

Elementos de Sistema - Projeto F - Assembly

Rafael Corsi - rafael.corsi@insper.edu.br

Março - 2018

! Prazo original : Terça Feira - 17/4/2018
! Atrasado até : Terça Feira - 24/4/2018

Descrição

Nesse projeto cada grupo terá que implementar diversos codigos em assembly a fim de entendermos a linguagem e as limitações do hardware propostos.

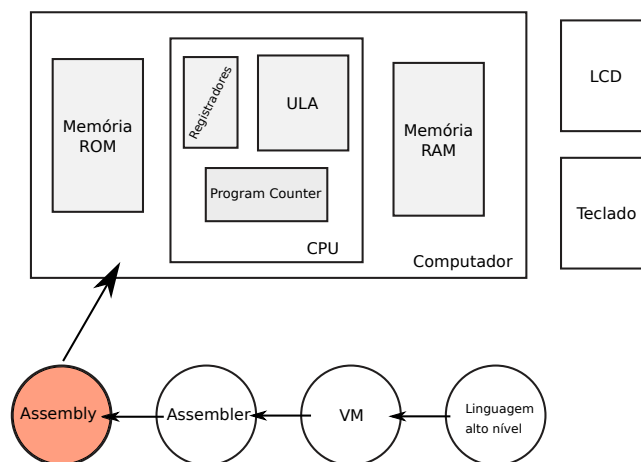


Figure 1: Assembly

Instruções

Seguir as instruções a seguir para desenvolvimento do projeto.

Entendendo a Organização do Projeto

A pasta do projeto D, no repositório Z01, possui a seguinte estrutura:

```
/F-Assembly
  /Quartus
  /script
    testeAssembly.py
    Z01simulator.py
  /src
    /nasm
      *.nasm
  /tests
    /tst
      /abs
      /add
      ....
```

1. Quartus: Projeto Quartus para gravar o computador na FPGA
2. scripts: Scripts em python que automatizam a execução dos testes;
3. src/nasm/*.nasm: Arquivos ASSEMBLY que serão implementados pelo grupo;
4. tests/tst/*: Arquivos que realizam o teste nos arquivos códigos do rtl.

Executando o Script de Teste

Abra o terminal na pasta *F-Assembly/script* e execute o script python localizado nessa pasta:

```
$ python testeAssembly.py
```

O mesmo irá compilar os arquivos src/rtl/*.nasm e executar os testes unitários em cada um deles. Nesse momento do teste, como os módulos não estão implementados, o resultado deverá ser falho.

Esse comando executa um teste unitário em cada um dos módulos, verificando se sua implementação está correta. O resultado é exibido na tela como : **pass** ou **fail**.



Lembrando que o arquivo tests/config.txt define quais testes serão executados.

Projeto

Deve-se implementar diversos programas na linguagem de máquina do Z01 que irão manipular a memória RAM a fim de implementar o que é pedido. **A descrição a seguir está classificada em ordem de dificuldade, começando pelos mais simples.**

Módulos



Esses arquivos estão localizados em F-Assembly/src/nasm/

- mov
 - **Arquivo** : mod.nasm
 - **Descrição** : Encontra o resto da divisão entre RAM[0] e RAM[1] e o salva em RAM[2]
- add
 - **Arquivo** : add.nasm
 - **Descrição** : Adiciona o valor de RAM[1] com RAM[0] gravando o resultado em RAM[2].
- sub
 - **Arquivo** : sub.nasm
 - **Descrição** : Subtração binária do valor de RAM[1] com RAM[0] gravando o resultado em RAM[2].
- abs
 - **Arquivo** : abs.nasm
 - **Descrição** : Copia o valor de RAM[1] para RAM[0] deixando o valor sempre positivo.
- pow
 - **Arquivo** : pow.nasm
 - **Descrição** : Eleva ao quadrado o valor da RAM[1] e o armazena na RAM[0]. (somente para números positivos)
- fatorial
 - **Arquivo** : fatorial.nasm

- **Descrição** : Calcula o fatorial de RAM[0] armazenando o resultado em RAM[1].

-
- mod
 - **Arquivo** : mod.nasm
 - **Descrição** : Encontra o resto da divisão entre RAM[0] e RAM[1] e o salva em RAM[2]

-
- Linha
 - **Arquivo** : line.nasm
 - **Descrição** : Desenha uma linha horizontal no centro do LCD.

- Quadrado - **Arquivo** :
quadrado.nasm - **Descrição**
: Desenha um quadrado no
centro do LCD.

- Letra Grupo - **Arquivo** :
grupo.nasm - **Descrição** :
Desenha a letra do grupo no
LCD.

Testando em Hardware

Seguir o ferramental F-Programando.

Rubricas para avaliação de projetos

Cada integrante do grupo irá receber duas notas: uma referente ao desenvolvimento total do projeto (Projeto) e outra referente a sua participação individual no grupo (que depende do seu papel).

Projeto

Conceito

- | | |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| I | - Menos da metade dos módulos funcionando |
| D | - Ao menos um módulo não está feito e não passa no testes. |
| C | <ul style="list-style-type: none">- Configurou o travis para operar com o novo projeto.- Todos os módulos sendo testados no Travis.- O ramo master passa nos testes do travis- Todos os módulos passam nos testes.- Fez o programa que desenha uma linha no LCD |
| B | - Fez o programa que desenha um quadrado no LCD |
| A | - Fez o programa que senha a letra do grupo no LCD |
-

Desenvolvedor

Conceito

Insatisfatório

- (I)
- Se comprometeu a fazer algum desenvolvimento (kanban) e não fez.
 - Criou o branch mas não fez o desenvolvimento completo.

Satisfatório

- (A)
- Desenvolveu as rotinas atribuídas pelo Facilitador para passarem nos testes!
 - Acompanhou o Kanban board (github project). Ex: Puxou tarefas, etc.
 - Submeteu alterações por pull requests.
-

Facilitador

Conceito

Insatisfatório

- (I)
- Não acompanhou o projeto, deixando os colegas sozinhos.
 - O relatório das atividades não é condizente com o real desempenho dos integrantes (analisado via git).

Satisfatório

- (A)
- Atualizou o repositório pelo Fork.
 - Fez a manutenção do Kanban board (GitHub project). Ex: cria cards, atribui tarefas, da feedback de insues.
 - Aceitou os pull-requests.
 - Resolveu conflitos de merge nos pull requests.
 - Acompanhou o desenvolvimento do grupo, dando o suporte sempre que necessário
 - **Entregou o branch master sem nenhum erro (passando no travis)**
 - Fez o relatório das atividades descrevendo o papel de cada integrante com clareza.
-