

PCS3115 - Sistemas Digitais I - Trabalho 5

por Bruno de Carvalho Albertini

28/10/2020

Neste trabalho você desenvolverá a sua primeira máquina de estados finita em VHDL.

Introdução

Um vírus se espalhou pelo mundo e rapidamente os seres humanos estão se tornando zumbis. Mas os pesquisadores conseguiram descobrir um método para reverter o processo e tornar os zumbis em humanos novamente. Os pesquisadores descobriram que os zumbis emitem radiações de cor azul, verde e laranja, em sequências aleatórias. Se o zumbi for atingido por um raio Alpha após uma sequência específica de radiações, ele se tornará humano novamente. O dispositivo para tornar os zumbis em humanos possui um sensor de radiação, cuja saída X é uma palavra binária que representa a cor da radiação. Também terá um circuito para permitir o disparo do raio Alpha. Sendo um aluno brilhante de Sistemas Digitais, você foi solicitado a projetar o circuito de controle do disparo, que essencialmente é uma Máquina de Estados Finita, com entrada X e cuja saída binária Z habilitará o disparo (quando $Z=1$) ou não (quando $Z=0$). O método criado pelos pesquisadores prevê as seguintes sequências para identificar o momento certo para o disparo:

1. Pode-se fazer o disparo se o zumbi emitir uma sequência com uma radiação azul, seguida de uma quantidade ímpar, com pelo menos uma, de radiação verde;
2. Pode-se fazer o disparo se o zumbi emitir uma sequência com uma radiação laranja, seguida de uma quantidade qualquer (incluindo zero) de radiação verde.

O circuito de controle do disparo é importante, pois se o disparo do raio Alpha for feito em momento diferente de um desses dois padrões de sequência, não será mais possível tornar o zumbi em humano e ele estará condenado.

Qualquer relação com a pandemia atual é mera coincidência.

Este enunciado faz parte da lista de exercícios propostas para este módulo e isto é proposital pois o prazo de entrega é reduzido. Este enunciado foi adaptado de uma prova de 2019.

Considere que você deve atirar imediatamente se o zumbi atender qualquer um dos padrões de sequências, e continuar atirando enquanto obedecê-lo, parando somente quando a sequência for quebrada.

Atividades

Para este trabalho, você pode fazer um *testbench* para testar as duas atividades, e também um único arquivo com a sua solução, contendo a entidade e as duas arquiteturas, conforme visto na *live* de VHDL do dia 28/out.

T5A1 Desenvolva a máquina de estados finita que resolva o problema proposto, considerando que a codificação de cores para a entrada X de dois bits é: $X_1X_0 = 00$: nenhuma cor detectada, $X_1X_0 = 01$: cor azul detectada, $X_1X_0 = 10$: cor laranja detectada, e $X_1X_0 = 11$: cor verde detectada. Sua descrição deve ser estrutural e **não** pode conter `process` ou `function`.

```
entity zumbi is
  port(
    clock, reset: in bit;
    x: in bit_vector(1 downto 0);
    z: out bit
  );
end entity;
architecture estrutural of zumbi is
  component ffd is
    port (
      clock, clear, set: in bit;
      d: in bit;
      q, q_n: out bit
    );
  end component;
begin
end architecture;
```

T5A2 Refaça o seu projeto, mas dessa vez chame a arquitetura de fsm. Para esta atividade, você **não** terá o ffd disponível, e deve fazer usando o estilo visto na *live* com um processo. Não é permitido usar `function`, mas obviamente é permitido usar `process`. Se quiser manter a arquitetura anterior, não há problemas pois o juiz só analisará a arquitetura fsm.

Instruções para Entrega

Para este trabalho está permitida apenas a biblioteca `ieee.numeric_bit` do pacote `ieee`. Na atividade A1 você não pode usar o `process` e na atividade A2 você deve utilizá-lo. As funções estão proibidas em ambas atividades. A violação destas restrições acarreta nota zero automaticamente, sem direito a revisão.

Note que a Atividade 1 não permite `process`, nem mesmo comentado no arquivo, então lembre-se de retirar a arquitetura da Atividade 2 antes de enviar para a Atividade 1 caso opte por essa abordagem.

Trabalho 5, Atividade 1, 5 envios, maior nota, 5 pontos

Este site (clique) pode te ajudar na minimização das expressões.

Figura 1: Listagem: Entidade VHDL para T5A1. Não mude nada desta entidade, apenas complete com o seu projeto. A descrição do ffd está no final deste arquivo caso queira usar nos seus testes. A descrição deste componente será carregado automaticamente pelo juiz, então basta declarar as instâncias que desejar (você não deve enviar a descrição do ffd com sua solução). A entidade para o T5A2 é idêntica, mas o nome da arquitetura muda (veja enunciado).

Trabalho 5, Atividade 2, 5 envios, maior nota, 5 pontos

Restrições, preste atenção!

Para cada atividade deste trabalho, há um *link* específico no e-Disciplinas. Acesse-o somente quando estiver confortável para enviar sua solução. Em cada atividade, você pode enviar apenas um único arquivo com sua descrição VHDL em UTF-8. O nome do arquivo não importa, mas sim a descrição que está dentro. As entidades devem ser como as especificadas ou o juiz te atribuirá nota zero.

Quando acessar o *link* no e-Disciplinas, o navegador abrirá uma janela para envio do arquivo. Selecione-o e envie para o juiz. Jamais recarregue a página de submissão pois seu navegador pode enviar o arquivo novamente, o que vai ser considerado pelo juiz como um novo envio e pode prejudicar sua nota final. Caso desista do envio, simplesmente feche a janela antes do envio.

Depois do envio, a página carregará automaticamente o resultado do juiz, quando você poderá fechar a janela. Se não quiser esperar o resultado, feche a janela após o envio e verifique sua nota no e-Disciplinas posteriormente. A nota dada pelo juiz é somente para a submissão que acabou de fazer. Sua nota na atividade poderá ser vista no e-Disciplinas e pode diferir da nota dada pelo juiz dependendo da estratégia de atribuição de notas utilizada pelo professor que montou o problema.

Atenção: não atualize a página de envio e não envie a partir de conexões instáveis (e.g. móveis) para evitar que seu arquivo chegue corrompido no juiz.



Pode demorar alguns segundos até o juiz processar seu arquivo.

```
entity ffd is
  port (
    clock, clear, set: in bit;
    d: in bit;
    q, q_n: out bit
  );
end entity;

architecture arch of ffd is
begin
  meuffd: process(clock, clear, set)
    variable tmp: bit;
  begin
    if clear='1' then tmp:='0';
    elsif set='1' then tmp:='1';
    elsif clock'event and clock='1' then tmp := d;
    end if;
    q<=tmp; q_n<=not(tmp);
  end process;
end architecture;
```

Figura 2: Listagem: O ffd que está disponível no juiz. Não envie este arquivo, a listagem está aqui para você usar nos seus próprios testes.