

Laboratório 02

Disciplina: *Organização e Arquitetura de Computadores – Turma B e Turma D*

Semestre: 1º/2018

Prof.: Flávio Vidal

Título: *Desenvolvimento da Unidade Operativa Pipeline MIPS*

Entrega:

Código-fonte/Relatório (PDF) (Aprender.unb.br): 17/06/2018 até às 23h00min

1. Objetivos

Permitir que o aluno se familiarize com a construção, execução de instruções e controle da unidade operativa *Pipeline MIPS*. O objetivo principal desta atividade é montar os módulos principais *da unidade operativa (Datapath)*, que compõe a arquitetura *MIPS Pipeline*. Estes módulos deverão ser capazes de processar conjuntos de instruções de 32 bits pré-compiladas, respeitando todas as condições de implementações necessárias para a execução destes tipos de instruções, fazendo uso de um conjunto de memória, esta seguindo a proposta apresentada em sala de aula.

O projeto desta disciplina é uma atividade planejada de forma a complementar e reforçar o conteúdo programático da disciplina *Organização e Arquitetura de Computadores*. Espera-se que nas atividades de projeto os grupos de alunos desenvolvam sua capacidade de observação, análise e compreensão das metodologias de organização e arquitetura de computadores.

Desta forma cabe ao(s) aluno(s), partindo da premissa que possui os requisitos para o curso, juntamente com o conteúdo adquirido nas aulas teóricas, desenvolver todas as etapas da implementação solicitada.

2. Metodologia

Deverá ser implementado utilizando o ambiente de desenvolvimento *Quartus II Altera (versão 9.1sp1)* preferencialmente ou qualquer versão gratuita atual que incorpore o módulo de simulação *Vector Waveform Editor*, um projeto **com todos os módulos (incluindo os módulos para tratamento de exceções, hazards de dados/controle e overflow)** que formam a unidade operativa da arquitetura *Pipeline MIPS* (Figura 1) de forma a permitir sua execução automática (sem interferência externo ou input de entrada manual) de todas as instruções armazenadas na memória de instruções, atendendo aos requisitos descritos a seguir.

Etapas 1: Criação da Unidade Operativa Pipeline MIPS

Requisito 1: *Implementação do Controle da Unidade Pipeline MIPS*

A partir da Unidade Operativa Pipeline desenvolvida e apresentada em sala de aula, realize as modificações necessárias (adicionando os módulos de Controle Principal e ULA, memórias, extensão

de sinal, extensão de sinal sinalizada/não-sinalizada, entre outros – vide Figura 1) de forma que esta arquitetura seja capaz de realizar o processamento seguindo a estratégia de *Pipeline completa*, conforme os requisitos apresentados em sala de aula.

Apresente a unidade de Controle (principal e ULA) para esta arquitetura desenvolvida, contemplando os sinais de controle em cada módulo e registradores (incluindo também os registradores internos quando houver).

Requisito 2: Bloco de Memória de Dados e Instruções

Para que seja possível a realização dos testes das instruções a serem utilizadas, faz-se necessária a criação dos blocos de memória de dados e instruções com organização tipo *Harvard*. A idéia básica é utilizar os recursos do Quartus (ex.: *Memory Megafunctions*) nos quais servirão para representar a memória de dados e de instruções. Desta feita, faz-se necessária a utilização do arquivo *PipelineInst.mif* e *PipelineData.mif*, que representam respectivamente o conjunto de endereçamento de memória para instruções e dados, sendo estes fornecidos pelo professor (disponíveis para download no Moodle da disciplina). Nestes arquivos estão os códigos-objeto pré-compilados padrão MIF (MIF é um acrônimo para *Memory Initialization File*) das instruções (segmento .text) e de dados (segmento .data) do arquivo de teste a ser utilizado e auxiliar durante o desenvolvimento do processador. Para facilitar o entendimento das instruções pré-compiladas será disponibilizado o arquivo fonte original (*Pipeline.asm*), que poderá ser alterado para realizar as instruções, salvo problemas com a modificação de endereços para acesso da implementação. Informo que no dia da apresentação serão fornecidos novos arquivos *.mif para as demonstrações.

Com relação aos endereços utilizadas para os dados e instruções é fortemente recomendado a utilização do mapa de endereço do MIPS fornecido pelo MARS, para que as instruções fornecidas ofereçam o mínimo de problemas de incompatibilidade possível no momento da apresentação. Observem que esta é uma sugestão, sendo de responsabilidade e criatividade de cada grupo a organização de endereço necessária.

A partir destes arquivos *.mif e utilizando os recursos do *Quartus*, deverão ser desenvolvido estes módulos de forma a suportar os respectivos tamanhos e codificações necessários para que os dados sejam processados pelos módulos subsequentes, devendo o(a) aluno(a) ser o responsável pela metodologia de sincronização com os demais módulos da unidade operativa desenvolvida.

