

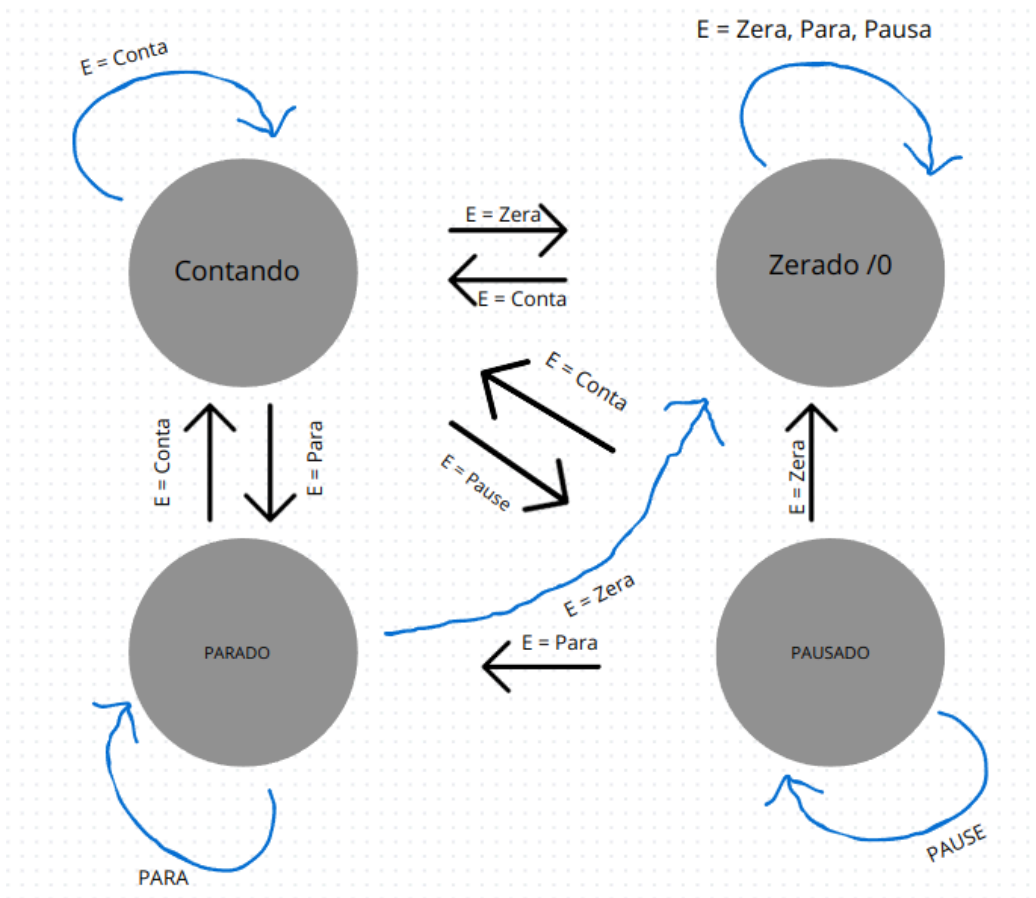
Universidade Federal de Pernambuco
Engenharia da Computação

Luan Silva, Erick da Silva, Jairo Neto,
Kauanny Barros, Nelson Junior

RELATÓRIO DO CRONÔMETRO

Recife
2024

Relatório - Código



Estado Atual	Entrada			
	reset	conta	pause	para
-				
Contando	Zerado	Contando	Pausado	Parado
Pausado	Zerado	Contando	Pausado	Parado
Parado	Zerado	Contando	Parado	Parado
Zerado	Zerado	Contando	Zerado	Zerado
Próximo Estado				

Os códigos apresentados implementam um cronômetro digital utilizando Verilog. O cronômetro possui entradas para controle (clock, reset, conta, para e pausa) e saídas para exibição dos dígitos em displays de 7 segmentos. A implementação é dividida em dois módulos: **contador** e **cronometro**.

O módulo **contador** é responsável por contar os pulsos do clock e atualizar os displays de acordo com os controles fornecidos pelos botões. As variáveis ``decimo``, ``SegundoUni``, ``SegundoDez`` e ``SegundoCen`` armazenam os valores dos décimos de segundo, unidades, dezenas e centenas de segundos, respectivamente. Um parâmetro ``parametro`` é usado para definir a quantidade de pulsos necessários para contar um décimo de segundo.

