



Trabalho de Lógica Programável

I. Introdução e Motivação para o Trabalho de Lógica Programável

Introdução

Bem-vindos ao estudo da Lógica Programável! Este trabalho prático visa introduzir vocês ao fascinante mundo da síntese digital utilizando dispositivos de Lógica Programável, especificamente CPLDs (*Complex Programmable Logic Devices*) e FPGAs (*Field Programmable Gate Arrays*). Esses dispositivos são componentes essenciais na construção de sistemas digitais modernos, permitindo a implementação de circuitos complexos com flexibilidade e eficiência. Durante este trabalho, vocês irão desenvolver, simular e testar uma série de circuitos digitais básicos utilizando VHDL (*Very High-Speed Integrated Circuit Hardware Description Language*), uma linguagem poderosa para modelagem de hardware.

Motivação

Aplicações em Soluções de Mercado

No mercado de tecnologia, CPLDs e FPGAs são amplamente utilizados devido à sua capacidade de rápida prototipagem e flexibilidade. Empresas de eletrônica e informática empregam esses dispositivos para desenvolver produtos inovadores e de alta performance. Por exemplo, em produtos de consumo como smartphones, tablets e dispositivos IoT (*Internet of Things*), FPGAs são utilizados para processar sinais rapidamente, melhorar a eficiência energética e aumentar a funcionalidade do dispositivo. Além disso, em equipamentos de rede e telecomunicações, esses dispositivos permitem a implementação de algoritmos de processamento de dados de alta velocidade, suportando a infraestrutura de internet e comunicação que utilizamos diariamente.

Aplicações Científicas

No campo científico, CPLDs e FPGAs desempenham um papel crucial em pesquisas avançadas e em aplicações que exigem processamento paralelo e alta precisão. Eles são usados em instrumentos científicos para adquirir e processar grandes volumes de dados em tempo real. Por exemplo, em física de partículas e astronomia, FPGAs são utilizados em telescópios e aceleradores de partículas para processar dados coletados em experimentos complexos, possibilitando descobertas revolucionárias sobre o universo. Em biotecnologia, esses dispositivos ajudam a realizar sequenciamento genético e análises bioinformáticas em velocidades muito superiores às abordagens tradicionais, acelerando a pesquisa e desenvolvimento de novos tratamentos e medicamentos.

Aplicações Militares

Em aplicações militares, a confiabilidade e a capacidade de reconfiguração rápida dos CPLDs e FPGAs são vitais. Eles são empregados em sistemas de radar, comunicações seguras, criptografia, e controle de drones. Em particular, esses dispositivos são cruciais para sistemas de defesa que exigem resposta em tempo real e adaptabilidade a diferentes cenários de combate. As FPGAs, por exemplo, podem ser reprogramadas em campo para responder a novas ameaças ou condições, aumentando a eficácia e a segurança das operações militares. Além disso, a capacidade de processar grandes quantidades de dados em tempo real permite uma melhor vigilância e reconhecimento, dando às forças armadas uma vantagem estratégica.



Finalmente

Ao completar este trabalho, vocês não apenas adquirirão habilidades práticas em VHDL e no uso de ferramentas como o Quartus II, mas também terão uma compreensão mais profunda de como esses conhecimentos são aplicados em indústrias de ponta. O domínio dessas tecnologias pode abrir portas para carreiras emocionantes em diversas áreas, desde a engenharia de produtos eletrônicos até a pesquisa científica e o desenvolvimento de tecnologias de defesa.

