



Projeto de Laboratório Digital

Documentação do Projeto

Simon's Dashboard

Bancada B1

Caio Amaral Gurgel Xavier - 11804293

Johan Su Kwok - 10770176

Ricardo de Arruda - 11804463

8 de Abril de 2022

Escola Politécnica da USP

Departamento de Engenharia de Computação e Sistemas Digitais

PCS3635 - Laboratório Digital I

Prof. Edson Midorikawa

1. Objetivo

O presente documento tem como objetivo documentar a especificação dos requisitos do projeto **Simon's Dashboard**, que consiste em adicionar melhorias ao jogo da memória baseado no **Genius** elaborado durante as 5 semanas anteriores à P1 da disciplina Laboratório Digital I.

Com o projeto, o objetivo é avançar os conhecimentos com a utilização do MQTT, da placa FPGA e com a implementação de uma conexão a um dashboard através do Python com utilização da plataforma AWS. Dessa forma, pretende-se criar um sistema que gerará análise de dados de usuários que estarão jogando o jogo da memória.

2. Escopo

O escopo deste projeto é o entretenimento, em que o jogo da memória terá suas modificações para atender funcionalidades adicionais.

No entanto, com a posterior análise de dados e implementações das ideias do grupo, entende-se que o projeto é de grande interesse na área da saúde, especialmente para médicos que buscam entender o funcionamento da memória de pacientes, como no caso de pessoas com Alzheimer, que têm sua memória danificada.

Assim, com uma análise de dados, é possível entender como está o processo de aprendizado desses pacientes e qual a intensidade da memorização em cada momento, podendo realizar testes diferentes para o estudo.

3. Comunicação com a FPGA

Devido ao fato das aulas serem ministradas online, não é possível acessar fisicamente o FPGA, na qual o circuito digital seria implementado. Por isso, foi utilizado o microcontrolador ESP8266, que possui software que possibilita conectar-se à internet. O ESP8266 serve de interface para que possamos mandar mensagens à distância, no conforto do nosso lar, ao FPGA no laboratório da Escola Politécnica, através do protocolo MQTT.

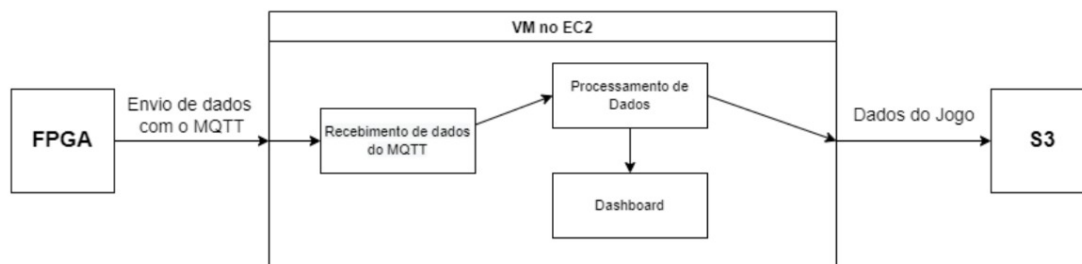


Figura 1: Diagrama do sistema

Em seguida, essas informações seriam recebidas através de mensagens por uma máquina virtual localizada na AWS, utilizando um script escrito em Python. Esse script recebe os dados e encaminha eles para o S3, lá ele pode ser processado e utilizado para analisar o desempenho dos jogadores. Com isso, podemos criar uma dashboard que mostre informações como o número de rodadas, a quantidade de derrotas e vitórias, os tipos de jogos que foram escolhidos ao longo do tempo e muito mais

4. Código VHDL

Desenvolveu-se o circuito digital por meio da linguagem de descrição de hardware VHDL(VHSIC Hardware Description Language) ao longo de 5 semanas. No fim desse período, obteve-se um circuito que assemelha-se o jogo **Genius**, seguindo as seguintes instruções:

