

UNIVERSIDADE FEDERAL DO PARANÁ

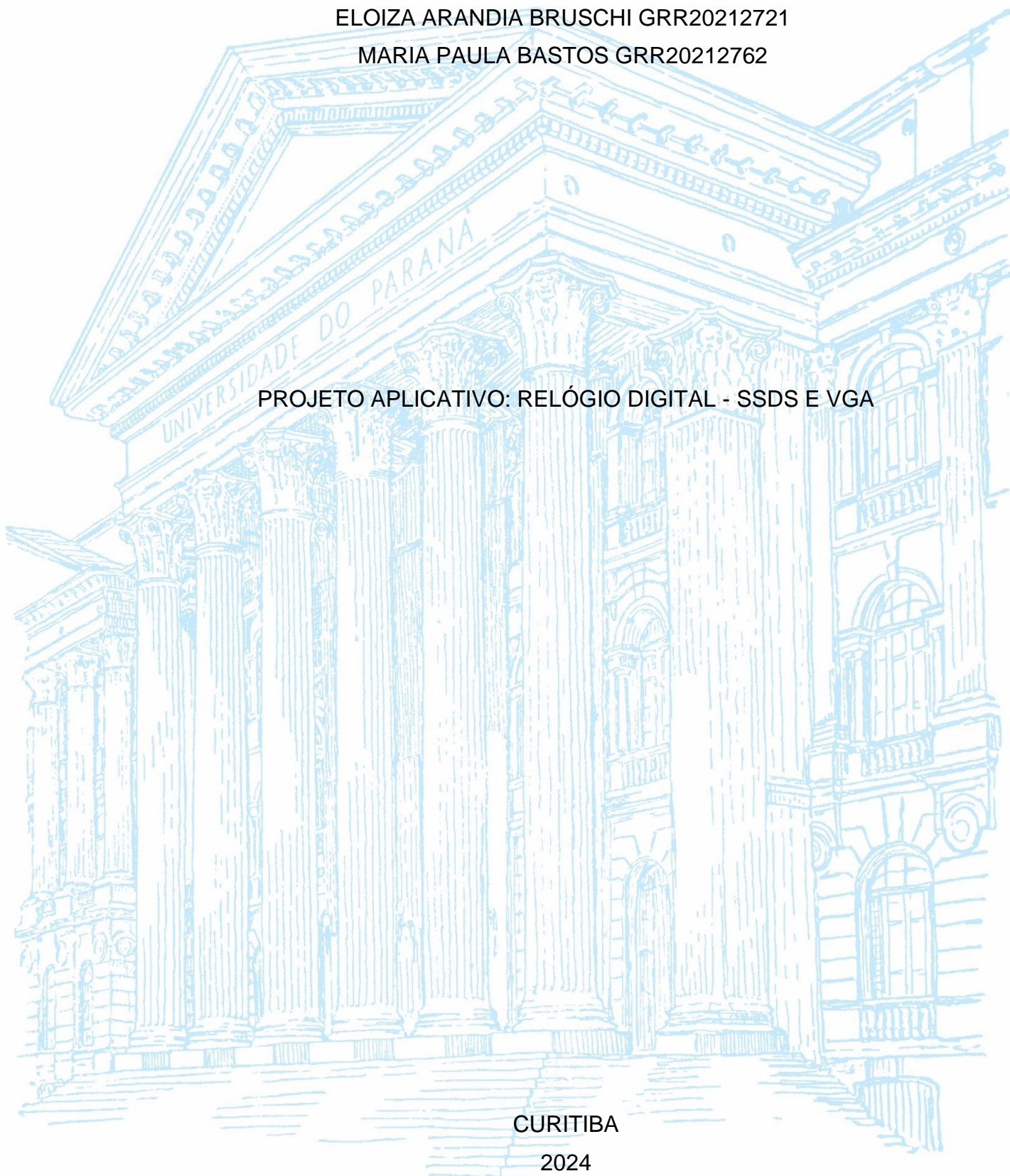
ELOIZA ARANDIA BRUSCHI GRR20212721

MARIA PAULA BASTOS GRR20212762

PROJETO APLICATIVO: RELÓGIO DIGITAL - SSDS E VGA

CURITIBA

2024



SUMÁRIO

1. INTRODUÇÃO	16
2. PROJETO APLICATIVO: RELÓGIO DIGITAL - SSDS E VGA	16
2. 1VGA	17
2. 2DISPLAY DE 7 SEGMENTOS	21
2. 3 SIMULAÇÕES	22
2.4 CÓDIGO COMENTADO	24
2.4.1 VHDL Module	24
2.4.1 VHDL Test-Bench	24
2.5 RECURSOS LÓGICOS	24
CONCLUSÃO.....	16
REFERÊNCIAS.....	17

1. INTRODUÇÃO

Este projeto tem como objetivo a implementação de um relógio digital utilizando o kit Nexys 2 e um monitor VGA. O sistema é composto por displays de 7 segmentos (SSDs) e um monitor VGA, sendo projetado para exibir horas, minutos e segundos. A funcionalidade do relógio digital é controlada por entradas como o botão RST para reinicialização, o botão INICIAR_PAUSAR para iniciar ou pausar a contagem do tempo, e o sinal CONF para alternar entre a exibição de horas e minutos ou minutos e segundos nos displays de 7 segmentos.