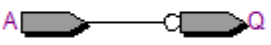
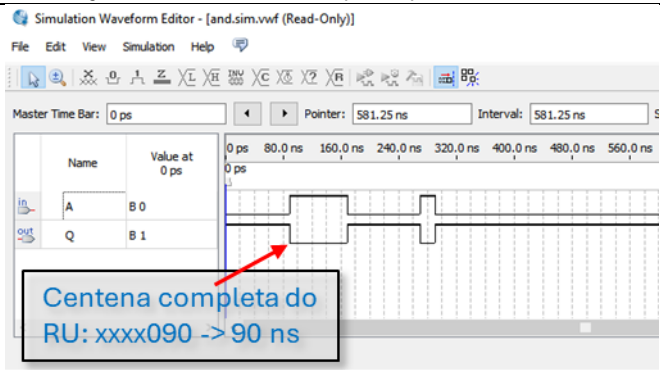

Boa sorte e bons estudos!

II. Instruções Gerais

Os alunos devem seguir as seguintes etapas para cada parte do trabalho:

1. Produzir o código VHDL.
2. Simular o código utilizando o "Waveform" do Quartus II, para todos os estados da tabela-verdade. Os estímulos dos sinais de entrada devem iniciar na centena completa do RU, convertida em ns (nanossegundos): Para o RU: 999999128, deve se utilizar 128 ns.
3. Confirmar se o diagrama RTL gerado está relacionado com o objetivo de cada parte.
4. Testar o circuito sintetizado no kit didático.
5. O trabalho deve conter as partes do roteiro, contendo o resumo de cada parte, o código VHDL comentado, a imagem da simulação do *Waveform*, o diagrama RTL e a sua imagem, com o nome e RU, testando o circuito no kit didático.
6. O arquivo de entrega deve estar em formato PDF.

Exemplo de registro dos dados e informações

Código VHDL (20 %) Obrigatório para cada parte	Diagrama RTL (20 %)
<pre>LIBRARY IEEE; USE IEEE.STD_LOGIC_1164.ALL; ENTITY porta_not IS PORT (A: in STD_LOGIC; Q: out STD_LOGIC); END porta_not; -- RU:999999090 Pedrito Shin da Silva ARCHITECTURE comportamento OF porta_not IS BEGIN Q <= NOT A; END comportamento;</pre>	
Simulação do "Waveform" (20 %)	Fotografia do teste (40 %)
	



Parte 1: Porta AND

Uma porta AND realiza a operação lógica de conjunção, onde a saída é verdadeira apenas se todas as entradas forem verdadeiras.

Parte 2: Porta XOR com Portas Discretas (AND e OR)

Uma porta XOR realiza a operação de disjunção exclusiva, onde a saída é verdadeira se e somente se as entradas forem diferentes.

Parte 3: Decodificador 3:8

Um decodificador 3:8 converte um código binário de 3 bits em um de 8 linhas, onde apenas uma das saídas será verdadeira para cada combinação de entrada.

Parte 4: Multiplexador de Quatro Entradas

Um multiplexador de quatro entradas seleciona uma das quatro entradas para ser passada para a saída com base nos sinais de seleção.

Parte 5: Decodificador de BCD para 7 Segmentos

Um decodificador BCD para 7 segmentos converte um código binário decimal em sinais que controlam um display de 7 segmentos.

Parte 6: Flip-Flop D e JK

Flip-Flops são elementos de memória usados para armazenar bits. O Flip-Flop D possui uma entrada de dados, enquanto o Flip-Flop JK possui entradas J e K que determinam o comportamento de sua saída.

Parte 7: Contador Binário Módulo 16

Um contador binário módulo 16 conta de 0 a 15 em binário, reiniciando para 0 após atingir o valor máximo.

Parte 8: Contador Decimal 00-99 com Reset

Um contador decimal de 00 a 99 conta em decimal e reinicia para 00 após atingir o valor 99, com capacidade de resetar para 00.

Parte 9: Contador Gray de 4 Bits, progressivo/regressivo

Um contador Gray conta em código Gray, que difere do binário normal, pois apenas um bit muda entre estados consecutivos.

Parte 10: Contador Decimal 00-99, progressivo/regressivo com Reset e *Preset*

Um contador decimal que conta de 00 a 99, podendo contar progressivamente ou regressivamente, com capacidade de resetar e definir um valor inicial (*preset*).