1. Para iniciar o jogo, aperte o botão JOGAR.
2. O circuito apresenta a primeira jogada, ativando um dos LEDs de saída.
3. O jogador deve acionar o botão de entrada associado ao LED da jogada anterior.
4. Em seguida, o circuito repete a primeira jogada e acrescenta mais uma jogada nos LEDs.
5. O jogador deve repetir estas duas primeiras jogadas.
6. O circuito repete as duas primeiras jogadas e acrescenta mais uma jogada nos LEDs.
7. O jogador deve repetir estas três jogadas na mesma sequência em que foram apresentadas.
8. O circuito vai aumentar a quantidade de jogadas uma a uma, e o jogador deve continuar a repetir a sequência de jogadas.
9. O jogador vai ganhar o jogo se conseguir repetir a maior sequência, composta por 16 jogadas. Isso é sinalizado pelo sinal de saída GANHOU.
10. Se cometer um erro em qualquer jogada, o jogador vai perder o jogo. Isso é sinalizado pelo sinal de saída PERDEU.
11. Em ambas as situações de término (itens “9” e “10”), o final de jogo também é sinalizado pela saída PRONTO.
12. Depois do final do jogo anterior, para iniciar um novo jogo, basta apertar o botão JOGAR.
13. Se desejar parar um jogo em andamento, basta apertar o botão RESET.

Desejou-se aprimorar o circuito durante as 4 semanas posteriores, adicionando as seguintes features:

- Número aleatório para a jogada, ao invés de ser pré-determinado.
- Seleção de modos de jogo diferentes (aleatório e normal).
- Adição de timeout para cada jogada.
- Modificação do timeout.

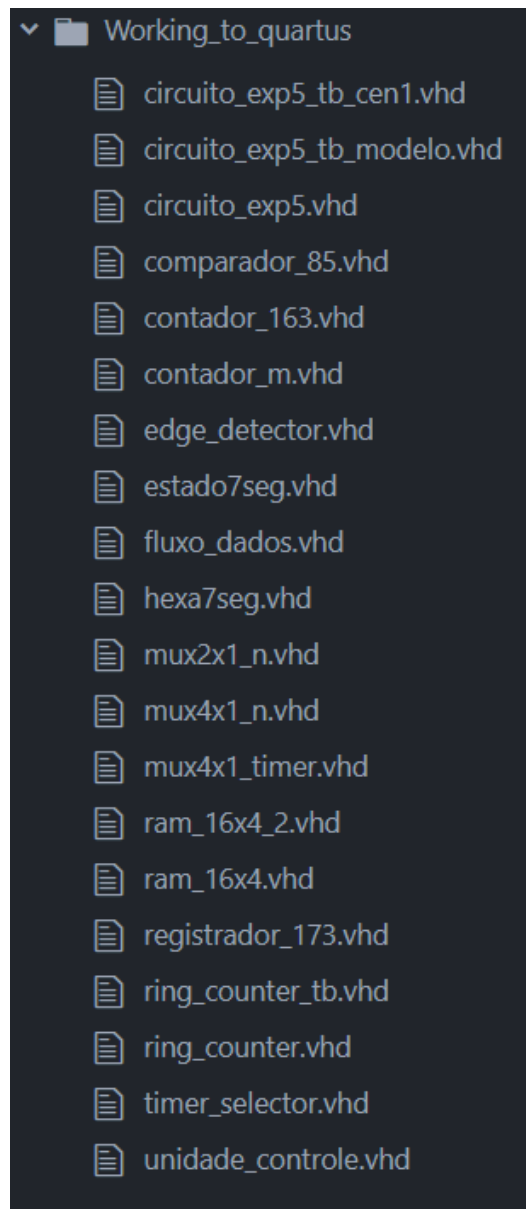


Figura 2: Componentes usados para o projeto

É possível dividir o circuito em duas grandes áreas:

- Fluxo de dados: Parte do circuito que cuida da informação que entra e sai, é onde fica a lógica de contagem de nível, memória de jogadas, timer, entre outros.
- Unidade de controle: Parte do circuito que cuida dos estados do circuito, representando cada etapa possível do que poderia acontecer em seu funcionamento.

