UNIVERSIDADE FEDERAL DO ABC - UFABC
CIÊNCIA DA COMPUTAÇÃO
SISTEMAS DIGITAIS

PROJETO: RECONHECEDOR DE SEQUÊNCIA 1001 EM VHDL

Luiz Felipe Leal Gomes Azzolini
Vinicius Medeiros França

Santo André, Agosto de 2022

Alunos:

Luiz Felipe Leal Gomes Azzolini - RA 11048715
Vinicius Medeiros França - RA11110716

Professor:

Prof. Jose Artur Quilici Gonzalez

SUMÁRIO

INTRODUÇÃO

A linguagem HDL foi criada na década de 1980 pela DARPA e padronizada posteriormente pelo IEEE, com o objetivo de ser uma linguagem de descrição de hardware, ou seja utilizada para simular ou implementar o funcionamento de um hardware.

Utilizado tanto para programação de FPGAs quanto para concepção de ASIC, como alguns processadores da Intel, por exemplo. É possível compilar código em uma netlist, que é uma estrutura de dados que descreve o comportamento de um circuito com as conexões e componentes combinacionais de um hardware.

Com isso também é possível fazer testes e simulações de lógica combinacional e circuitos sem necessidade de projetá-los com componentes reais, de forma totalmente virtual.

OBJETIVO

O projeto tem por objetivo implementar e um circuito reconhecedor de sequência 1001, utilizando dois softwares diferentes, o GtkWave e o Quartus II com o ModelSim, de modo a obter conhecimentos acerca de implementação de código VHDL, assim como testes e análise dos dados gerados pelo código e utilização de softwares profissionais e simplificados para sistemas embarcados.

JUSTIFICATIVA

O trabalho contribui com conhecimentos básicos acerca de desenvolvimento de código VHDL utilizados em diversos processadores e sistemas embarcados, necessários para um aluno de Ciência da Computação.

METODOLOGIA

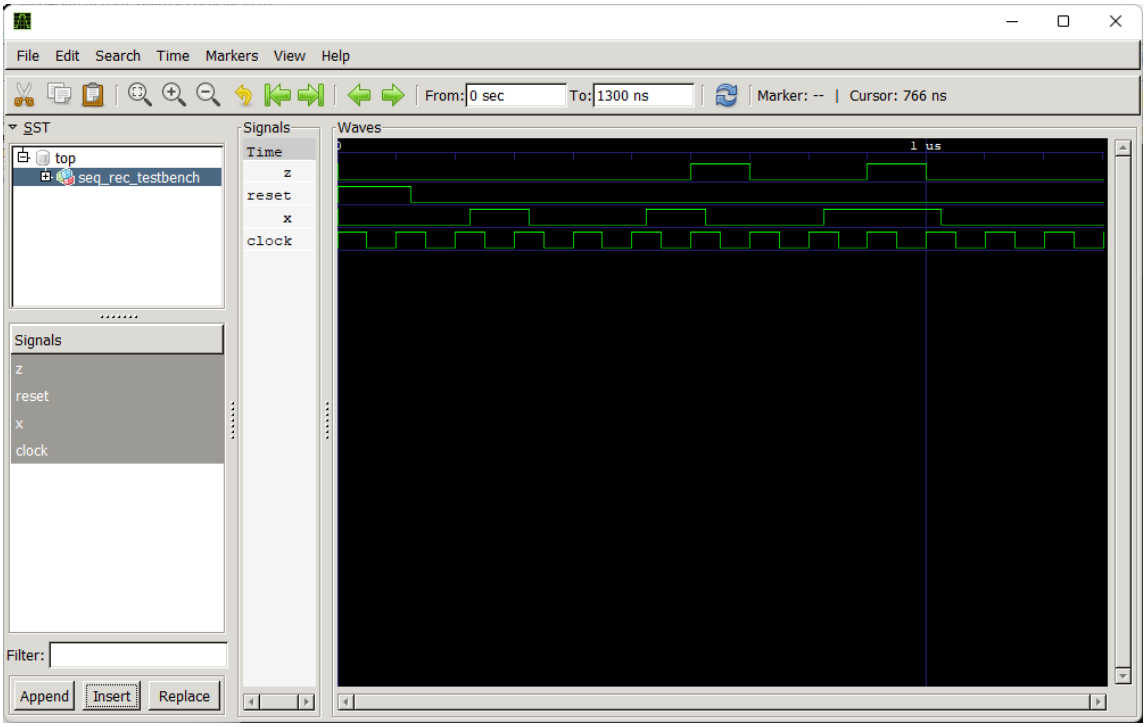
A metodologia do projeto se baseou em rodar testes de mesa e comparar com o resultado da saída, tanto para entradas em que a sequência ocorre mais de uma vez, como apenas uma vez, nenhuma vez ou duas vezes de forma sobreposta (1001001 por exemplo).

