CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
CURSO DE ENGENHARIA DA COMPUTAÇÃO

PROPOSTA DE PROJETO PARA ESTÁGIO SUPERVISIONADO OBRIGATORIO

Curitiba
Novembro/2018

BRUNO DUARTE GOMES

PROPOSTA DE PROJETO PARA ESTÁGIO

Proposta de estágio apresentado ao Curso de Engenharia da Computação do Centro Universitário UNINTER.

Curitiba
Novembro/2018

Sumário

1 Objetivos

1.1 OBJETIVO GERAL

Executar um programa compilado em um processador virtual a ser criado.

1.2 OBJETIVOS ESPECIFICO

Criar uma biblioteca em Python que possibilite simular circuitos digitais; usando a biblioteca construida, construir um processador seguindo uma arquitetura conhecida; executar programas no processador a fim de testar sua funcionadidade

2 Revisão da Literatura / Fundamentação Teórica

A importância do processador em sistemas de computação é imensa, ele é responsável por todas as operações matemáticas executadas pelo computador; devido a essa importância fica claro que conhecer o funcionamento de um processador é algo vital para um engenheiro da computação. Uma das melhores maneiras para entender como funciona um processador é construindo um, e de fato, grande parte dos estudantes de engeharia da computação acabam fazendo isso em alguma ponto durante sua formação acadêmica. Esse ato é tão comum que D. Harris 2012 dizem que a construção de um processador é quase que um rito de passagem para qualquer engenheiro da computação.

Existem diferentes soluções disponíveis para se construir um processador no mercado. Uma delas é montando o processador físico em protoboards usando diversos chips de circuito integrado, muitas protoboards e muito fio. Uma alternativa um pouco mais barata e mais simples seria através de ferramentas que simulam circuitos digitais.

Existem diversas ferramentas para simular circuitos disponiveis, cada uma com suas particularidades, vantagens e desvantagens. A dificuldade com essas ferramentas vem do fato de elas serem uma solução comercial e não serem flexíveis, pois nem sempre é possível trabalhar em módulos; ou seja, criar um bloco de lógica digital, salvar esse bloco e usar usá-lo em um outro circuito, isso limita a abstração, ferramenta que é muito usada em circuitos digitais. Linguagens HDL são uma alternativa para o problema porém elas nem sempre são intuitivas e requerem que o usuario aprenda a línguagem antes de começar.

Uma solução para esse problema seria: uma ferramenta capaz de simular lógica digital que aplicasse os conceitos de modularidade, que ao mesmo tempo disponha de uma interface que seja relativamente intuitiva e que seja capaz de abstrair e simplificar o processo de criação de um circuito digital.

3 Metodologia / Recursos

Para criar uma ferramenta capaz de simular circuitos digitais devem ser criadas classes que abstraiam os elementos principais da mesma. Existem diversas maneiras equivalentes para realizar essa tarefa, o importante é que no final a ferramenta seja capaz de simular a lógica digital. Utilizarei a linguagem Python para criar classes e terei objetos que abstraiam clocks, blocos de circuito, fios, sinais e portas lógicas.

Apos o módulo inicial ter sido implementado e testado, podemos ter certeza que as classes funcionam de acordo com as regras da lógica digital. O proximo passo será criar blocos de lógica digital a partir dessas abstrações feitas. A construção de circuitos como multiplexadores, demultiplexadores, somadores, flip-flops e etc, seguirão a descrição desses circuitos feita no "Digital Logic and Computer Architecture"por D. Harris e S. Harris. Pode-se considerar um bloco de lógica digital como correto quando a tabela verdade gerada por ele no simulador é igual a tabela verdade exposta na literatura.

Tendo sido criado os blocos da lógica digital é, então, possivel combina-los e construir um processador. O processador a ser construído será o processador de ciclo unico descrito na literatura. Esse é um processador de 32 bits que segue a arquitetura MIPS e que possui instruções de soma, subtração, operações lógicas (AND, OR e XOR) e comparações lógicas. O processador será construído de acordo com a micro-arquitetura descrita no livro. Analogamente aos blocos de lógica digital, o processador pode ser considerado correto quando dada uma instrução esse circuito executa a instrução como ela é descrita na literatura.

Após as instruções individuais terem sido testadas e sido possível comprovar que as mesmas funcionam corretamente, será escrito um programa para o processador o execute. Esse programa pode ser escrito em assembly e compilado para código-máquina ou pode ser escrito diretamente em código-máquina. Para alimentar o processador com as intruções, existem diversas maneiras. Uma delas seria deixar as instrunções escritas no código fonte do processador. O objetivo é que o processador seja capaz de executar os programas compilados para ele, porém, respeitando as suas limitações em termos de memória.

4 Cronograma / Atividades

Semana	Horas	Atividades
1	20	Pesquisar sobre eletrônica digital e lógica digital; projetar classes para biblioteca;
2	20	Codificar classes; testar classes; projetar classes para portas lógicas
3	20	Testar portas lógicas; implementar elementos da lógica combinacional; testar lógica combinacional
4	20	Implementar elementos da lógica sequencial; testar lógica sequencial
5	20	Projetar abstração para bloco de memoria ROM; definir como será feito o armazenamento de memoria ROM
6	20	Construir processador seguindo a micro-arquitetura do livro
7	20	Testar instruções do processador; escrever programas em assembly
8	20	Gerar codigo-maquina; rodar programas compilados; debugar
9	20	Finalizar documentação do código; gerar diagramas UML

Bibliografia

D. Harris, S. Harris (2012). *Digital Design and Computer Architecture.*