No início, ao ligar o dispositivo, será apresentada uma mensagem de inicialização nos displays e no monitor VGA, exibindo o texto "RELOGIO DIGITAL", que se moverá de forma a permitir a visualização completa da mensagem. A partir daí, a contagem do tempo poderá ser iniciada ou pausada pelo botão INICIAR_PAUSAR, com o relógio exibindo o tempo no formato configurado. Caso o botão RST seja acionado, o sistema será reiniciado, voltando para a mensagem inicial. Neste relatório técnico detalhado que incluirá a descrição do desenvolvimento, as simulações das entradas, códigos do design e testbench, bem como a análise dos recursos lógicos utilizados.

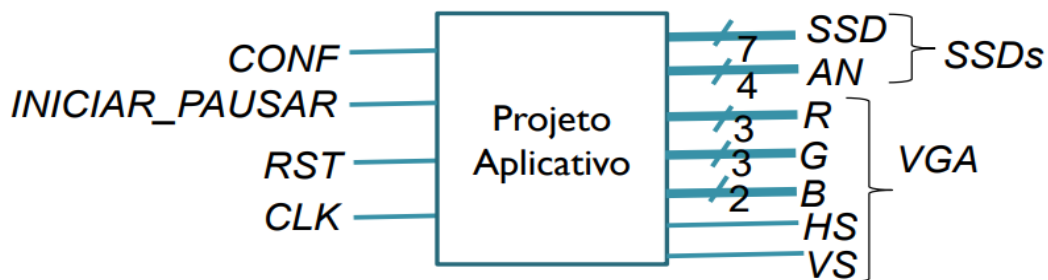
2. PROJETO APLICATIVO: RELÓGIO DIGITAL - SSDS E VGA

A Figura 1 apresenta um relógio digital que exibe horas, minutos e segundos em dois formatos diferentes: nos displays de 7 segmentos (SSDs) e em um monitor VGA. O sistema é controlado por sinais que oferecem funcionalidades como reinício,

pausa e configuração do conteúdo exibido. Quando o sinal RST está ativo em nível lógico alto, o relógio é reiniciado, e todos os valores exibidos são ajustados para zero. Essa operação tem prioridade sobre qualquer outra entrada, garantindo que o sistema seja imediatamente zerado quando necessário.