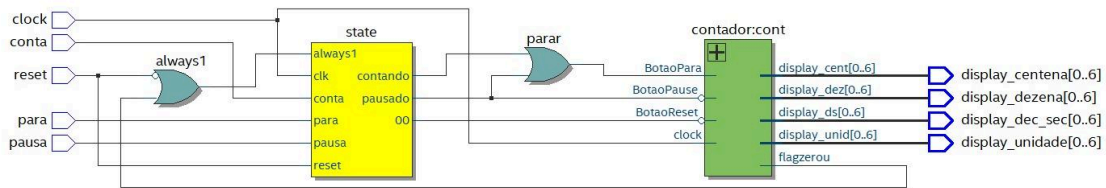
O contador possui lógica para resetar, pausar e incrementar os valores com base nos botões `BotaoReset`, `BotaoPause` e `BotaoPara`. A função `display` converte os valores binários para o formato necessário para controlar os displays de 7 segmentos. Há também uma lógica para detectar quando o cronômetro atinge o valor máximo (999.9 segundos), sinalizando isso com a flag `flagzerou`.

O módulo **cronometro** controla o estado do cronômetro, alternando entre quatro estados: `contando`, `zerado`, `pausado` e `parado`. Baseado nos inputs (reset, conta, para e pausa), ele define o comportamento do cronômetro em cada estado. As flags `zerar`, `parar` e `pausar` são usadas para determinar as ações apropriadas em cada estado.

A instância do módulo **contador** dentro do módulo **cronometro** conecta os sinais necessários para operar o cronômetro corretamente. A lógica combinacional no módulo **cronometro** define as ações para cada estado, enquanto a lógica sequencial é responsável pela transição entre estados baseados nas condições fornecidas pelos inputs e a flag `flagz` (indicando que o cronômetro atingiu o valor máximo).

Essa implementação permite que o cronômetro seja iniciado, pausado, parado e resetado, com atualização adequada dos displays para refletir o estado atual do cronômetro.

Relatório - Diagrama de blocos



O diagrama mostra a interação entre os módulos do cronômetro digital descrito nos códigos em Verilog. Ele detalha as conexões e os fluxos de sinal entre o módulo principal **cronometro** e o módulo **contador**.

Módulo Principal (cronometro)

Entradas e Sinais de Controle:

- clock: O sinal de clock principal que sincroniza as operações do cronômetro.
- conta, reset, para, pausa: Sinais de controle que definem o comportamento do cronômetro, permitindo iniciar, parar, pausar e resetar a contagem.

Máquina de Estados:

- O bloco “state” representa a máquina de estados, que controla os estados do cronômetro (contando, zerado, pausado, parado).
- O bloco “always1” é a lógica combinacional que define as ações a serem tomadas em cada estado, baseado nos sinais de controle.

Sinais Gerados pelo Estado:

- clk, conta, pausa, reset: Controlam o comportamento do módulo contador.
- parar: Um sinal de controle derivado do estado atual, usado para gerenciar o comportamento do cronômetro.

Módulo Contador (contador)

Entradas:

- clock: Sinal de clock.
- BotaoPara, BotaoPause, BotaoReset: Sinais de controle vindos da máquina de estados.

Saídas:

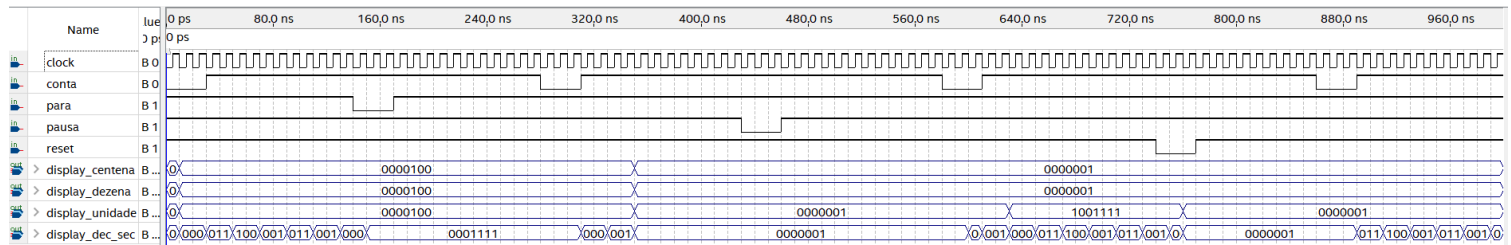
- display_cent, display_dez, display_ds, display_unid: Saídas que representam os valores a serem exibidos nos displays de 7 segmentos.
- flagzerou: Sinal que indica quando o cronômetro atingiu o valor máximo (999,9 segundos).

Conexões Internas

- O diagrama mostra as conexões de entradas e saídas entre os módulos:
 - Os sinais de controle clk, conta, pausa, reset e parar são derivados do bloco state e conectados às entradas do módulo **contador**.
 - As saídas do módulo contador (display_cent, display_dez, display_ds, display_unid, flagzerou) são usadas para exibir os valores contados e sinalizar eventos específicos, como quando o cronômetro atinge seu limite máximo.

