

Trabalho Prático – Arquitetura de Computadores I (Versão 9)

Escopo do trabalho:

O Trabalho Prático (TP) consiste em **projetar/implementar em VHDL** uma versão do **MIPS com pipeline**, explorando os conceitos apresentados em sala de aula.

A arquitetura implementada deverá ser capaz de **executar programas assembly, descritos e validados com a ferramenta MARS**.

A implementação partirá de uma descrição VHDL da arquitetura MIPS trabalhada pelos alunos na disciplina de Organização de Computadores. Ao final do trabalho deverá ser disponibilizada uma arquitetura do mesmo processador empregando pipeline com as técnicas descritas neste documento.

Autoria dos trabalhos:

O trabalho poderá ser desenvolvido em **grupos de até dois alunos**.

Material a ser entregue ao professor:

Cada grupo deverá preparar um documento de projeto completo descrevendo tudo o que foi realizado. O documento de projeto deverá ser claro o suficiente, de forma a ser utilizado pelo professor como um manual para o desenvolvimento de programas para a arquitetura proposta. O documento deverá possuir os detalhes necessários para que todo o fluxo de projeto seja executado sem a necessidade de consultar os projetistas (alunos). Mais precisamente, **é obrigatório o detalhamento do fluxo de projeto** que inicia com a validação do código assembly no ambiente MARS até a simulação da arquitetura descrita em VHDL. Para as simulações empregando os arquivos VHDL podem ser usadas as ferramentas aprendidas em Organização de Computadores (e.g., ISE ou Modelsim).

Os arquivos a serem entregues em cada etapa deverão ser postados em um único arquivo, compactado (.zip). O nome do arquivo compactado deverá ser composto pelo(s) nome(s) completo(s) do(s) autor(es), sem espaços, conforme o exemplo: Nome1Sobrenome1_Nome2Sobrenome2.zip.

O desenvolvimento do presente trabalho deverá ser realizado em 4 etapas.

T1 (1 ponto) - *Estudar o MIPS multiciclo para converter no MIPS pipeline.*

Os objetivos da primeira etapa são: *(i)* compreensão da arquitetura em VHDL, bem como do *testbench* para validação da mesma, *(ii)* implementação de programas de verificação, e *(iii)* compreensão das alterações necessárias pra transformar a arquitetura multiciclo em pipeline. Espera-se:

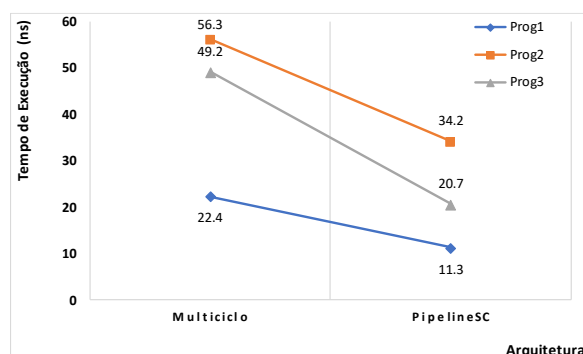
- **(0,5 pontos) Um arquivo PDF** com a seguinte estrutura:
- Introdução
 - Explicação do que o aluno irá apresentar no documento;
 - Organização do documento.
- Compreensão da arquitetura MIPS multiciclo
 - Quais instruções serão exploradas pelo aluno (add, addiu, beq, bne, etc.). OBS.: caso a versão utilizada pelo aluno não disponha da instrução NOP, esta deve ser implementada;
 - Deverão estar especificadas características da arquitetura tais como:
 - Máquina de estados, registradores e suas finalidades.
 - O VHDL da arquitetura MIPS multiciclo é disponibilizada para o aluno na área de Recursos do Moodle.
- Ferramentas adicionais
 - Quais ferramentas serão utilizadas na validação e por que (e.g., MARS, Modelsim).
- Validação
 - Qual o procedimento de validação do material que está sendo disponibilizado pelo aluno;
 - Quais programas foram desenvolvidos para esta etapa;
 - Como gerar os arquivos que são empregados na simulação em VHDL.
- Apresentar o fluxo de validação com a inclusão de imagens de execução no ambiente MARS e a validação do que foi descrito em simulação no ISE ou Modelsim;
- Análise de Projeto para o MIPS pipeline
 - **Explicação detalhada de como planeja alterar a arquitetura multiciclo para torná-la pipeline. Tipicamente deve analisar quais entidades arquiteturas devem ser alteradas; o que deve ser retirado e o que deve ser inserido;**
 - **Planejamento da implementação das barreiras temporais. Estas deverão ser planejadas e justificadas;**
 - **Embora não seja necessária ter a implementação do pipeline funcionando, as barreiras temporais têm que ser ilustradas na documentação (não no fonte VHDL).**

