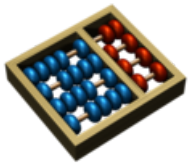
Universidade Estadual de Campinas
Instituto de Computação
MO601 – Arquitetura de Computadores II
Prof. Rodolfo Jardim de Azevedo



Projeto 2

Um simulador simples do processador RISC-V

Rubens de Castro Pereira
RA: 217146

Campinas - SP
26 de abril de 2023

Índice

1 Introdução

Este relatório tem o propósito de apresentar o resultado do projeto 1 da disciplina Arquitetura de Computadores II (MO601) ministrada pelo Prof. Rodoldo Jardim de Azevedo, cujo objetivo é a implementação e testes de um simulador super básico de circuitos lógicos. A seção 2 apresenta a descrição geral do projeto, a seção 3 relaciona as tecnologias utilizadas no ambiente de desenvolvimento, a seção 4 apresenta o algoritmo de simulação de circuitos lógicos, a seção 5 detalha a forma como os testes foram realizados e a seção 6 apresenta as considerações sobre o aprendizado nesse projeto.

2 Descrição geral

O objetivo deste projeto é implementar um simulador básico de circuitos lógicos considerando portas lógicas, estímulos de entrada e tipo de atraso em cada simulação.

As portas lógicas possíveis nesse simulador são AND, OR, NOT, NAND, NOR e XOR. Os estímulos podem ser variáveis ou indicadores de tempo. As variáveis indicam os valores dos sinais de entrada (0 ou 1) durante a simulação do circuito lógico, sendo permitida até 24 variáveis distintas identificadas pelas letras do alfabeto do idioma português. Os indicadores de tempo são identificados com o caractere '+' seguido de um número inteiro qualquer *n*, significando que deve-se avançar *n* tempos de simulação.

O comportamento do simulador considera dois tipos de atraso: a) atraso 0 (*delay 0*) indicando que as portas lógicas operam no mesmo tempo da simulação em curso; ou b) atraso 1 (*delay 1*) onde as portas lógicas operarão com atraso de **um** tempo de simulação e, portanto, os sinais de entrada nas portas lógicas serão aqueles sinais do tempo imediatamente anterior ao tempo atual da simulação.

Para cada unidade de tempo da simulação, deve-se emitir uma saída com uma lista contendo o valor do tempo da simulação e os valores de cada variável.

Para o encerramento de uma simulação específica, será considerada como condição de parada a produção de duas linhas de saída exatamente iguais, indicando que o circuito não possui estímulos a processar e que todas as portas lógicas também processaram seus sinais levando em conta o tipo de atraso do circuito. O encerramento da simulação significa que o circuito estabilizou e que não haverá nenhuma nova saída diferente das anteriores.

Para testar o simulador, vários cenários de teste são fornecidos contendo diferentes circuitos lógicos e estímulos. Para cada cenário de teste são executadas as principais etapas abaixo:

1. Criação do circuito com suas respectivas portas lógicas.
2. Processamento de todos os estímulos com atraso 0 (sem atraso).
3. Geração da saída da simulação com atraso 0.
4. Processamento de todos os estímulos com atraso 1.
5. Geração da saída da simulação com atraso 1.

3 Ambiente de Desenvolvimento

O ambiente de desenvolvimento do simulador está organizado com as seguintes tecnologias:

- Sistema operacional: Windows 10
- Ambiente integrado de Desenvolvimento (IDE): Visual Studio Code (VS Code) version 1.76.2
- Linguagem de programação: Python versão 3.9.5
- Pacotes Python: OS e Pandas
- Embecosm - Tool Chain Downloads
  - https://www.embecosm.com/resources/tool-chain-downloads/
  - Pacote selecionado para instalação do compilador RISC-V 32 bits: Windows 10 - 32-bit GCC (.zip) - https://buildbot.embecosm.com/job/riscv32-gcc-win64/169/artifact/riscv32-embecosm-gcc-win64-20230402.zip
  - riscv32-embecosm-gcc-win64

4 Algoritmo de Simulação

O algoritmo do simulador do processador RISC-V RV32IM é apresentado em *algorithm 1*.

