

Prezados,

para que o script teste / simulador funcione teremos que definir um padrão para o nome da entidade CPU no arquivo **Computador.vhd**, o port map da CPU precisa necessariamente chamar **MAIN\_CPU**, caso contrário o simulador não **funcionará**.

### Computador.vhd :

```
MAIN_CPU : CPU PORT MAP (  
    clock      => CLK_SLOW,  
    inM        => OUTPUT_RAM,  
    instruction => INSTRUCTION,  
    reset      => RST_CPU,  
    outM       => INPUT,  
    writeM     => LOAD,  
    addressM   => ADDRESS,  
    pcout      => PC  
);
```

Outra coisa que é necessária para o simulador funcionar, é utilizar os sinais já definidos no CPU.vhd, lá estão padronizados os nomes que controlam os mux, regs e ULA, na hora de executarmos os testes verificamos por esses sinais.

Além dessa definição, precisamos também padronizar o nome da ROM32K

```
ROM : ROM32K PORT MAP (  
    CLK_FAST => CLK_FAST,
```

E do MemoryIO :

```
MEMORY_MAPED : MemoryIO PORT MAP (  
    CLK_FAST => CLK_FAST,
```

## MemoryIO.vhd :

Será necessário também padronizar o nome da memória RAM dentro do MemoryIO:

```
RAM: RAM16K PORT MAP(  
address
```

A onde está isso ? No arquivo list.do dentro da pasta Z01-Simulator-RTL/do/list.do :

```
add list /computador_tb/c1/MAIN_CPU/clock  
add list /computador_tb/c1/MAIN_CPU/instruction  
add list /computador_tb/c1/MAIN_CPU/pcout  
add list /computador_tb/c1/MAIN_CPU/s_regDout  
add list /computador_tb/c1/MAIN_CPU/s_regSout  
add list /computador_tb/c1/MAIN_CPU/s_regAout  
add list /computador_tb/c1/MAIN_CPU/s_muxALUI_A  
add list /computador_tb/c1/MAIN_CPU/s_muxSD_ALU  
add list /computador_tb/c1/MAIN_CPU/outM  
add list /computador_tb/c1/MAIN_CPU/writeM  
add list /computador_tb/c1/MAIN_CPU/inM
```

## DICA :

Utilize o projeto do Quartus que está na pasta do projeto G para validar se a implementação está correta, para isso gere os RTLs e valide com o computador proposto.

