

Trabalho Prático 2 - Lógica Sequencial

1 Contexto

As investigações acerca da língua *xylo* continuaram no campus. Com a equipe ainda liderada pela mesma professora, conseguiram realizar avanços na interpretação da linguagem. Agora já se sabe como classificar morfologicamente algumas classes de palavra: substantivos, adjetivos, advérbios e verbos.

Uma característica curiosa das classes de palavra do povo *xylo* é que elas possuem uma estrutura muito simples, lembrando que cada letra da língua é uma nota musical. Os substantivos têm tamanho indefinido, mas sempre terminam em dó, ré ou mi, no tom baixo. Os adjetivos, advérbios e verbos têm uma quantidade fixa de letras, mas apresentam mudanças no fim ou no começo da palavra para indicar suas transformações ou conjugações. Mais detalhes de como essas regras funcionam estão na próxima seção.

Dessa vez, a tarefa delegada aos alunos de ISL foi dividida em detectar cada um desses tipos de palavra e classificá-los. Assim, algumas equipes desenvolverão o módulo de detecção de substantivos, outras equipes desenvolverão o módulo de adjetivos e advérbios, e outras equipes desenvolverão o módulo dos verbos.

A história ainda é fictícia, mas, novamente, o desafio é real.

2 Descrição

Este trabalho consiste em elaborar um dos possíveis módulos de análise morfológica da linguagem fictícia *xylo*, apresentada no trabalho anterior da disciplina. Todos os módulos terão a mesma interface mostrada na Figura 1, e deverão ser implementados usando o conceito de **máquina de estados finita de Moore**.

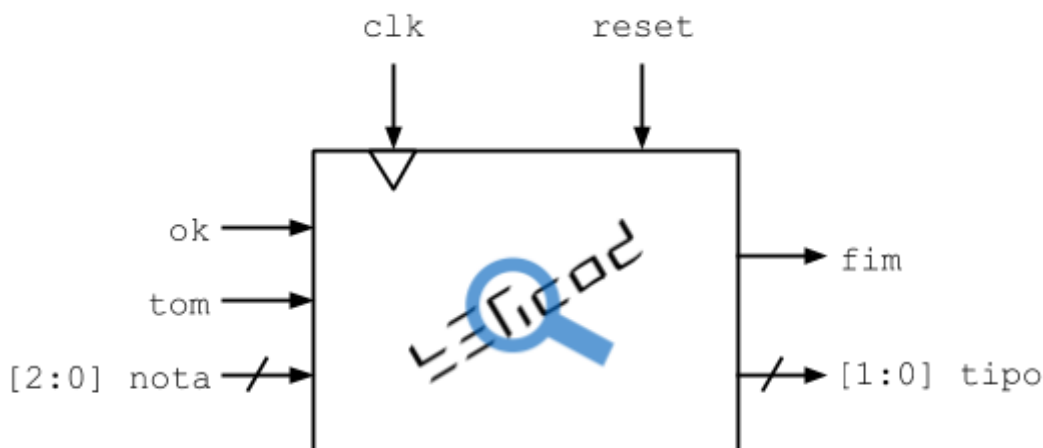


Figura 1 - Módulo genérico de análise morfológica *xylo*.

Esse módulo funciona, genericamente falando, da forma descrita a seguir. Será passada, letra a letra, uma palavra da língua *xylo*, e a máquina de estados vai detectar se a palavra, até um dado momento, se encaixa em alguma classificação morfológica. Quando for informado o fim da palavra - mais detalhes sobre isso logo a seguir -, a máquina então indica que chegou ao fim, e dá a classificação final da palavra.

O sinal `clk` corresponde ao *clock*. O sinal `reset` é responsável por colocar a máquina de estados de volta ao seu estado inicial. As entradas `tom` e `nota` servem para indicar qual é a próxima letra da língua *xylo* que será lida, seguindo os padrões definidos pelo seu grupo no trabalho anterior. A entrada `ok` serve para confirmar a entrada de uma nova letra, ou seja, apenas quando o sinal `ok` estiver ligado é que a máquina vai de fato ler a entrada de uma letra e realizar uma transição. O fim da palavra vai ser indicado entrando com uma letra inválida - lembrem-se que há mais de uma combinação de nota que gera uma letra inválida.