Requisito 3: Suporte a Tipos de instruções

Para verificar o funcionamento dos módulos, estes deverão ser testados utilizando o ambiente de simulação *Waveform Editor*, em que as entradas serão os bits do campo *op* (no caso da unidade de controle) e o campo *funct* sendo os bits de controle da ULA (*OpAlu*), quando necessário. No caso da operação da ULA, esta deverá realizar a operação completa, selecionando a operação pelos bits *OpAlu*, apresentando o resultado na saída da unidade da operação realizada, selecionando o endereço e o conteúdo do registrador de destino. Reforça-se que deverá ser implementado todas as instruções que estiverem sido apresentadas/solicitadas listadas a seguir.

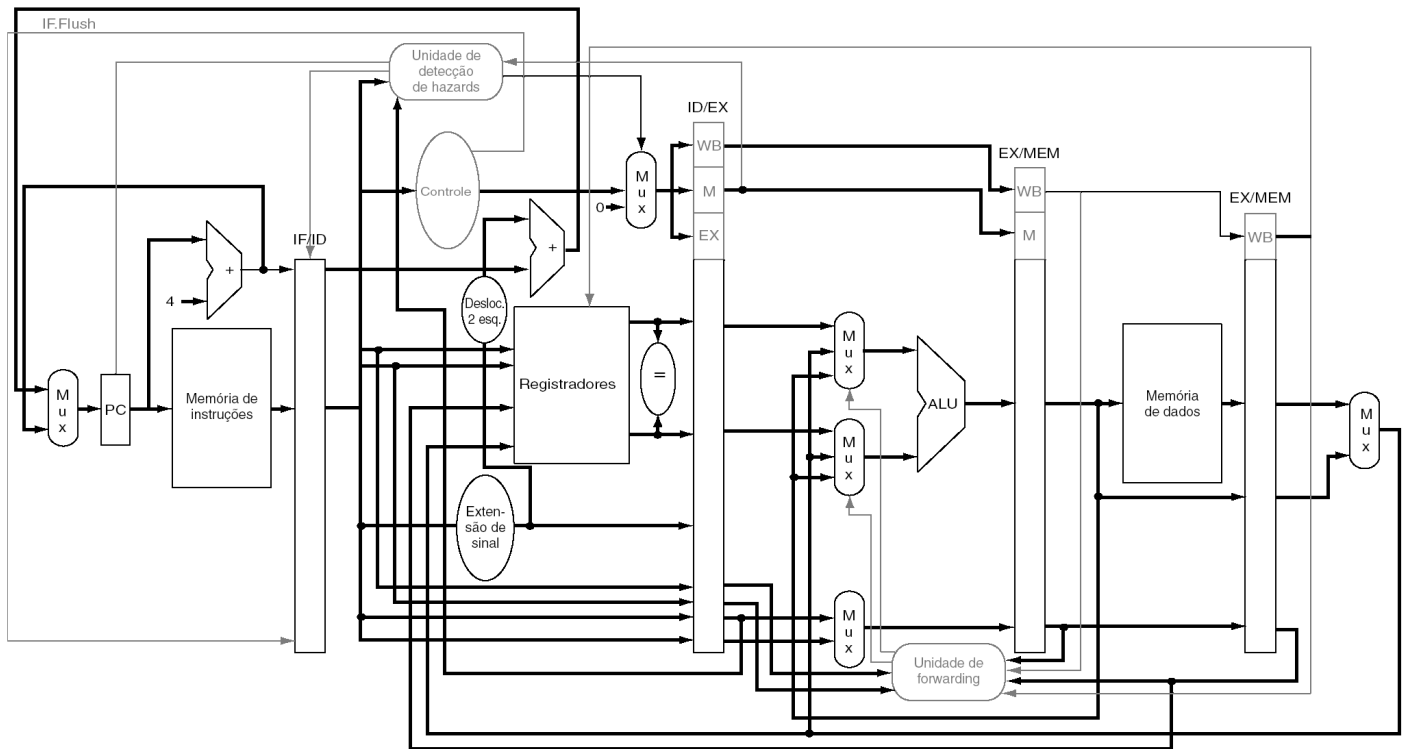


Figura 1 - Unidade Operativa MIPS Pipeline (versão resumida).