No monitor VGA, as horas, minutos e segundos são mostrados simultaneamente, enquanto nos SSDs a exibição é configurável. Quando o botão CONF está em '1', os displays de 7 segmentos apresentam as horas e minutos. Com CONF = '0', eles passam a mostrar os minutos e segundos. Além disso, o botão INICIAR_PAUSAR permite iniciar ou pausar a contagem do relógio sem zerar os valores já registrados, garantindo flexibilidade para ajustes e controle do cronômetro. Essa versatilidade na exibição e controle torna o sistema útil para diversas aplicações.

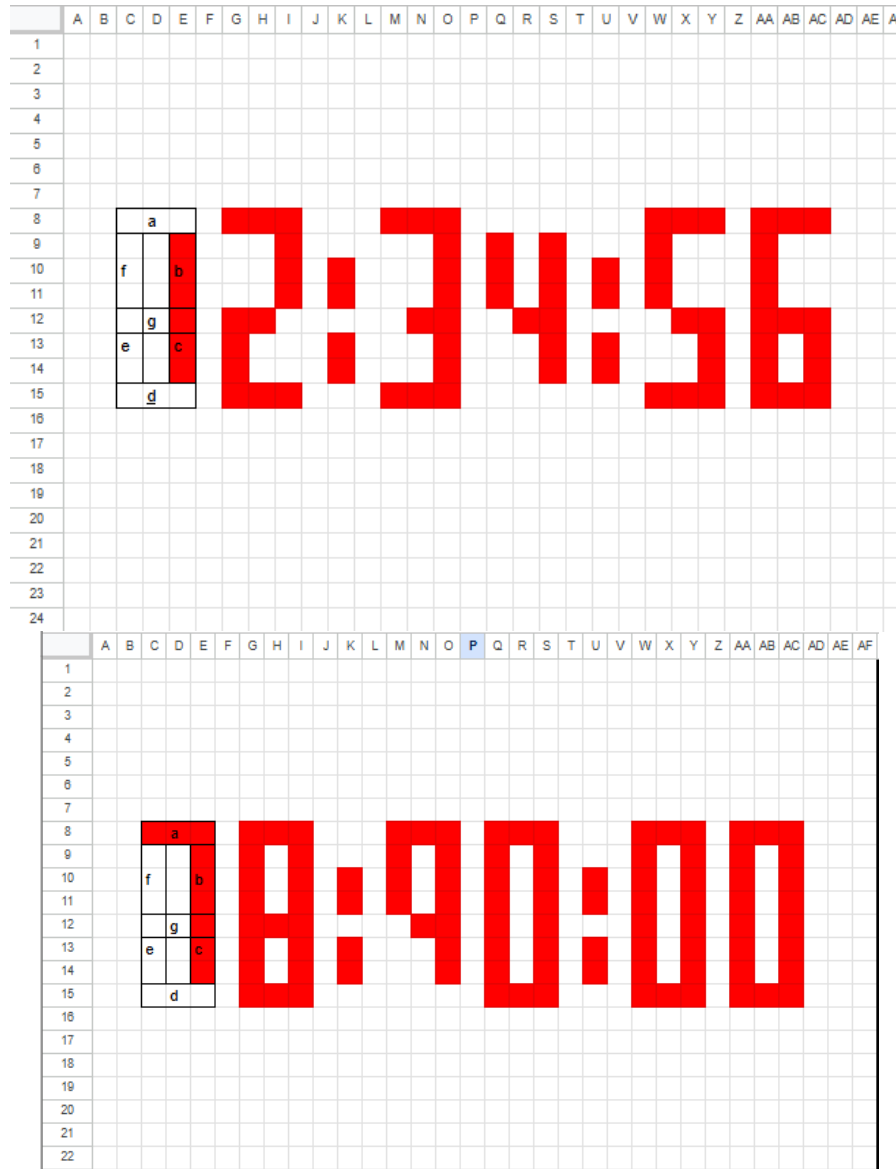
FIGURA 1 – ESQUEMÁTICO CONCEITUAL DO PROJETO



FONTE: 'Projeto Aplicativo', Sibilla França.