5 Testes

Os testes do projeto foram organizados em pastas de testes específicos localizados em */test* no projeto *RCP-MO601-Project-01*. Cada pasta de teste contém a especificação do circuito lógico desejado (*circuito.hdl*), os estímulos (*estimulos.txt*) a serem processados ao longo do tempo (valores de variáveis e/ou indicadores de tempo), os arquivos com as saídas esperadas (*esperado0.csv e esperado1.csv*) e os arquivos resultantes da simulação (*saida0.csv e saida1.csv*).

Novos testes podem ser adicionados à pasta principal de testes (*/test*), contudo o simulador exige a existência dos arquivos *circuito.hdl* e *estimulos.txt*. Além disso, não existe limite para a quantidade de testes incluídos, sendo que o simulador processará todos aqueles que existirem na pasta principal de testes.

Durante a simulação dos testes, conforme pode-se observar no *algorithm 1*, todas as pastas de testes específicos são consideradas, seus arquivos de entrada são processados e os resultados (saídas) são armazenados nas mesmas.

6 Considerações sobre o aprendizado nesse projeto

O assunto tratado nesse projeto é muito diferente dos temas que venho trabalhando nas últimas três décadas de desenvolvimento de sistemas de informação. Contudo, desafios são excelentes oportunidades de aprendizado, fixação de novos conceitos e exploração de tecnologias e ferramentas.

Neste projeto foi possível apreender conhecimentos principalmente relacionados à área de arquitetura de computadores com destaque ao conceito de atraso em circuitos lógicos e funcionamento das portas lógicas, elementos muitos simples, mas combinadas entre si podem produzir resultados interessantes e importantes. A compreensão do conceito de atraso no circuito lógico aplicado às portas lógicas ao mesmo tempo que os sinais de entrada das variáveis são inseridos, representou o ponto mais relevante deste trabalho. Embora houveram dificuldades no início, mas após estudos e diálogos com o professor, foi possível compreender a ideia do atraso e, sabe-se que quando entendemos a definição de um problema, boa parte de sua resolução está encaminhada.

Outro ponto de destaque é o uso de novas ferramentas no desenovlvimento e entrega do trabalho combinando os produtos de GitHub e Docker, atualmente ferramentas de ampla utilização não somente no meio acadêmico mas no âmbito organizacional e empresarial.

Ressalto dois pontos importantes para o sucesso do trabalho: 1) a especificação do projeto está muito bem detalhada permitindo o pleno entendimento do trabalho a ser desenvolvido e das entregas esperadas; e 2) elevada disponibilidade do professor no atendimento aos alunos.

Por fim, destaco que os resultados alcançados nesse projeto são fruto da combinação de uma completa especificação do projeto, boa disponibilidade do professor em esclarecer e colaborar com o estudante, elevada dedicação no desenvolvimento do projeto conforme especificado e foco na entrega do produto dentro do prazo esperado.

Algorithm 1: Algoritmo do simulador do processador RISC-V RV32IM.

Entrada: *programa compilado/linked*

Saída: *arquivo de log de saída programa compilado exemplo: 000.main.log*

```
1  SimularExecucaoPrograma(programa compilado) begin
2      inicializar circuito atribuindo valor 0 a todas as variáveis
3      inicializar indicador de tempo da simulação
4      while circuito não estabilizar do
5          foreach estimulo do
6              if variavel then
7                  atribuir valor lido à respectiva variável
8                  retornar para ler novo estímulo
9              else
10                 Processar conforme a quantidade de indicador de tempo
11                 while circuito não estabilizar do
12                     calcular portas lógicas do circuito conforme o atraso (quando atraso
                       0, utilizar os valores do tempo atual nas entradas das portas lógicas
                       e quando atraso 1, utilizar os valores do tempo de simulação
                       anterior nas entradas das portas lógicas).
13                 end
14                 gerar saída utilizando o tempo atual da simulação e os valores de todas
                    as variáveis do circuito
15             end
16             incrementar indicador de tempo da simulação
17         end
18         /* Executa novamente o circuito lógico visando sua estabilização
            devido ao término dos estímulos de entrada.  */
19         if circuito não estabilizar then
20             calcular portas lógicas do circuito conforme o atraso
21             incrementar indicador de tempo da simulação
22         end
23     end
24 end

25 /* Programa principal que gerencia a simulação de programas no
    processador RISC-V RV32IM*/

26 foreach programa compilado do
27     ler programa compilado
28     SimularExecucaoPrograma(programa compilado)
29     gerar arquivo de log de saída do programa compilado
30 end
```