# LABORATÓRIO DE ARQUITETURA DE COMPUTADORES

## Projeto Final

2017

# SUMÁRIO

# ATENÇÃO

Este arquivo pode ser atualizado à medida que dúvidas forem surgindo. As atualizações estarão sempre em cor vermelha. Sempre que houver uma atualização uma mensagem será enviada através do AVA.

## 1 Introdução

MIPS R2000 é um microprocessador de arquitetura RISC criado em 1986 com 5 estágios pipeline e 450.000 transistores. Este foi o primeiro microprocessador RISC comercial. O MIPS possui instruções de tamanho fixo de 32 bits e 32 registradores de propósito geral de 32 bits cada um. O registrador 0 sempre possui valor zero. O tamanho da palavra de memória é de 32 bits. O documento Cap13.pdf contém mais informações sobre este microprocessador, assim como um exemplo de implementação simplificada (monociclo) em VHDL. Outra fonte sobre a especificação das instruções MIPS é o livro "Organização e Projeto de Computadores" do autor Hennessy Patterson, que possui exemplares na biblioteca.

A disciplina de Circuitos Digitais introduziu os conceitos básicos de projeto de circuitos, enquanto a disciplina de Arquitetura e Org. de Computadores introduziu o conceito de arquitetura de instruções e linguagem de máquina. Nos experimentos a arquitetura do MIPS a ser utilizada no projeto foi apresentada. Utilize os materiais de aula e livros textos para realizar o projeto.

## 2 Descrição do Projeto

Utilize o projeto do MIPS fornecido como base. O projeto deverá conter os módulos da CPU (ULA e UC) e Memória. O MIPS utiliza memórias separadas para instruções (program.mif) e dados (dmemory.mif). Complemente o projeto realizando as alterações necessárias para suportar as instruções: J_format (J e Jal), R_format (Jr, Sll e Srl) e I_format (Addi, Bne). Para a visualização de resultados na placa, deve ser complementado o multiplexador que alimenta a apresentação do display de LCD.

Uma vez complementado o processador MIPS, escreva um código em linguagem de máquina para ler o número de elementos de um vetor e ler os elementos deste vetor, diretamente da memória. Depois apresentar no display de LCD o maior valor, o menor valor e a soma dos valores. O código de máquina no arquivo MIF[1] deve vir com o código assembly correspondente comentado a frente.

**Observação1**: Condições de erro devem ser detectadas e tratadas de acordo com decisão de projeto de cada grupo. Esta decisão TEM QUE constar na documentação.

**Observação2**: Toda e qualquer dúvida deve ser tirada com o(a) professor(a). Caso não seja possível, uma decisão deve ser tomada e registrada na documentação do projeto.

[1] O arquivo de memória inicial do simulador (mif) é um arquivo texto, portanto, para facilitar a entrada do programa em linguagem de máquina, pode ser utilizado um simulador assembly do MIPS que gere o código de máquina correspondente.

# 3 Organização da Memória


Deve ser reservada região de memória específica para programas, sub-programas e dados. É utilizada memória separada para programa (program.mif) e dados (dmemory.mif), da mesma forma que está implementado no MIPS. O tamanho da memória deve ser suficiente para o armazenamento. NÃO ESQUECER DE DEFINIR O FORMATO DE APRESENTAÇÃO (RADIX) DO ARQUIVO DE MEMÓRIA INICIAL (MIF).

# 4 Documentação

Deve ser entregue uma documentação do projeto com toda e qualquer informação necessária para a compreensão das decisões de projeto, codificação, simulação e teste funcional. **O teste de simulação de cada instrução implementada deve ser apresentado separado**. Pode ser utilizado um arquivo de forma de onda de simulação (waveX.do) e memória de instrução (programX.mif) diferente por teste de instrução, desde que especificado na documentação. Inclua o código de máquina do programa feito para vetores, com código *assembly* incluído como comentário, em um arquivo de nome **vetor.mif**. Além dos dados dos membros do grupo. **Deixe claro o que foi implementado, se há extras, e como sua implementação funciona**. O relatório deve ser alto-contido, não deve ser preciso consultar outros arquivos ou imagens para compreendê-lo. Os arquivos adicionais entregues devem ser apenas para testar o projeto e não para a compreensão do relatório.