2.1 VGA

A implementação de um relógio digital utilizando VHDL em uma tela VGA é baseada em uma lógica em que todos os segmentos possuem o mesmo tamanho, e as informações para desenhá-los são armazenadas em uma matriz de 7x3. Essa matriz contém as coordenadas iniciais de cada segmento, além de um terceiro parâmetro que indica se o segmento está orientado horizontalmente ou verticalmente. A uniformidade no tamanho dos segmentos permite calcular suas dimensões com base em parâmetros fixos, eliminando a necessidade de armazenar posições finais individualmente.



FONTE: As autoras (2024).

Para saber qual segmento está acesso ou desligado utilizou a mesma função que é a base do display de 7 segmentos, é claro que por esses serem anodos, ou seja, '0' significa desligado, foi preciso utilizar o inversor de bit, o 'not'.

Além disso, para escrever 'RELOGIO FIGITAL' também foi feito o mesmo tipo de abordagem que utiliza mapeamento de bit, e foi criado uma grande matriz para projetar o texto no monitor, como é possível entender na figura abaixo, em que novamente cada quadrado é o equivalente a 20x20 pixels.

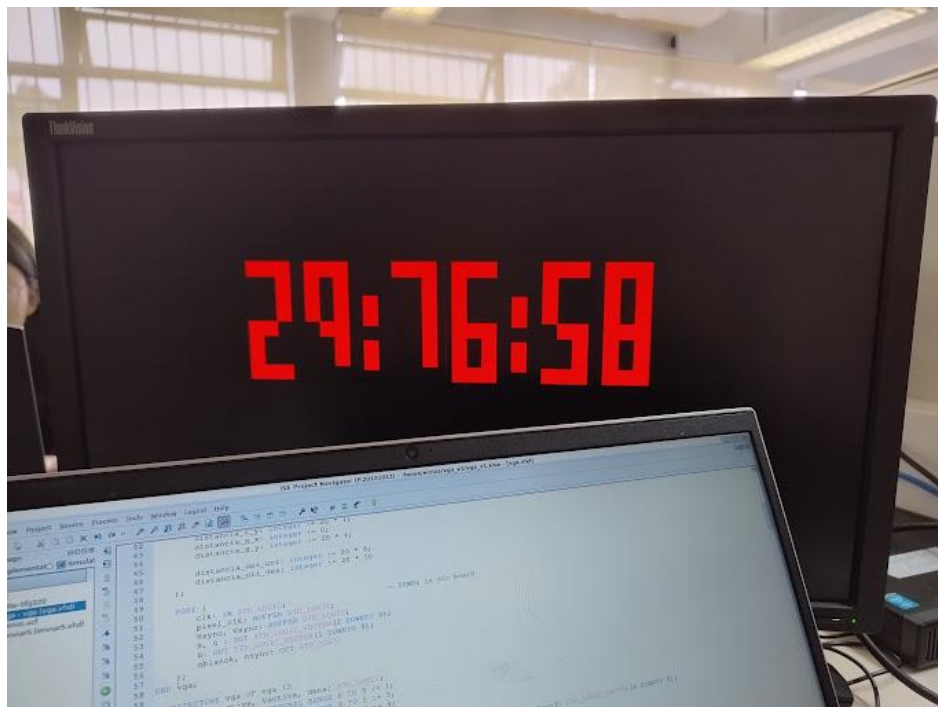