- **(0,5 pontos) Dois arquivos VHDL e três ASM:**
 - Deve ter um arquivo VHDL para a arquitetura multiciclo do MIPS e outro com o *testbench*.
 - **Arquivos ASM:**
 - PROG1.ASM: Um arquivo descrevendo um algoritmo que a partir de dois vetores de entrada ENT_1 e ENT_2, de mesmo tamanho, gera um terceiro vetor de saída SAI, igualmente de mesmo tamanho. Cada elemento do vetor SAI deve ser o resultado de uma operação entre elementos de mesma posição dos vetores ENT_1 e ENT_2. Esta operação será uma **soma** sempre que os elementos dos dois vetores forem igualmente pares ou igualmente ímpares, caso contrário, será uma **subtração**;
 - PROG2.ASM: Um algoritmo de ordenação de vetores que receba 20 inteiros quaisquer e, reordene os mesmos. O vetor original e o ordenado devem estar na memória;
 - PROG3.ASM: Um arquivo contendo uma função que calcula fatorial e um programa que utiliza esta função apresentando o cálculo dos fatoriais de 0 a 5. O resultado do cálculo de cada fatorial deve ser armazenado em um vetor na memória;

T2 (3 pontos) - Implementar a primeira versão do MIPS pipeline.

O objetivo desta etapa é desenvolver uma arquitetura MIPS empregando pipeline de cinco estágios. Esta implementação deverá ser uma evolução da versão multiciclo, explorada na fase anterior e **não deverá dar suporte ao tratamento de conflitos de dados ou controle**. A solução dos conflitos de dados deverá ser feita em software, a partir da inserção de *nops* entre instruções que tenham dependência verdadeira ou ainda através da reordenação das instruções. Solução similar deverá ser aplicada para conflito de controle, ou seja, inserindo instruções de *nop* logo após a instrução de salto para evitar que instruções incorretas sejam carregadas para o pipeline. É pressuposto que o *testbench* apresentado na primeira etapa do trabalho seja reutilizado aqui, com as inserções de *nops* necessárias para o funcionamento do mesmo. Os seguintes recursos são esperados:

- **(1,5 pontos) Um arquivo PDF contendo:**
 - Uma introdução descrevendo o que será apresentado no documento, bem como descrevendo o objetivo da etapa;
 - As características da arquitetura do MIPS empregando pipeline;
 - Uma avaliação **de desempenho** comparado com a arquitetura multiciclo inicial, tal como ilustrado na figura abaixo. Os programas com **adaptação (a inserção de nops)** para rodar na versão pipeline sem tratamento de conflitos.



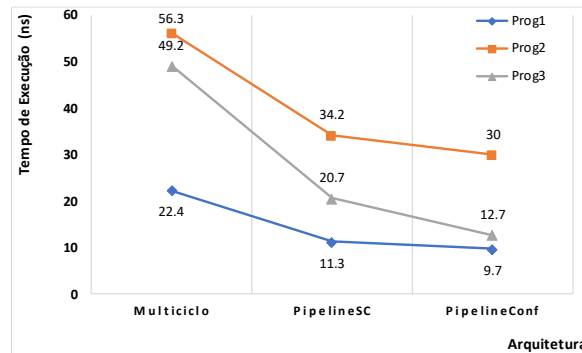
- **(1,5 pontos) Os arquivos fonte:**
 - O arquivo VHDL contendo a descrição da arquitetura com pipeline. Cada bloco operativo deve estar separado em pares entidade-arquitetura;
 - O arquivo VHDL empregado como testbench;
 - Os arquivos assembly empregados nas validações/avaliações apontadas no arquivo PDF.