E temos o circuito top-level, que realiza a integração desses dois grandes componentes, como pode-se ver no circuito abaixo:

Foi usado o software Quartus para fazer a compilação e carga do circuito no FPGA. O programa também tem a funcionalidade de gerar diagramas como pode ser visto na imagem anterior e na seguinte, que mostra os componentes dentro do fluxo de dados:

Foi usado o software Quartus para fazer a compilação e carga do circuito no FPGA. O programa também tem a funcionalidade de gerar diagramas como pode ser visto na imagem anterior e na seguinte, que mostra os componentes dentro do fluxo de dados:

Documentação do Projeto Simon's Dashboard

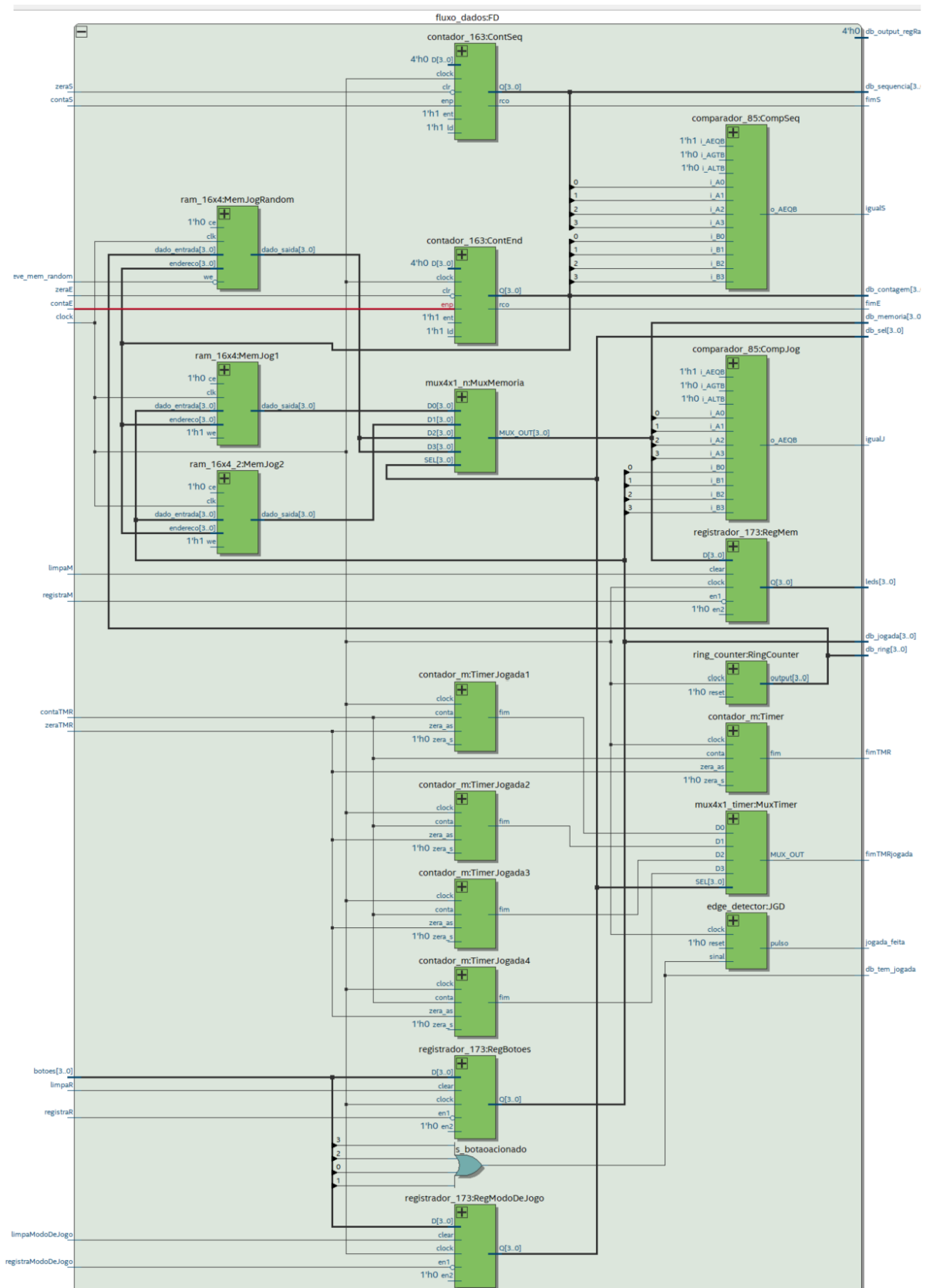


Figura 4: Diagrama de fluxo de dados

São três *RAMs*, duas que armazenam jogadas pré-determinadas, uma que armazena jogadas aleatórias geradas pelo *Ring Counter*. Há *Contadores* para manter a informação do nível atual do jogo e também para a contagem do timeout, e também há *Comparadores* para saber se o jogo ou o nível atual terminou. Também há diversos registradores para armazenar informações como o input do usuário ou o conteúdo da memória apontada pelo contador.