FIGURAS 5 – MAPEAMENTO DE LETRAS

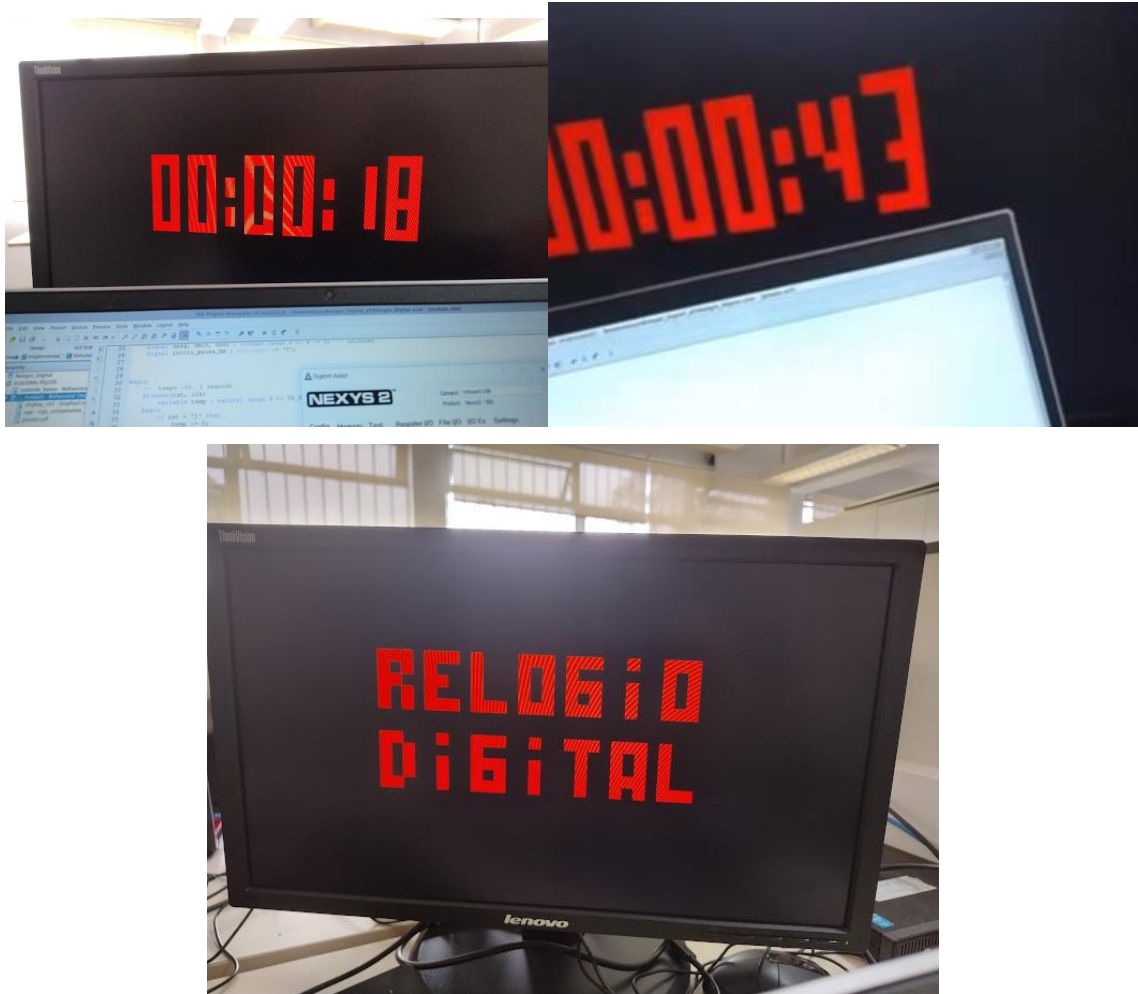
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE		
1																																	
2	(1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	1	1	0	0	1	0	0	1	1	1)				
3	(1	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1)				
4	(1	1	0	0	1	1	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	1	0	0	1	0	1)				
5	(1	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	1	0	0	1	0	0	1	0	1)				
6	(1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	0	1	0	0	1	1	1)				
7	(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0)				
8	(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0)				
9	(1	1	0	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	1	1	1	0	1	0	0)			
10	(1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0)			
11	(1	0	1	0	0	1	0	0	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0)			
12	(1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	0)			
13	(1	1	0	0	0	1	0	0	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1)			
14																																	
15																																	
16																																	

FONTE: As autoras (2024).

