

MO601 (Arquitetura de Computadores II)

Projeto 1 - Um simulador super básico de circuitos lógicos

Nesse projeto, cada aluno deve implementar um simulador básico de circuitos lógicos. A intenção é praticar conceitos de simulação e entender como eles podem ser implementados em código, além de ter uma visão da eficiência desses algoritmos.

1 Descrição geral do projeto

O objetivo geral do projeto, é a partir de um circuito descrito no formato “hdl”, simular seu comportamento e apresentar suas saídas para atraso 0 e atraso 1 em formato “CSV”, depois de receber estímulos, dentro de ciclos, até não restar nenhum novo e as entradas e saídas tiverem seus estados idênticos ao do ciclo anterior.

Cada linha da descrição é uma saída de um circuito lógico, representada por letras de “A” a “Z”. Os circuitos simulados foram “AND”, “OR”, “NOT”, “NAND”, “NOR” e “XOR”, onde cada um deles espera receber duas entradas (a não ser pela “NOT”, que só recebe uma), além de uma saída poder ser entrada de outro circuito.

Já os estímulos estão no formato “txt” e cada linha representa a mudança de uma entrada ou um conjunto das mesmas, que são separados por ciclos que são representados por “+” seguido pelo número de ciclos que passaram até os próximos estímulos.

2 Descrição do seu ambiente de desenvolvimento

Esse projeto foi desenvolvido no sistema operacional Ubuntu 18.04 LTS, como linguagem de programação foi usado o Python versão 3. Não foi usada nenhuma biblioteca externa, somente as nativas, esse ambiente é simulado usando um “dockerfile”.

3 Descrição do seu algoritmo de simulação

Primeiramente o algoritmo coleta todos os nomes dos diretórios dentro da pasta “test”. Em seguida, para cada diretório, ele configura as variáveis iniciais e define as funções que podem ser usadas pelo simulador.

O próximo passo é coletar as descrições dos circuitos dentro do arquivo “circuito.hdl” e dividi-los em linhas, um para cada saída. Em seguida, cada linha foi organizada adicionando vírgulas e parênteses, para que o Python conseguisse interpretá-las corretamente como chamada de uma função.

Para o atraso 0, foi usada a forma que cada saída de circuito dependia apenas dos estímulos de entrada, pois dessa forma independente da ordem de cálculo a saída sempre estaria o mais atualizada possível, pois um circuito não precisaria esperar para receber a saída de outro, evitando problemas de dependência. Os resultados foram salvos no arquivo “saída0.csv”.

Já para o atraso 1, foi usada a forma original já que os circuitos são calculados no ciclo atual usando os valores de entrada do último ciclo, então nesse caso a dependência entre eles é importante. Os resultados foram salvos no arquivo “saída1.csv”.

Com as descrições dos circuitos organizadas, o próximo passo foi organizar os estímulos de entrada, coletadas do arquivo “estimulos.txt”. Como haviam dois layouts diferentes, um que todos os estímulos de um ciclo eram comprimidos em uma linha e outro que era um estímulo por linha, independente do layout usado eles foram convertidos para o formato de um por linha.

O último passo foi para cada ciclo calcular o estado das entradas de estímulos e saídas dos circuitos e salvá-los nos arquivos CSV, até que a curva atual fosse igual a anterior e não houvessem mais estímulos para receber, tanto para o atraso 0 quanto 1.

4 Descrição de como você testou seu projeto

Para testar o projeto foi usado o circuito fornecido pelo professor além de outros que eu mesmo montei para os testes. Cada conjunto é formado pela descrição dos circuitos, os resultados esperados e os estímulos recebidos. Para saber se o simulador estava correto, tanto a saída para atraso 0 quanto atraso 1 foram manualmente comparadas com os resultados esperados e todos os exemplos testados funcionaram corretamente.

5 Considerações gerais sobre seu aprendizado nesse projeto

Eu nunca tinha trabalhado com simuladores ou manipulação de strings, achei bem interessante a experiência de ter que desenvolver uma forma própria de decodificar circuitos descritos em um formato diferente do padrão, pois sempre trabalhei com formatos de dados já estruturados onde não era necessária nenhuma organização do dado, normalmente os pacotes já entregavam os mesmos prontos e só era necessário utilizá-los.

Isso foi uma ótima preparação para próximos projetos que visem alguma forma de simulação que recebem como entradas strings, já que foram aprendidas diversas ferramentas para trabalhar com os mesmos, criando uma forma de estrutura nova e adequada para os propósitos do projeto.

Outro aprendizado foi o uso de dockerfiles, achei muito interessante como eles funcionam e a possibilidade de usá-los para testar algoritmos em ambientes já configurados, onde não há necessidade de instalar pacotes ou fazer alterações na máquina pessoal, só é preciso executar o docker e ele roda perfeitamente, contanto que esteja configurado corretamente.