Quando o final da palavra for atingido, a saída `fim` deverá ser ligada, e em todos os estados anteriores, essa saída fica desligada. Em **todos** os estados a saída `tipo` deverá ser atualizada para corresponder à classificação morfológica à qual a palavra pertence até aquele dado momento.

Para todos os módulos, o valor `2'b00` para a saída `tipo` indica uma palavra inválida. Porém, a definição das classificações morfológicas é que mudará para cada grupo, que deverá **escolher** qual dos três módulos deseja fazer e **definir** um valor para cada possível saída. Segue a definição de como funciona cada módulo:

1. Módulo classificador de substantivos: este módulo vai detectar substantivos, classificando-os entre substantivos concretos, abstratos e de nome próprio.

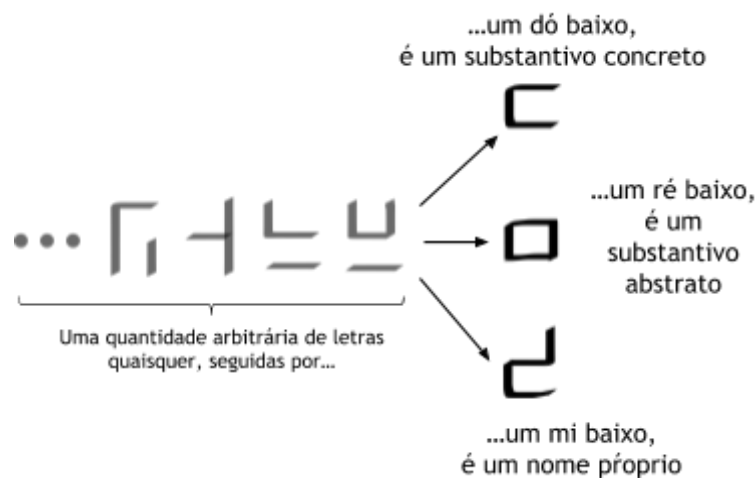


Figura 2 - Regra dos substantivos.

2. Módulo classificador de adjetivos e advérbios: este módulo vai detectar palavras que se classificam entre adjetivos, adjetivos de comparação e advérbios.

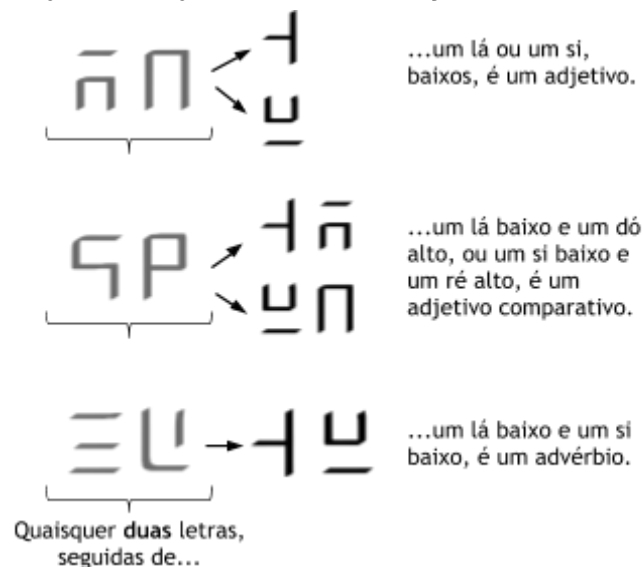


Figura 3 - Regra dos adjetivos e advérbios.

3. Módulo classificador de verbos: este módulo vai detectar verbos e classificar se ele está conjugado no infinitivo, no passado ou no futuro.