Saindo da teoria e aplicando o projeto na prática, os resultados atenderam exatamente às expectativas. Os números exibidos e a escrita das palavras no monitor VGA seguem o formato planejado, com uma apresentação limpa e clara, como pode ser observado nas figuras abaixo. Algo que deve se comentar é que a primeira figura não é do projeto funcionando com o relógio integrado, por isso que os números passam das limitações numéricas de um horário real.

FIGURAS 6,7,8, 9 – RESULTADO PRÁTICO UTILIZANDO A VGA





FONTE: As autoras (2024).

2.2 DISPLAY DE 7 SEGMENTOS

Para fazer o rolamento de texto, se optou por criar uma função com entrada de números inteiros que retornavam 7 bits que escreviam cada uma das letras, ou seja, para escrever “RELOGIO DIGITAL” se usou 14 possíveis entradas, para cada letra e espaço. Para trocar de letra corretamente, colocou um contador de tempo que quando atingisse 10 segundos um valor de index era atualizado, sendo que cada display estava atrasado em relação ao outro. De forma que, enquanto o display 1 estava pescando a letra correspondente ao index atual dentro da função, o display 2 estaria pescando a letra $\text{index} + 1$.

A forma de como manter todos os displays ligados já foi explicada em relatórios anteriores, como foi escrito no relatório 4:

“Como não é possível ativá-los simultaneamente, foi necessário realizar uma lógica que ativa eles em sequência em uma frequência acima de 60Hz, frequência máxima que o olho humano pode processar.

De acordo com o manual da placa, é sugerido que a frequência esteja entre 1000 kHz e 60Hz, mas são 4 displays então foi utilizado como base 4000 kHz, conforme a fórmula abaixo, que mostra que a cada 12500 períodos o display ligado muda, para ligar o próximo.”

A fórmula citada no texto seria essa:

$$\begin{aligned} \text{Quantidade de períodos} &= \frac{\text{frequência do clock}}{\text{frequência do manual} \cdot 4} = \frac{50\text{MHz}}{1000\text{kHz} \cdot 4} \\ &= 12500 \text{ de períodos} \end{aligned}$$

2.3 SIMULAÇÕES

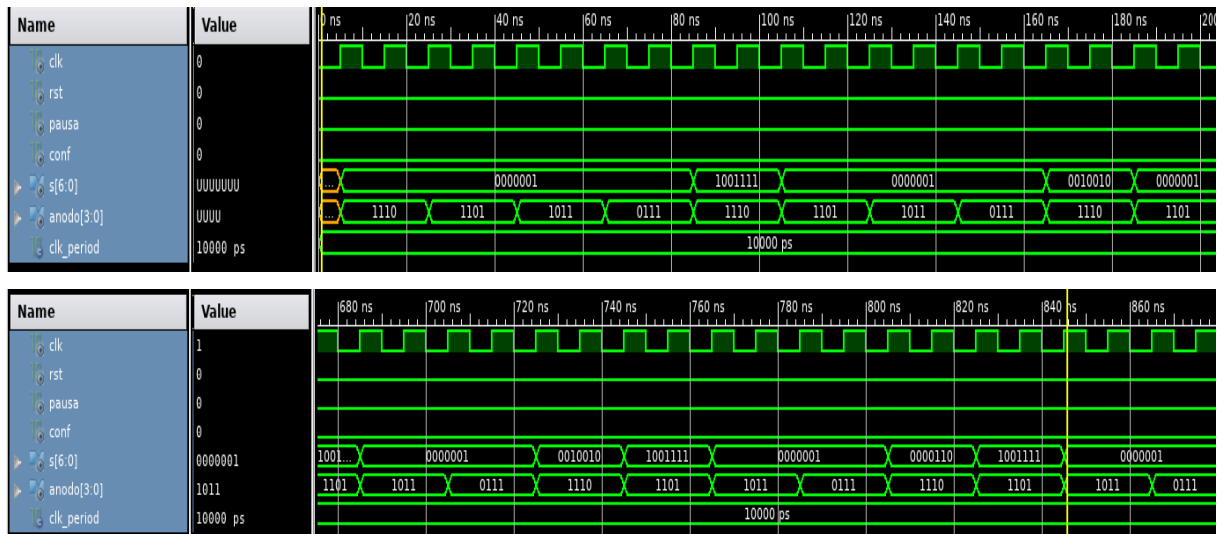
Neste projeto, a simulação foi desenvolvida utilizando um testbench em VHDL, que emula o comportamento das entradas e monitora as saídas do controlador em diversos cenários. Como a simulação do monitor de vga não estava aparecendo nada de útil se optou por simular o funcionamento do resto da programação, principalmente da contagem de tempo.