O programa foi executado diversas vezes a fim de contemplar vários casos com sequências diferentes de 11 bits em cada um dos *testbenches* de modo a cobrir vários casos e verificando se havia o resultado esperado na saída.

Para a exibição dos testes foram utilizadas as ferramentas gtkwave e Quartus II, juntamente ao ModelSim. Não foi identificada a razão pela qual o código testbench gerado pelo Quartus II não rodou no gtkwave, então foi desenvolvido a parte um código novo para a ferramenta em questão que apresentou o mesmo comportamento observado no ModelSim

ANÁLISE DOS RESULTADOS

Colocando a sequência 01001001100 podemos observar que foram identificados duas vezes a sequência retornando em dois momentos 1 na saída Z, sendo assim, a forma como foi feito o código identifica sequências sobrepostas.



Visualização da execução de código vhdl no gtkwave

Note que a saída Z só retorna 1 quando o clock está na primeira borda de subida após a sequência ser reconhecida na entrada X, como era o comportamento esperado.

EXEMPLO DE FUNCIONAMENTO

Para o gtkwave a sequência de entrada pode ser alterada no arquivo *sec_req_testbench.vhd* na *linha 13* após isso é necessário executar comandos com o terminal, ou prompt de comando, mas no projeto foi deixado um arquivo .bat para padronizar esses comandos já que são os mesmos sempre.

São eles:

```
ghdl -a seq_rec.vhd
ghdl -a seq_rec_testbench.vhd
ghdl -e seq_rec_testbench
ghdl -r seq_rec_testbench --wave=result.ghw --stop-time=1300ns
```

Após isso basta abrir o gtkwave, no menu suspenso, clicar em file >> open new tab e selecionar o arquivo result.ghw gerado pelos comandos anteriormente, na mesma pasta que os códigos vhdl, assim a tela exibirá a execução do código e para a entrada teste definida no testbench.

Para o Quartus II + ModelSim é necessário usar o Quartus II para abrir o arquivo req_sec_1001.qpf como projeto. Clicando em Tools >> Run simulation tool >> RTL Simulation, o ModelSim abrirá, selecione nessa janela a library “work”, no menu suspenso clique em Compile >> Compile…, selecione o arquivo simulation\modelsim\seq_rec_1001.vht e clique por fim no botão “Compile” e após em “Done” ao terminar as mensagens de compilação na aba de transcript.

Clique library work.seq_rec_1001_vhd_tst duas vezes, uma nova janela irá se abrir. Após, com o botão direito, selecione a linha seq_rec_1001_vhd_tst e vá em Add to >> Wave >> All items in region, procure pelo botão Run All e rode a simulação, espere alguns segundos e aperte o botão Stop.

Após dar um “zoom full” aproxime o zoom até conseguir ver nitidamente as oscilações do CLK e corra com a barra de scroll da janela wave - default completamente para esquerda e o sinal poderá ser visualizado e analisado

CONCLUSÃO

Tanto no gtkwave quanto no ModelSim os resultados foram apresentados como o esperado, com a saída Z retornando 1 na próxima borda de subida do clock a partir do momento em que a sequência é inserida na entrada X.

O software Quartus II facilita a implementação de parte do código do testbench, de modo que o usuário precise se preocupar apenas com a definição da entrada e como os valores são passados para ela em projetos mais simples como este

REFERÊNCIAS

- **Finite state machines.** FPGA designs with VHDL. Disponível em: <https://vhdlguide.readthedocs.io/en/latest/vhdl/fsm.html>. Acesso em 08 de Ago. de 2022.
- **VHDL Básico: Parte 1 – Entidade.** Wmbarcados. Disponível em: <https://embarcados.com.br/vhdl-basico-parte-1-entidade/>. Acesso em 12 de Ago. de 2022.