Além disso, podemos observar o diagrama da Unidade de Controle:

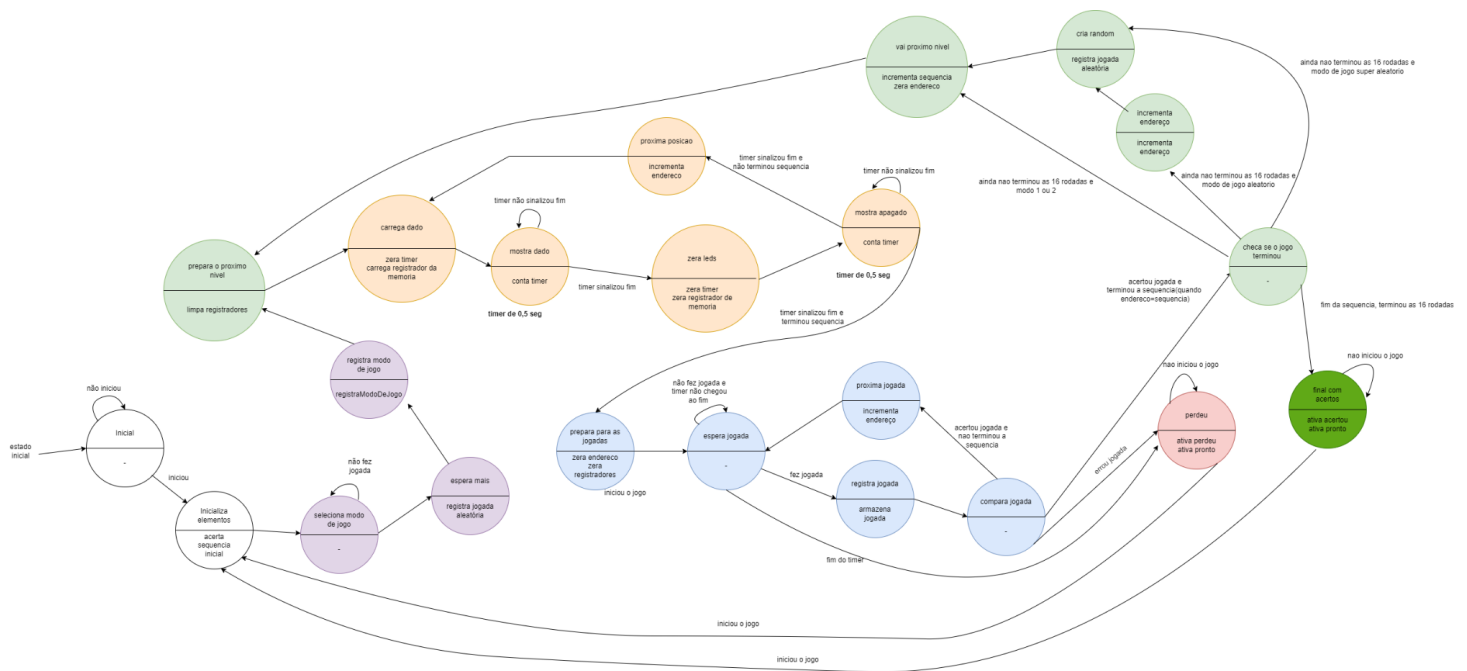


Figura 5: Diagrama de Unidade de Controle

Pode-se separar o diagrama em diferentes áreas, a branca e roxa que inicializa os elementos e cria uma jogada aleatória no caso do jogador escolher o modo de jogo randomizado, a amarela que cuida da ativação dos LEDs para que o usuário veja as jogadas, a azul que é responsável por pegar o input do usuário e fazer comparação e a verde que faz a parte de avançar de nível.