Para simplificar a simulação, o número de ciclos de clock para a contagem de segundos foi ajustado para 5 períodos, o que possibilitou uma verificação eficaz do funcionamento do relógio digital. A simulação comportamental, apresentada no Anexo 2, foi configurada para exibir em tempo real as alterações nos valores dos segundos e responder às entradas acionadas. Além disso, essa ideia também foi utilizada dentro do debouncing dos botões, para considerar clicado apenas 1 ciclo de clock precisa ser realizado.

A entrada "pausa", quando em nível lógico alto, interrompe a atualização do display; a entrada "reset" trava o relógio em 00:00, zerando todos os contadores; e a entrada "config" altera a exibição de segundos e minutos para horas e minutos. Os resultados da simulação, apresentados na figura abaixo, confirmaram que o temporizador funcionou corretamente nos intervalos de tempo definidos, evidenciando a eficácia do projeto em responder adequadamente às entradas de controle.

Observa-se que o ânodo menos significativo, que representa os segundos, avançou os valores conforme a tabela verdade do display de 7 segmentos, assim como ocorreu a mudança de minutos.

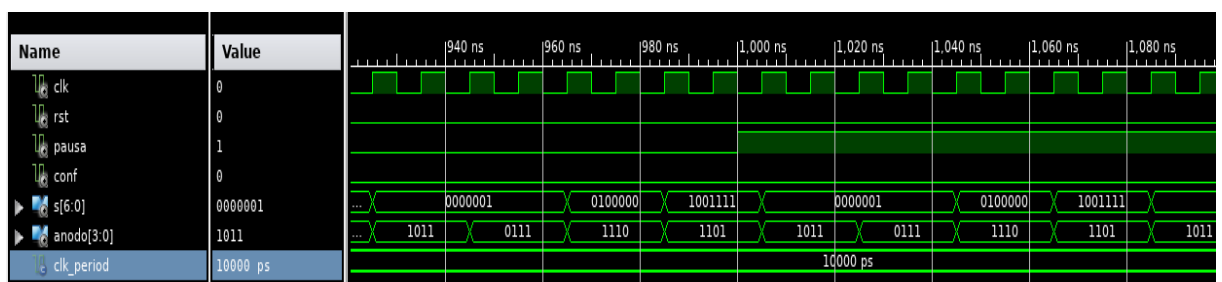
FIGURA 10, 11 – SIMULAÇÃO DO PROJETO



FONTE: As autoras (2024).

O exemplo, na figura abaixo, do funcionamento da pausa durante simulação, em que é possível identificar que após o acionamento da pausa, o contador é 'travado', já que o valor da variável S se torna constante para os 4 valores de display. No projeto finalizado, isso também trava a contagem de tempo mostrado no VGA, já que para a realização do monitor foi criado um componente que recebe os valores de dos caracteres de hora, assim, se pausa dentro dos displays, indica que também foi pausado no monitor. Esse fato continua verdade dentro do reset em relação a contagem de tempo.

