RELATÓRIO Prática 2

NOME: Luis Henrique Ponciano dos Santos NUSP: 155777660

NOME: Gabriel Araújo NUSP: 14571376

NOME: Gabriel Demba NUSP: 15618344

Parte 1: Contador de 8 Bits

Visão geral
- Função: Implementar um contador de 8 bits que divide a contagem em dois nibbles para exibição em displays de 7 segmentos.
- Operações: Cada bit do contador é armazenado em flip-flops tipo T, interconectados com portas AND para controlar a ativação dos próximos estágios.

Descrição do código
O contador é composto por oito flip-flops tipo T conectados em cascata. Cada flip-flop recebe um sinal de habilitação baseado no estado do flip-flop anterior. Os resultados são divididos em nibbles alto e baixo, que são decodificados para exibição em dois displays de 7 segmentos.

Simulações
- O funcionamento do contador foi validado para entradas de clock e sinais de reset.
- Observações: A contagem é exibida corretamente nos displays de 7 segmentos.

Parte 2: Contador com 4 Displays de 7 Segmentos

Visão geral
- Função: Implementar um contador de 16 bits com exibição do valor em 4 displays de 7 segmentos.

Descrição do código
O contador de 16 bits utiliza flip-flops para armazenar os valores binários. O valor do contador é incrementado em cada borda de subida do clock, e a saída é dividida em 4 nibbles, que são decodificados para exibição nos displays de 7 segmentos.

Simulações
- O contador foi simulado e exibiu os valores corretamente nos 4 displays.
- Resultados: A contagem incrementa corretamente e os sinais de reset funcionam como esperado.

Relatório - Parte 3

Descrição do Projeto

O módulo **Part3** implementa um contador de 16 bits que utiliza um clock de 1 segundo para gerar pulsos e controlar a contagem. O valor do contador é dividido em 4 nibbles, os quais são exibidos em displays de 7 segmentos. Além disso, há a funcionalidade de ativar um LED quando o contador atinge o valor 30.

O design utiliza os seguintes componentes e funcionalidades principais:

Divisor de Clock:

Um componente denominado Clock_Divider_1s foi utilizado para reduzir o clock de entrada de 50 MHz para pulsos de 1 segundo.
Esse clock gerado controla as operações do contador, garantindo a precisão na contagem.

Contador:

Um contador de 16 bits (Q) foi implementado usando operações aritméticas fornecidas pelo pacote IEEE.NUMERIC_STD.
O contador é resetado pelo sinal clear e é incrementado no pulso ascendente do clock de 1 segundo, desde que o sinal enable esteja ativo.

Divisão de Nibbles:

O valor de 16 bits do contador é dividido em quatro nibbles de 4 bits:
nibble1 representa o valor menos significativo.
nibble4 representa o valor mais significativo.

Displays de 7 Segmentos:

Quatro componentes display foram instanciados para converter cada nibble em sinais para os displays.
Cada nibble foi conectado aos displays correspondentes (seg1, seg2, seg3, seg4).

LED de Saída:

Um processo adicional controla o LED, que é ativado quando o contador atinge o valor 30 e apagado para outros valores.

Relatório - Parte 4

O módulo **Part4** implementa um sistema para exibir a palavra "dE0" de forma rotativa em quatro displays de 7 segmentos. A rotação dos caracteres é controlada por um contador de posição que utiliza um clock dividido para gerar pulsos de 1 segundo.

Arquitetura do Código

O design faz uso dos seguintes componentes principais:

Divisor de Clock:

Um componente clock_divider é utilizado para gerar pulsos de 1 segundo a partir do clock de entrada.
Este pulso controla a atualização do contador de posições.
Contador de Posição:

O componente position_counter é responsável por alternar entre as quatro posições possíveis (representadas por dois bits).
Cada posição determina como os caracteres serão exibidos nos displays.
Caracteres da Palavra "dE0":

Os caracteres "d", "E" e "0" são codificados em formato binário (char0, char1, char2), enquanto a quarta posição exibe um espaço em branco ("0000").
Displays de 7 Segmentos:

A função get_segment_value converte os valores binários dos caracteres em sinais compatíveis com os displays de 7 segmentos.
Os quatro displays (seg0, seg1, seg2, seg3) exibem os caracteres conforme a posição atual do contador.
Rotação dos Caracteres:

A rotação é implementada utilizando um processo que verifica o valor do contador de posição (pos) e distribui os caracteres nos displays de acordo com as combinações possíveis.

Relatório - Parte 5

O código para parte5 é uma extensão lógica da parte anterior, utilizando um contador de posição de 3 bits para permitir a rotação de até 6 displays de 7 segmentos. O objetivo é exibir a palavra "dE01" nos displays, com rotação cíclica em diferentes posições.

Aqui estão algumas observações e explicações detalhadas sobre o funcionamento do código:

Principais Componentes

Divisor de Clock (clock_divider):
Gera um pulso a cada segundo para controlar a atualização dos displays.
Contador de Posição (position_counter):
Contador de 3 bits que determina a posição atual de exibição.
Função get_segment_value:
Converte os valores binários de caracteres (como "d", "E", "0", "1") no formato compatível com displays de 7 segmentos.

Exibição de "dE01":

Os caracteres são codificados como char0, char1, chard e chare.
Um valor adicional charnulo ("1111111") é usado para representar o estado apagado do display.

Processo de Rotação:

Dependendo do valor do contador pos, a sequência "dE01" é exibida de forma rotativa pelos 6 displays. Por exemplo:
Quando pos = "000", os primeiros 4 caracteres são exibidos nos displays iniciais (seg0 a seg3), com os displays restantes apagados.
Quando pos = "001", os caracteres são deslocados para os próximos displays, mantendo o comportamento rotativo.