Este requisito é a referência de tipos de instruções que deverão ser implementadas nos módulos, de forma a serem suportadas durante todo o processo de execução da unidade operativa, a saber:

- `lw $t0, OFFSET($s3)`
- `add/sub/and/or/nor/xor $t0, $s2, $t0`
- `sw $t0, OFFSET($s3)`
- `j LABEL`
- `jr $t0`
- `jal LABEL`
- `beq/bne $t1, $zero, 0xFFFF`
- `slt $t1, $t2, $t3`
- `lui $t1, 0xFFFF`
- `addu/subu $t1, $t2, $t3`
- `sll/srl $t2, $t3, 10`
- `addi/andi/ori/xori $t2, $t3, -10`

- `mult $t1, $t2`
- `div $t1, $t2`
- `li $t1, XX` (incluindo forma pseudo)
- `mfhi/mflo $t1`
- `bgez $t1, LABEL`

Para a instrução `mult`, esta poderá ser desenvolvida utilizando um dos circuitos multiplicadores apresentados em sala de aula, ou o Algoritmo de Booth, devendo esta ser um módulo à parte no processador utilizando o seu respectivo coprocessador. Uma das possíveis implementações deste algoritmo em hardware pode ser encontrada no link: http://www.pcs.usp.br/~labdig/pdf/files_2010/multipl_sinal.pdf. Lembrando ainda que o resultado da instrução `mult` deverá obrigatoriamente colocar os valores nos registradores `$hi` e `$lo`. Como o intuito desta atividade se faz na implementação do *Pipeline MIPS*, caso o grupo deseje, este poderá utilizar os recursos de funções implementadas no *Quartus* do tipo *Megafunction*. O equivalente pode ser realizado com a instrução `div`.

Requisito 4: Análise de Desempenho das Unidades Desenvolvidas

Nesta etapa, por meio da utilização do ambiente de simulação *Waveform Editor*, deverá ser apresentado as instruções que estão sendo executadas a cada etapa do Pipeline construído. Estas instruções deverão ser lidas e apresentadas com seus respectivos conteúdos, de forma a permitir a verificação dos valores contidos na saída da memória de instruções implementada, incluindo também os demais módulos (banco de registradores, saída da ULA,...). As informações devem ser apresentadas tendo como base todos os valores armazenados nos registradores que compõem os módulos do caminho de dados Pipeline, *incluindo os bits de controle*.

Deverá ser feita a análise do tempo de execução de cada unidade funcional desenvolvida na execução das instruções utilizadas, de forma a identificar os pontos de “gargalos” (*bottlenecks*) em que o atraso na propagação do sinal de controle (e/ou dados) prejudica o desempenho desta unidade funcional. Construa uma tabela para cada unidade funcional, especificando o tempo total e média de cada instrução por módulo componente da unidade funcional, apresentando preferencialmente o valor absoluto (em submúltiplos de segundos) e o percentual dos valores correspondente ao tempo utilizado.

A simulação deverá utilizar o modo de compilação *Timing* de forma a levar em consideração os atrasos de cada unidade/elemento funcional utilizado, de maneira a estimar a duração do *Ciclo de Clock* necessário que contemple o tempo de execução da instrução que maior duração do caminho de dados Pipeline MIPS, sendo este o período de *Clock* aplicado às demais instruções.

Se faz como exigência que o arquivo de simulação do *Waveform Editor* esteja configurado de forma a apresentar claramente na tela para avaliação durante a apresentação, o conteúdo de todos os registradores, utilizados ou não.

Observações Importantes:

1 – A data de apresentação contemplará duas aulas, como apresentado no cronograma de atividades e em horário de aula, sendo o local da apresentação no LISA (Lab. 1) localizado no prédio CIC/EST. Cada grupo terá disponível até 15min para realizar o processo de apresentação do código disponibilizado funcionando corretamente. Neste semestre será avaliado três códigos distintos e em sequência do nível fácil ao nível difícil, de forma que o grupo só se habilitará para apresentar o segundo código, se e somente se o primeiro estiver funcionando perfeitamente. O terceiro código será apresentado somente se o segundo código estiver funcionando corretamente. Cada código avaliado contemplará 33,3% da nota de apresentação. A implementação que não compilar no dia de sua apresentação será atribuída nota 0 (zero) na nota de apresentação.

2 - Atrasos dos grupos não serão tolerados, salvo situações excepcionais em que o docente tenha compromisso emergencial junto ao departamento, ou motivo pessoal devidamente justificado, sendo as apresentações remarcadas para data conveniente. É de fundamental importância a ciência que o material utilizado para os testes serão aqueles postados no ambiente *Aprender.unb.br*, podendo ser disponível em arquivos separados, no qual o professor é responsável em fornecê-lo no dia da apresentação. Como o tempo e número de dias são limitados, a ordem e dia de apresentação será definida por sorteio, utilizando como pré-requisito os grupos que entregaram no dia anterior os arquivos da implementação e o relatório (PDF). O sorteio será realizado em sala de aula, no dia da apresentação inicial, 10 minutos antes do início das apresentações.

3 – Cada grupo será atendido somente uma vez.