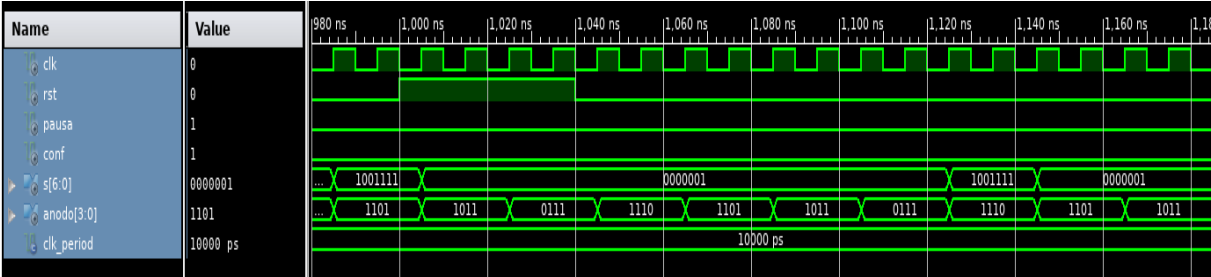
FIGURA 5 – SIMULAÇÃO 2 DO PROJETO



FONTE: As autoras (2024).

O exemplo do funcionamento do reset pode ser encontrado abaixo, em que é possível identificar que após o acionamento do reset, o contador é reiniciado, já que o valor da variável S se torna '0000001' após reset for acionado e desativado que pela tabela verdade do display é são os pontos necessários para formar o caractere 0.

FIGURA 4 – SIMULAÇÃO 2 DO PROJETO



FONTE: As autoras (2024).

2.4 CÓDIGO COMENTADO

O código comentado é dividido em duas partes, no código de fato da placa física e no código para simular, com as mudanças que seriam feitas se fosse possível testar na placa.

2.4.1 VHDL Module

O código VHDL Module, do projeto, está apresentado sem alterações no Anexo 1 deste presente trabalho.


2.4.1 VHDL Test-Bench

O código VHDL Test Bench, do projeto, está apresentado sem alterações no Anexo 2 deste presente trabalho.

2.5 RECURSOS LÓGICOS

Os recursos lógicos resumem a utilização do dispositivo e os recursos do projeto não apareceram, mesmo após vários testes, então abaixo está o consumo lógico do relatório do laboratório 4, que não inclui o consumo do uso da VGA, nem do rolamento de letras no início.

FIGURA 7 – RECURSOS LÓGICOS PROJETO

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	16	9,312	1%	
Number of 4 input LUTs	15	9,312	1%	
Number of occupied Slices	22	4,656	1%	
Number of Slices containing only related logic	22	22	100%	
Number of Slices containing unrelated logic	0	22	0%	
Total Number of 4 input LUTs	41	9,312	1%	
Number used as logic	15			
Number used as a route-thru	26			
Number of bonded IOBs	12	232	5%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	1.82			

FONTE: As autoras (2024).

CONCLUSÃO

Através do uso de displays de 7 segmentos e a integração com o monitor VGA, foi possível criar um sistema funcional capaz de exibir horas, minutos e segundos, com controles para reinicialização, pausa e configuração da visualização. A simulação comportamental demonstrou que o sistema responde corretamente às entradas, incluindo os sinais de controle como "pausa", "reset" e "config", e a contagem de tempo foi precisa e adaptável conforme as configurações selecionadas.

A implementação no monitor VGA e nos displays de 7 segmentos seguiu o planejamento inicial, com a exibição clara das informações e a funcionalidade de rolagem de texto funcionando conforme esperado. As simulações confirmaram o correto funcionamento das entradas e a precisão do temporizador.

Esse projeto não seria possível sem o uso do livro "*Circuit Design and Simulation with VHDL*" do professor-doutor Volnei Pedroni, já que a base utilizada para controlar o VGA foi estruturada com os exemplos presentes no livro.

REFERÊNCIAS

PEDRONI, V. A. Circuit Design and Simulation with VHDL. 2 ed. MIT Press, 2010.