

## 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.

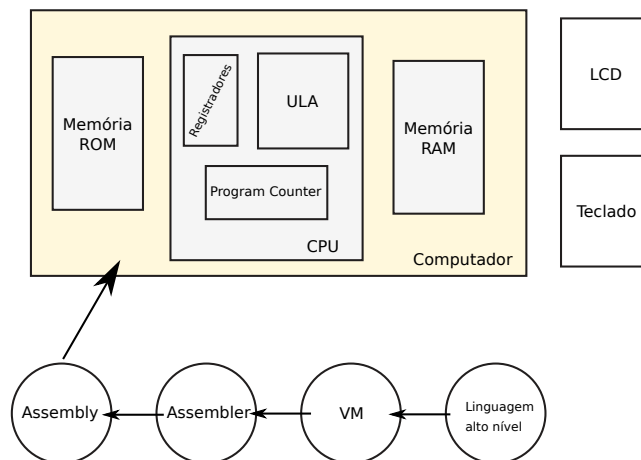


Figure 1: Assembly

### Instruções

Seguir as instruções a seguir para desenvolvimento do projeto.

## Entendendo a Organização do Projeto

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

```
/G-Computador
  /Quartus
  /Z01-Simulator-RTL
  /script
    testeAssembly.py
    testeCU.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. Z01-Simulator-RTL : Pasta com o simulador do computador (usa o hardware criado por vocês)
4. src/nasm/\*.nasm: Arquivos ASSEMBLY que serão implementados pelo grupo;
5. tests/tst/\*: Arquivos que realizam o teste nos arquivos códigos do rtl.

### Testando o Control Unit

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

```
$ python testeCU.py
```



Esse script testa **apenas o Control Unit** e somente uma pequena parte de toda sua funcionalidade, passar nesse teste não indica 100 % que o projeto está correto.

### Testando o projeto completo

Para testar se o computador está correto, iremos executar alguns programas realizados na etapa F-Assembly porém agora no Hardware que vocês montaram. Para isso execute o