T3 (4 pontos) - Implementar a segunda versão do MIPS pipeline.

O objetivo desta etapa é implementar a arquitetura do processador MIPS empregando pipeline com tratamento do conflito de dados e controle. Para o conflito de dados, deverá ser incluído o mecanismo de adiantamento e o pipeline deverá ter uma solução de inserção automática de bolha em situações em que o adiantamento não resolva. O tratamento de conflito de controle deve ser feito de forma automática através do emprego da política “**salto nunca ocorre**”. Assim, deve ser excluído do pipeline (*flush*) as instruções que tenham sido inseridas incorretamente, quando um desvio ocorrer. É pressuposto que o *testbench* apresentado na primeira etapa do trabalho seja reutilizado aqui. Os seguintes recursos são esperados:

- **(1,5 pontos) Um arquivo PDF contendo:**
 - Uma introdução descrevendo o que será apresentado no documento, bem como o objetivo da etapa;
 - As características da arquitetura do MIPS empregando pipeline;

- Uma avaliação de ganho/perda de desempenho comparado com os trabalhos T1 e T2. Os programas não precisam mais ter a **adaptação**, pois vão agora ser executados na versão pipeline **com tratamento de conflitos**;

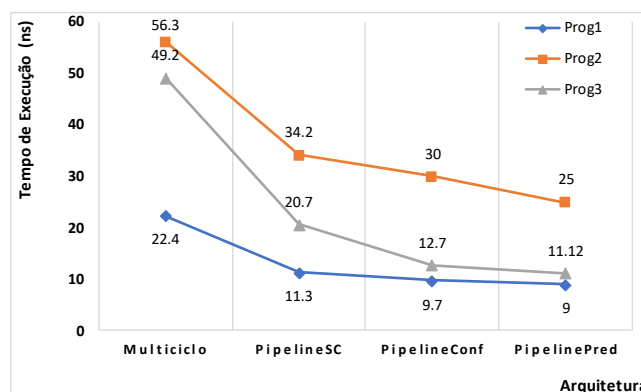


- **(2,5 pontos)** Os arquivos fonte:
 - O arquivo VHDL contendo a descrição da arquitetura com pipeline;
 - Cada bloco operativo deve estar separado em pares entidade-arquitetura.
 - O arquivo VHDL empregado como testbench;
 - Os arquivos assembly empregados nas validações/avaliações apontadas no arquivo PDF.

T4 (2 pontos) - Otimização do mecanismo de tratamento de conflito de controle.

O objetivo da quarta etapa é implementar um mecanismo de predição dinâmica com dois bits partindo inicialmente do estado “salto não ocorre” (*not taken*). **A política deve ser mantida para cada instrução de salto condicional que venha a ser encontrada durante a execução de um programa qualquer.** Note que para manter a política para toda e qualquer instrução de salto condicional, deve existir um mecanismo que associa endereços de salto com os históricos. Implemente o histórico para 4 instruções de desvio. Caso o programa tenha mais de 4 instruções de desvios, as demais devem ser implementadas com a política default *not taken*.

- **(1 ponto)** Um arquivo PDF contendo:
 - Uma introdução descrevendo o que será apresentado no documento, bem como descrevendo o objetivo da etapa;
 - As características da arquitetura do MIPS empregando pipeline;
 - Os meios empregados para a validação;
 - Uma comparação de todas as arquiteturas geradas de ganho/perda de desempenho quando comparado todos os modelos elaborados.



- **(1 ponto)** Os arquivos fonte:
 - O arquivo VHDL contendo a descrição da arquitetura com pipeline;
 - O arquivo VHDL empregado como testbench;
 - Os arquivos assembly empregados nas validações/avaliações apontadas no arquivo PDF.