Aulas 4 e 5 – Contadores:

Parte 1: implementamos um contador síncrono de 8 bits utilizando flip-flops T pelo diagrama de blocos do Quartus, simulamos e verificamos o circuito e mapeamos as entradas para chaves, botões e displays de 7 segmentos de uma FPGA. Os blocos de flip-flops T foram gerados no Quartus e implementados com VHDL, enquanto os blocos “displaysegmentos” foram gerados em um circuito criado por diagrama de blocos.

Diagrama

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Excel

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Diagrama, Esquemático

Descrição gerada automaticamente

Bloco “displaysegmentos”

Parte 2: implementamos um contador de 16 bits utilizando a operação de incremento do VHDL, e simulamos no Quartus com a exibição das saídas do contador em um display de 7 segmentos.

Tabela

Descrição gerada automaticamente com confiança médiaCódigo para a parte 2. Visualização do código no RTL viewer

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Embora tanto a parte 1 como a parte 2 implementem contadores, a parte 2 conta com uma lógica que realiza adição de suas entradas.

A lógica do código é bem simples. O vetor counter armazena um número na base binária e 4 vetores binários, q1, q2, q3 e q4, armazenam, cada um, a sequência de bits necessária para representar um dígito, na base hexadecimal, de counter pelos displays de 7 segmentos. Essa sequência é determinada pela função "int\_to\_7seg”, que recebe como parâmetro um inteiro, sequência de 4 bits de counter convertida para a base 10, e retorna a sequência. Finalmente, cada um dos 4 vetores binários é mapeado por meio do pin assignment para um display de 7 segmentos, para representar um dígito.

Parte 3: desenvolvemos um circuito que exibe dígitos de 0 a 9 no display de 7 segmentos, alternando a cada 1 segundo. Um contador é responsável por controlar os intervalos de tempos.

Texto

Descrição gerada automaticamente com confiança médiaCódigo para a parte 3

Este código reutiliza a função “int\_to\_7seg” e seu contador é semelhante ao programa anterior. Definimos a constante COUNT\_MAX, que corresponde à frequência do clock, ou seja, quando counter alcançar o antecessor desse número (pois counter começa do número zero), 1 segundo terá se passado, o que implicará no incremento do sinal “clk\_1s”. O sinal “digit\_counter” é um contador cíclico que é incrementado a cada ciclo de “clk\_1s”, iterando entre os números 0 e 9. Finalmente, o vetor seg armazena a sequência de bits, gerada pela função “int\_to\_7seg”, que representa o dígito atual de “digit\_counter” e é mapeado para um display de 7 segmentos da FPGA.

Parte 4: implementamos um código VHDL que rotaciona a palavra "dE0 " nos displays HEX3-0 no sentido da direita para a esquerda com intervalo de 1 segundo entre cada deslocamento da palavra.

Texto

Descrição gerada automaticamenteCódigo para a parte 4

Nesse programa, a função “int\_to\_7seg” foi modificada, de modo que aceita como parâmetros apenas os números 0, 1 e 2, cada 1 retornando a sequência para representação em um display de 7 segmentos dos caracteres “d”, “e”, e “0” e “ “ respectivamente, mas preserva a lógica do contador de 1 segundo. A cada segundo, o sinal “digit\_counter” é incrementado e 4 vetores binários que armazenam a sequência necessária para acender os LEDs de um display, seg0, seg1, seg2 e seg3, recebem o valor de retorno da função “int\_to\_7seg”, cujo parâmetro é um número de 0 a 3, graças a operação de módulo aplicada em “digit\_counter”. Cada vetor recebe um valor diferente da função, já que cada parâmetro é incrementado por um número diferente, fazendo com que cada um se diferencie do outro. Essa variação permite que, a cada segundo, um dos 4 displays de 7 segmentos receba um caractere diferente de forma cíclica, sendo que o vetor cuja função receber como argumento o número 3 receberá uma sequência de bits que torna seu display apagado. Caso a chave corresponde ao reset seja acionada, os caracteres param de oscilar, transmitindo, apenas, a sequência “d” “e” “0” e “ ”.

Parte 5: expandimos o circuito da parte 4 para que a palavra percorra todos os displays de 7 segmentos da FPGA.

Texto

Descrição gerada automaticamenteCódigo parte 5

Nesse programa, os caracteres “d”, “e”, e “0” e “ “ precisam se deslocar para a direita a cada segundo passando por todos os 6 displays de 7 segmentos, o que exige a criação de 6 vetores pra armazenar as sequências de bits retornadas por “int\_to\_7seg”. Novamente, por causa do caráter cíclico da aritmética modular, o parâmetro “digit\_counter”, responsável pela iteração entre os caracteres, é incrementado por um número diferente e é convertido para um inteiro no módulo 6. Assim, a cada segundo, um vetor assume a sequência de um caractere diferente, e aqueles cujas funções receberem como argumentos números diferentes de 0 a 2 receberão uma sequência que torna seu display apagado.