Relatório - WaveForm

Cronômetro:



Na imagem acima tem-se o waveform do cronômetro com as entradas: clock, conta, para, pausa, reset; e saídas: display_centena, display_dezena, display_unidade, display_dec_sec.

O cronômetro foi inicializado em 999,0 segundos e seu parâmetro de 5.000.000 foi ajustado para 2, para melhor visualização do waveform.

Analisaremos, portanto, o comportamento do cronômetro a cada pulso.

Análise por pulsos:

-Pulso 1 (Conta = 0):

Quando conta = 0, o cronômetro inicializa a contagem até 999,7 segundos, momento em que recebe o pulso 2.

-Pulso 2 (Para = 0):

Quando para = 0, o cronômetro mantém o display em 999,7 segundos e interrompe a contagem até o próximo pulso.

-Pulso 3 (Conta = 0):

Quando conta = 0, o cronômetro retoma a contagem até 999,9 segundos, momento em que zera a contagem, atualizando o display para 000,0 segundos finalizando a contagem.

-Pulso 4 (Pausa = 0):

Quando pausa = 0, o cronômetro inicializa a contagem novamente mantendo o display onde parou (000,0 s).

-Pulso 5 (Conta = 0):

O display volta a ser atualizado retomando a contagem a partir de 000,8s.

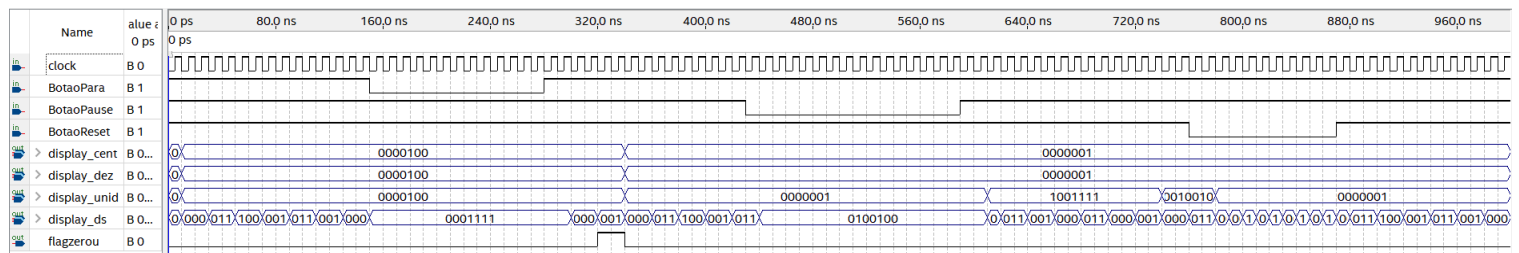
-Pulso 6 (Reset = 0):

O display é novamente zerado juntamente com a contagem mantendo 000,0s até o próximo pulso.

-Pulso 7(Conta = 0):

Inicia a contagem novamente.

Contador:



Na imagem acima tem-se o waveform do contador com as entradas: clock, BotaoPara, BotaoPause, BotaoReset; e saídas: display_cent, display_dez, display_unid, display_ds, flagzerou.

O waveform do contador foi gerado com o cronômetro inicializado em 999,0 segundos e seu parâmetro de 5.000.000 ajustado para 2, para melhor visualização do waveform.

Analisaremos, portanto, o comportamento do contador a cada pulso.

Análise por pulsos:

-Início (BotaoPara = 1, BotaoPause = 1, BotaoReset = 1):

Sempre que as entradas estiverem configuradas com 1 o contador iniciará a contagem e atualizará o display.

-Pulso 1 (BotaoPara = 0, BotaoPause = 1, BotaoReset = 1):

Quando BotaoPara = 0, o contador mantém o display em 999,7s e interrompe a contagem. Retomando assim que acaba o pulso, momento em que BotaoPara = 1.

-Saída (flagzerou = 1):

Antes do pulso 2 o contador atinge o número 999,9s, fazendo com que flagzerou = 1, zerando o display.

obs: a contagem permanece pois BotaoPara = 1, BotaoPause = 1, BotaoReset = 1 no exemplo do waveform, mas o contador garantirá que BotaoReset = 0 quando flagzerou = 1, durante a execução do código.

-Pulso 2 (BotaoPara = 1, BotaoPause = 0, BotaoReset = 1):

Quando BotaoPause = 0, o contador mantém o display em 000,5s, mas mantém a contagem internamente.

-Pulso 3 (BotaoPara = 1, BotaoPause = 1, BotaoReset = 0):

Quando BotaoReset = 0, o contador zera o display e interrompe a contagem.