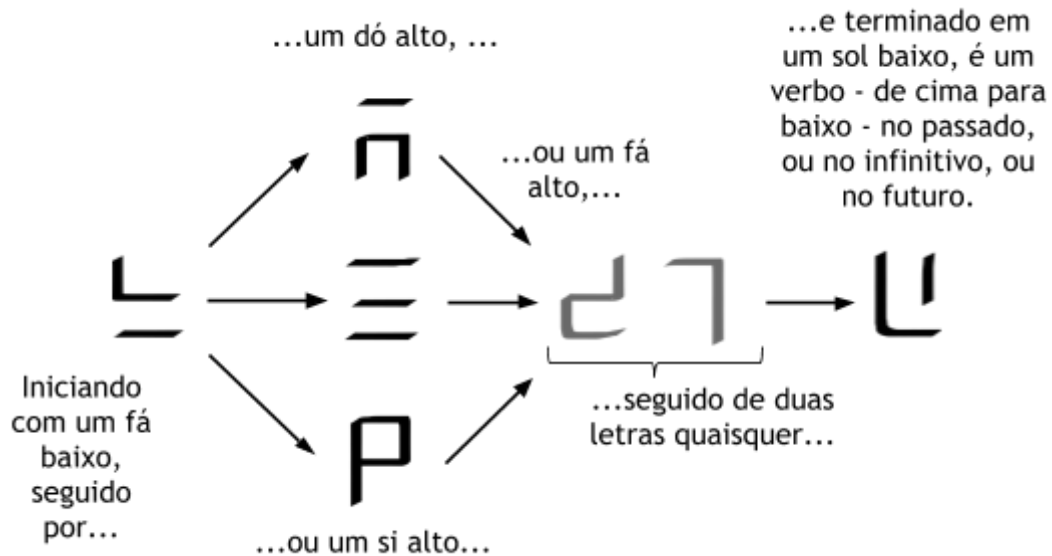


Figura 4 - Regra dos verbos.

Exemplo: seu grupo poderia escolher fazer o módulo de verbos, e definir que a saída tipo vai valer: 2'b00 para palavra inválida - padrão para todos os grupos -, 2'b01 para passado, 2'b10 para infinitivo e 2'b11 para futuro.

3 Elaboração

1. Escolher qual módulo será feito;
2. Criar um diagrama da máquina de estados finita usando o software [JFLAP](#) ou similar;
3. Implementar em Verilog a máquina de estados finita;
4. (Opcional) Implementar em Verilog um módulo de simulação para sua máquina;
5. (Opcional) Visualizar as formas de onda da simulação;
6. Implementar em Verilog um módulo para síntese no FPGA.

4 Observações

- Note que os passos 4 e 5 são opcionais, mas servem de grande ajuda para verificar se o módulo está funcionando corretamente antes de passar para o FPGA.
- Seu grupo pode reaproveitar o módulo criado no trabalho anterior para visualizar no display de sete segmentos qual letra está sendo inserida na máquina.
- A interface genérica apresentada na Figura 1 apresenta um mínimo a ser feito. Apenas com aquelas portas é possível realizar tudo que é pedido, mas se o grupo quiser adicionar alguma funcionalidade para ajudar a visualização do que está sendo feito, mais portas podem ser adicionadas.
- Ao modelar o diagrama da máquina de estados finita, você pode omitir os estados de erro - estados nos quais a máquina fica "presa", apenas informando que a palavra é inválida - com o objetivo de deixar o diagrama mais claro/limpo. Se isso for feito, o grupo deve documentar essa omissão.

- Todos os passos da elaboração devem ser documentados no relatório, de forma sucinta e significativa, usando ilustrações significativas sempre que possível. Além disso, qualquer decisão extra tomada pelo grupo também deve ser documentada.
- O trabalho deve ser entregue exclusivamente pelo PVANet, sendo que apenas um aluno de cada grupo precisará submetê-lo ao sistema.
- Apenas um arquivo compactado deve ser entregue, contendo: o relatório do trabalho em PDF, os arquivos Verilog (.v) de cada módulo, e, se houver, o arquivo das formas de onda gerados na simulação (.vcd).
- Não há necessidade de colocar código fonte completo no relatório. Isso é redundante, visto que os arquivos do código já devem ser entregues com o relatório.
- Cópias de trabalhos práticos serão exemplarmente penalizadas. A punição será a mesma para quem copiou e para quem forneceu o trabalho prático.