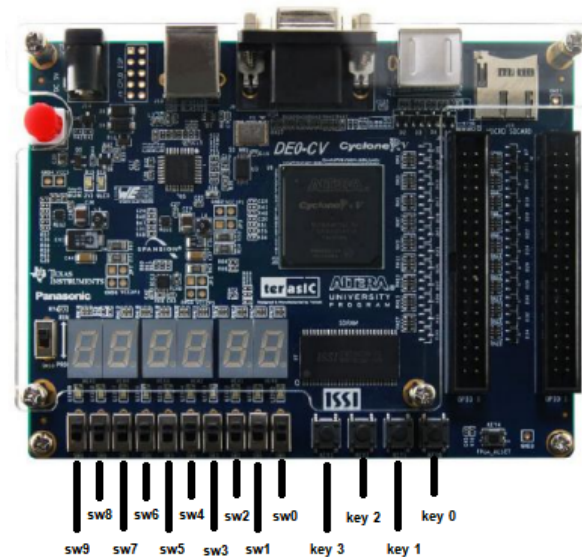


Figura 1: Posição dos botões (*keys*) e *switches* (sw) na placa FPGA.

| Sinal | Função | Pino na placa |
|--------------|---|---------------|
| <i>reset</i> | Aborta uma partida prematuramente, retornando ao estado de espera por um jogo. | key 0 |
| jogar | Inicia uma partida correspondente ao modo de jogo definido pelo seletor. | key 1 |
| seletor(0) | Caso seja '1', está selecionado o modo para gravação de uma nova música. Se '0' significa que se está jogando uma música - a pré-gravada ou a gravada. | sw0 |
| seletor(1) | Seleciona a memória cuja sequência será jogada. Se o sinal acima for '1' este precisa ser '0'. Caso contrário, pode-se escolher '0' - jogar com música gravada - ou '1' - jogar com a música pré-gravada. | sw1 |
| seletor(2) | Seleciona a dificuldade do jogo: '0' para o modo normal e '1' para o desafio. | sw2 |

Tabela 10: Sinais de *input* da interface externa do circuito e modo de operá-los.

As jogadas a serem feitas são indicadas por meio de uma matriz de leds quatro por quatro. A linha mais inferior mostra a jogada que deve ser feita no

momento, enquanto as três acima mostram as jogadas seguintes, no sentido de baixo para cima. Desse modo, a cada janela de tempo para uma jogada que é findada, as jogadas descem uma linha na matriz de leds. A segunda linha mais inferior se torna a mais inferior - jogada atual -, a linha 3 de baixo para cima se torna a linha 2, e assim por diante, com uma nova jogada futura sendo introduzida na linha 4. Os botões, mostrados no canto inferior direito da figura 2, têm correspondência aos leds da seguinte forma: o botão mais à esquerda deve ser pressionado se o led verde da última linha for acendido; o segundo botão da esquerda para a direita deve ser pressionado se o led vermelho for acendido; o terceiro botão da esquerda para a direita deve ser pressionado se o led amarelo for acendido; o botão mais à direita deve ser pressionado se o led azul for acendido. No canto superior direito, tem-se três *displays* de sete segmentos, que mostram, em decimal, a pontuação do jogador em tempo real. Nota-se que uma jogada correta realizada em tempo ótimo vale 4 pontos, correta realizada em tempo mediano vale 2 pontos e correta realizada em tempo atrasado vale 1 ponto. Erros ou não pressionamentos de botões não causam mudança na pontuação. Após a figura, na tabela 11, detalham-se os tempos considerados para cada pontuação em cada modo.

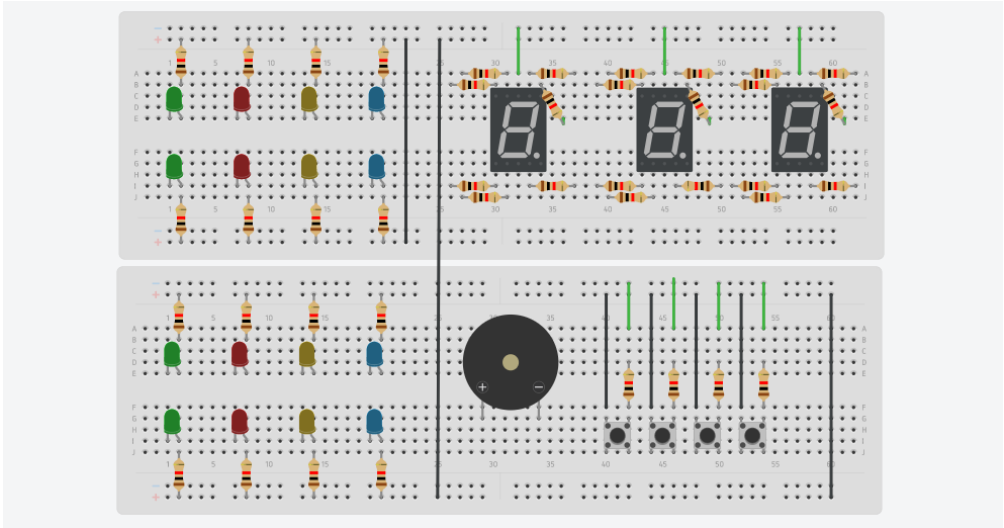


Figura 2: Visão superior do circuito, com componentes externos e sem fios conectados a GPIOs da FPGA.

| Modo de jogo | Intervalo de tempo para marcar 4 pontos | Intervalo de tempo para marcar 2 pontos | Intervalo de tempo para marcar 1 ponto |
|--------------|---|---|--|
| normal | 0 a 0.5 segundo | 0.5 a 0.75 segundo | 0.75 a 1 segundo |
| desafio | 0 a 0.2 segundo | 0.2 a 0.4 segundo | 0.4 a 0.6 segundo |

Tabela 11: Especificação da relação entre pontuação e *delay* da ação do jogador.

Por fim, observa-se que, caso o jogador tente jogar com o seletor em “100” ou “000” sem antes ter gravado qualquer conteúdo na memória, tal ocorrerá com a sequência *default*, ou seja, a mesma coisa que “010” ou “110”. Além disso, no caso da gravação (seletor(0) = ‘0’), em vez de descer o efeito visual aplicado é que as jogadas subam na matriz de leds, mostrando ao jogador as quatro últimas escritas.