O Quartus II gera um conjunto de arquivos de dados que auxiliam no processo de compilação e simulação. Estes arquivos ocupam um extenso espaço em disco e **não devem** ser entregues com o projeto. Então, além do arquivo de documentação deve ser entregue **todos e somente** os arquivos gerados pelo desenvolvedor (vhdl, do, mif, csv, etc.) de forma a reduzir o tamanho do arquivo postado no AVA. Exporte sua configuração do Pin Planner através o menu File-Export (.csv), se houve alteração. Devido a isto, **a documentação também deve conter instruções sobre como reconstruir (criar) o projeto a partir dos arquivos entregues**.

ATENÇÃO 1: garanta que os arquivos entregues, quando simulados, geram os resultados corretos e realmente validem a funcionalidade.

ATENÇÃO 2: não obedecer a qualquer das regras acima implicará em descontos na nota. A entrega de todos os arquivos de projeto e não apenas os arquivos gerados pelo desenvolvedor acarretará desconto de 0,5 ponto.

# 5 Prazos

O projeto será entregue em 2 fases, cada qual receberá uma nota. A fase intermediária terá a **apresentação funcional** na placa, sem direito a reapresentação. A fase final terá a **apresentação funcional** na placa, sem direito a reapresentação, e inclui um **relatório**. Para cada apresentação vocês deverão **gerar um conjunto de instruções** (assembly e código de máquina) para testar as instruções, e criar o arquivo.mif

correspondente com o respectivo **código assembly comentado** por instrução. O código gerado não precisa ter sentido, mas TEM QUE mostrar que cada uma das instruções está funcionando corretamente.

- **Fase Intermediária** (09/06 aula): teste e simulação de funcionamento das instruções J, Bne e Jal. Entregar os arquivos gerados (vhd) e as instruções de máquina com assembly comentado. O código de máquina deve mostrar que as instruções projetadas funcionam (Nota peso 3).

Não entregar a fase intermediária no prazo ou não apresentar na aula (presença de todos os membros do grupo) implica em nota zero para a fase. Erros de funcionamento ou testes inadequados acarretam notas parciais, podendo ser nula.

- **Fase Final**: entrega do código com todas as instruções e um relatório final. Apresentação na placa. (Nota Peso 7)

  - ✓ **Código (arquivos gerados) e Relatório (arquivo pdf)**

  Data: 30/06

  Horário: 23:55h entrega no AVA e apresentação na aula.

Para código e relatório entregar arquivo único de NOME: <T>_Grupo<NN>.zip ou <T>_Grupo<NN>.rar, onde <T> deve ser substituído pela turma (A ou B) e <NN> pelo número do grupo com 2 dígitos. Exemplo: grupo 1 da turma A, Deverá ter nome "*A_Grupo01*". Será descontado 0,5 ponto por nome de arquivo errado.

Atraso aceito apenas para relatório: desconto de 0,5 ponto para entrega até às 24:00h após o prazo, 1 ponto para entrega até às 36:00h do prazo e 2 pontos para entrega até as 48:00h do prazo . Não serão aceitos em hipótese nenhuma projetos entregues após este último prazo.

**Avaliação Oral Individual e Apresentação do grupo (na placa)**

Data: 30/06 Apresentação, 07, 14/07 (e 21/07 se necessário) avaliação oral

Horário de aula

Local: LSA (ATLab-104)

A ordem das apresentações será sorteada. Atrasos podem incorrer em descontos na nota.

# 6 Critérios e Pesos

Grupos conforme definido para os experimentos.

A fase 1 receberá peso 3, enquanto a fase final receberá peso 7.

Na avaliação será considerada a qualidade quanto a:

- aplicação de conceitos,
- aplicação das técnicas de codificação e projeto,
- documentação compreensível e completa,
- documentação de código e esquema (comentários, uso adequado de identificadores (portas e sinais) e organização (ordenação, identação, arquivos, componentes, etc.)
- Funcionalidade
- Testes realizados