5. Python e AWS

Essa parte do projeto de subdivide em 4 pedaços:

- Configurar a Máquina Virtual
- Configurar o bucket de arquivos do S3
- Receber os dados e salvar eles utilizando um script Python
- Processar os dados e criar a dashboard utilizando outro script Python

Para a Máquina Virtual, utilizamos uma máquina chamada t2.micro, rodando linux com 1 core e 1GB de RAM, da AWS EC2. Nessa máquina vamos rodar os scripts Python

Para o S3, nós criamos um bucket chamado labdigi-filebucket, e configuramos ele para que o acesso fosse público (para que todos possam ver a dashboard).

Documentação do Projeto Simon's Dashboard

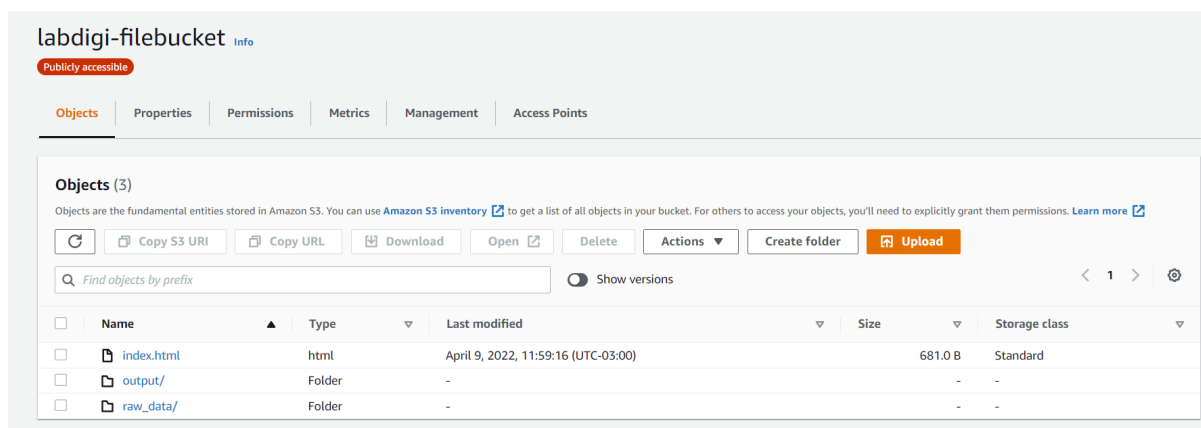


Figura 6: Home do labdigi-filebucket, indicando que ele está público e com os arquivos dentro dele

Depois disso, criamos o script que recebe os dados da FPGA e salva eles no S3. Uma das threads desse script utiliza a biblioteca paho-mqtt para receber os dados da FPGA, enquanto a outra thread salva esses dados no S3 a cada 3 segundos

Finalmente, outro script lê os dados do S3 do jogo mais recente, baixa eles e processa esses dados para que possamos criar nossa dashboard. Esse script atualiza a dashboard com os dados mais recentes a cada 5 segundos. Um dos desafios nessa etapa foi transformar os dados do formato que o MQTT para um formato que pudesse facilitar nossa análise, dado que o MQTT só envia “updates” (uma mensagem de que um tópico específico foi atualizado para determinado valor), por conta disso não conseguimos criar todos os gráficos que desejávamos.

O resultado final pode ser visto [nesse link](#)

	topic	value
time		
2022-04-06 12:35:57.531	grupo2-bancadaB1/E2	1
2022-04-06 12:35:57.853	grupo2-bancadaB1/S4	1
2022-04-06 12:35:58.514	grupo2-bancadaB1/E2	0
2022-04-06 12:37:07.682	grupo2-bancadaB1/E3	1
2022-04-06 12:37:08.060	grupo2-bancadaB1/S0	1
...
2022-04-06 13:20:39.740	grupo2-bancadaB1/S7	1
2022-04-06 13:24:59.917	grupo2-bancadaB1/E2	1
2022-04-06 13:25:00.147	grupo2-bancadaB1/E2	0
2022-04-06 13:25:00.237	grupo2-bancadaB1/S4	1
2022-04-06 13:25:00.377	grupo2-bancadaB1/S7	0

Figura 7: Tabela das informações recebidas por cada tópico MQTT

Documentação do Projeto Simon's Dashboard