```
$ python testeAssemblyMyCPU.py
```

Esse script irá compilar todos os módulos (desde o projeto C) e executar o top level Computador.vhd, iremos comparar se a resposta (memória RAM) possui o resultado esperado. Como os programas são complexos, esperamos com isso conseguir testar uma grande parte do projeto.

## Projeto

Deve-se implementar o Control Unit e integrar os módulos : MemoryIP, CPU e Computador.

## Módulos

Esses arquivos estão localizados em G-Computador/src/rtl/



Os módulos estão listados de maneira Top - Down

- Computador
  - **Arquivo** : computador.vhd
  - **Descrição** : TopLevel do projeto, entidade que integra a memória ROM o MemoryIO, CPU e PLL
  - **Dependências** :
    - \* Dispositivos/ROM/ROM32K.vhd : ROM a ser utilizada no projeto (já foi dado pronto)
    - \* Dispositivos/PLL/PLL.vhd : PLL a ser utilizada no projeto (já foi dado pronto)
- MemoryIO
  - **Arquivo** : MemoryIO.vhd
  - **Descrição** : Faz o mapa de memória para a CPU.
  - **Dependências** :
    - \* Dispositivos/RAM/RAM16K.vhd : RAM a ser utilizada no projeto (já foi dado pronto)
    - \* Dispositivos/Screen/Screen.vhd : Controlador do LCD a ser utilizada no projeto (já foi dado pronto)
- CPU
  - **Arquivo** : CPU.vhd
  - **Descrição** : CPU do Z01 integra registradores, controlUnit, ULA e PC.

- **Dependências :**
  - \* ControlUnit.vhd : Unidade de controle a ser implementada
  - \* ULA.vhd : Unidade lógica desenvolvida no projeto D
  - \* PC.vhd : Program counter do projeto E
  - \* Registradores, Mux : Projeto E

- 
- ControlUnit
    - **Arquivo :** ControlUnit.vhd
    - **Descrição :** Unidade de controle da CPU do Z01.
    - **Dependências :**
      - \* não há

## Diagramas

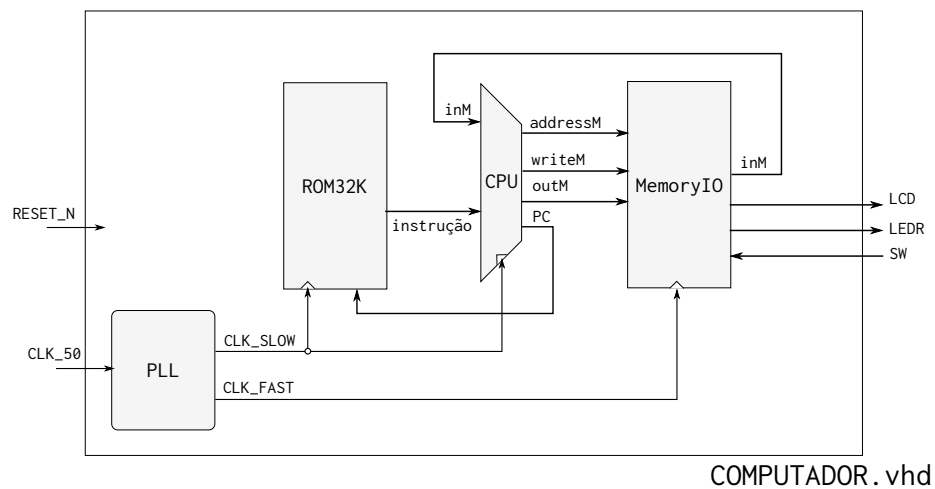


Figure 2: Computador.vhd

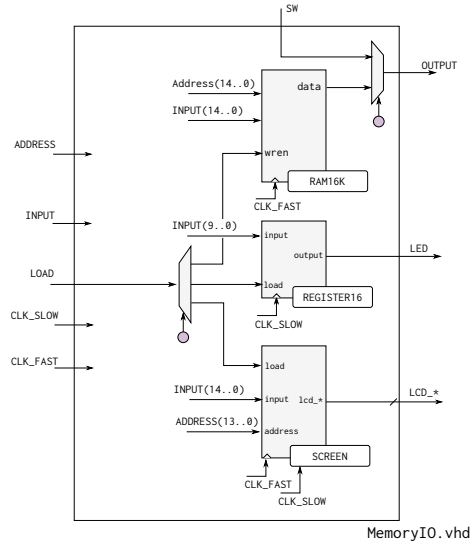
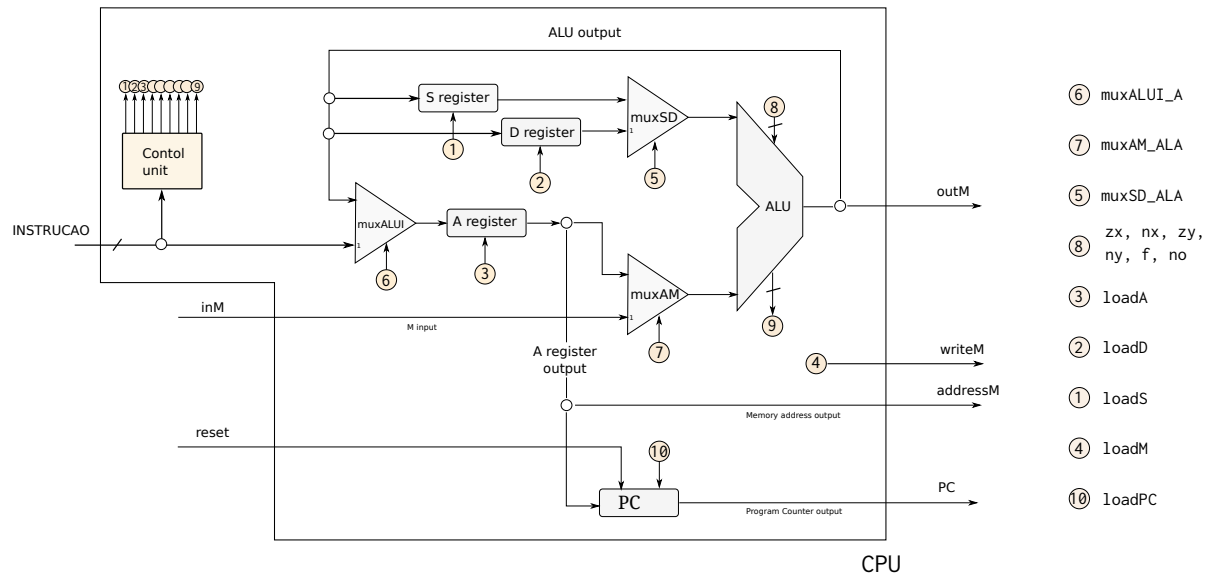


Figure 3: MemoryIo.vhd



- ⑥ muxALUI\_A
- ⑦ muxAM\_ALA
- ⑤ muxSD\_ALA
- ⑧ zx, nx, zy, ny, f, no
- ③ loadA
- ② loadD
- ① loadS
- ④ loadM
- ⑩ loadPC

Figure 4: CPU.vhd

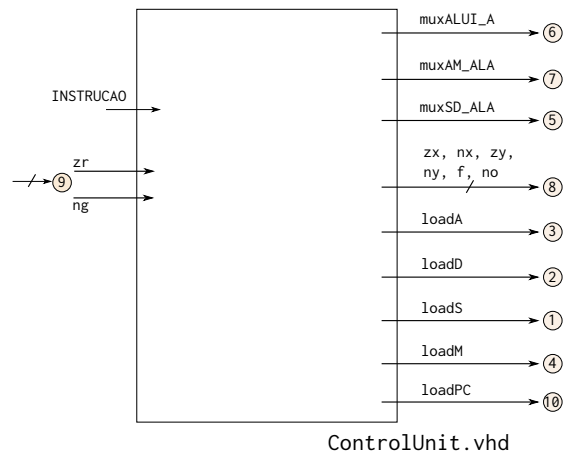


Figure 5: ControlUnit.vhd

## 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 | - Construiu com os módulos do grupo o seu próprio computador<br>- Todos os módulos sendo testados no Travis.<br>- Todos os módulos passam nos testes.<br>- Executa o programa que desenha uma linha no LCD usando o seu computador (grava na FPGA) |
| B | - Adiciona o teclado como um periférico a mais do LCD e demonstra o seu funcionamento ou,<br>- Adiciona os displays de 7 segmentos da FPGA como periféricos do Computador e demonstra o seu funcionamento  |
| A | - Adiciona e demonstra os dois periféricos extras  |
-

## 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.
-