4 – Caso ocorra algum fato excepcional aos abrangidos neste documento, caberá ao professor definir a melhor solução para o problema, respeitando os critérios doravante apresentados.

3. Grupos

Neste projeto será permitido a formação de grupos de no máximo 3(três) alunos. A partir do grupo formado, deverá ser indicado um líder, no qual este líder será o responsável pelo envio dos arquivos fontes para o sistema *Aprender.unb.br*. Somente serão aceitos os arquivos fontes enviado pelo líder do grupo. Cabe ao grupo indicar no relatório o nome de todos os membros.

4. Relatório

O relatório deve demonstrar que a respectiva atividade de laboratório foi realizada com sucesso e que os princípios subjacentes foram compreendidos.

O relatório da atividade de laboratório é o documento gerado a partir do trabalho realizado seguindo as orientações exigidas na metodologia de laboratório. Este deve espelhar todo o trabalho desenvolvido nos processos de obtenção dos dados e sua análise. Apresentamos a seguir uma recomendação de organização para o relatório da atividade de laboratório. Deverá conter as seguintes partes:

i. Identificação: Possuir a indicação clara do título do experimento abordado, a data da sua realização, a identificação da disciplina/turma, os nomes dos componentes do grupo, número de matrícula e email.

ii. Objetivos: Apresentar de forma clara, porém sucinta, os objetivos do laboratório.

iii. Introdução: Deve conter a teoria necessária à realização da atividade de laboratório.

iv. Materiais e Métodos: É dedicada à apresentação dos materiais e equipamentos, descrição do arranjo experimental e uma exposição minuciosa do procedimento de laboratório realmente adotado.

v. Resultados: Nesta parte são apresentados os resultados das implementações efetuadas, na forma de tabelas e gráficos, sem que se esqueça de identificar em cada caso os parâmetros utilizados.

vi. Discussão e Conclusões: A discussão visa comparar os resultados obtidos e os previstos pela teoria. Deve se justificar eventuais discrepâncias observadas. As conclusões resumem a atividade de laboratório e destacam os principais resultados e aplicações dos conceitos vistos.

vii. Bibliografia: Citar as fontes consultadas, respeitando as regras de apresentação de bibliografia (autor, título, editora, edição, ano, página de início e fim).

O relatório do laboratório deverá ser confeccionado em editor eletrônico de textos, utilizando o padrão de formatação descrito no arquivo de exemplo, disponibilizado no website da disciplina (oficial e/ou mirror). Está disponibilizado um único padrão de formatação para editores científicos LATEX (arquivo extensão *.zip contendo arquivo de exemplo do uso do pacote), cabendo ao grupo a escolha de qual editor Latex será utilizado. Este modelo pode ser acessado no aprender da disciplina. Somente serão aceitos para avaliação relatórios em PDF feitos seguindo esta formatação.

Todos os arquivos dos projetos desenvolvidos deverão ser entregues via *upload* no ambiente Aprender.unb.br, em arquivo *.zip completando a atividade designada ao laboratório correspondente (vide ambiente aprender.unb.br para maiores detalhes).

Os arquivos de projeto deverão ser criados seguindo os princípios de funcionamento do ambiente Quartus II, de forma que estes poderão ser abertos e verificados no ambiente de desenvolvimento. Não serão aceitos trabalhos entregues fora do prazo estipulado, sendo atribuída nota zero ao grupo. Não serão aceitos qualquer tipo de material (relatório e arquivos de projeto) via email do professor. O único método de envio deverá ser feito pelo Aprender.unb.br.

Vale ressaltar que será atribuída nota zero, definida como atividade “incompleta”, ao grupo que não entregar o relatório e/ou arquivos de projeto devidamente identificado. Entende-se como atividade completa relatório no escaninho do professor e arquivos de projeto corretamente enviados ao endereço eletrônico *Aprender.unb.br*.

a. Critérios Empregados na Correção do Relatório de Laboratório

A avaliação dos relatórios terá em consideração os seguintes itens:

No.	Item	Descrição	Peso (%)
1	Apresentação	Qualidade dos gráficos, impressão, tabelas, vocabulário, legendas, etc.	10%
2	Aspectos Teóricos	Apresentação e descrição da base teórica utilizada. Avaliação da bibliografia utilizada se necessário.	20%
3	Materiais e Métodos	Descrição de todos os procedimentos utilizados, contemplando dados técnicos,	10%

		bem como a metodologia utilizada no decorrer do projeto.	
4	Resultados	Todos os resultados alcançados no projeto.	30%
5	Discussão e Conclusões	Discussão objetiva e devidamente explicada a respeito do projeto. Inclui-se também a pontuação por iniciativa.	30%

Dúvidas deverão ser encaminhadas ao fórum de discussão específico no ambiente *Aprender.unb.br*.