

ESCOLA: Escola Politécnica

DISCIPLINA: Organização e Arquitetura de Processadores

CÓDIGO: 98G03

CRÉDITOS: 04

SEMESTRE: 2022/I

CARGA HORÁRIA: Total: 60 hs **Presencial** **On-line**

PROFESSOR (ES)

NOME	E-MAIL
Ney Laert Vilar Calazans	ney.calazans@puccs.br

EMENTA: Apresentação de modelos gerais para arquiteturas e organização de computadores. Introdução aos elementos definidores de uma arquitetura de processador: conjunto de registradores, conjunto de instruções, formatos de instrução, modos de endereçamento, linguagem de montagem, modelo de acesso a memória, programação em linguagem de montagem. Uso de linguagens de descrição de hardware para especificar processadores. Estudo do modelo bloco de dados-bloco de controle para o projeto de processadores. Implementação de módulos de sistemas computacionais de processadores.

OBJETIVOS DE APRENDIZAGEM/COMPETÊNCIAS

O cumprimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Relacionar conceitos básicos de sistemas digitais com os conceitos abstratos subjacentes a arquiteturas e organizações de processadores.
2. Aprender os conceitos fundamentais relacionados a conjuntos de instruções de processadores, à linguagem de montagem destes e à linguagem objeto subjacente.
3. Dominar a programação de um processador usando sua linguagem de montagem.
4. Relacionar o código de programas escritos em linguagens de alto nível com o código em linguagem de montagem equivalente a este, para um processador dado.
5. Estender o domínio da arquitetura e organização de um processador à compreensão de conceitos de processadores em geral.
6. Compreender como uma organização de processador executa programas escritos em linguagem de montagem.
7. Distinguir os componentes do bloco de dados e do bloco de controle de um processador quanto a função, composição e estrutura.

8. Compreender a estrutura de hardware de organizações de processadores em linhas gerais e associar corretamente diagramas de blocos a descrições destes em modelos executáveis.
9. Compreender um conjunto de conceitos fundamentais úteis para aumentar o desempenho de processadores, incluindo o uso de paralelismo de execução em hardware e aceleradores de hardware específicos.
10. Entender os fundamentos e algumas das abstrações relacionados a operações de entrada e saída e a hierarquias de memória em processadores.

PROCEDIMENTOS METODOLÓGICOS

A Unidade 01 comporta 30% da disciplina. Os itens 1.1 e 1.2 desta Unidade são cobertos em sequência, e o item 1.3 perpassa a abordagem do item 1.2. O item 1.1 é a base da Unidade, ele caracteriza uma arquitetura de processador como um conjunto de seis abstrações que interagem para definir a visão arquitetural de um hardware que executa software. O item 1.2, por outro lado, baseia-se em um estudo de caso de processador (recomenda-se o uso de um processador real, tal como o MIPS, uma arquitetura RISC) e explora em detalhe a programação deste em linguagem de montagem. Uma abordagem diferencial é empregada para mostrar como decisões arquiteturais em diferentes processadores levam a diferentes características operacionais, usando o processador alvo de estudo e o comparando com outro(s) processador(es). Os alunos realizam um trabalho prático onde pesquisam um processador diferente do visto em aula. Para o item 1.3, recomenda-se o uso de uma linguagem de programação de alto nível ou pseudocódigo, desde que a escolha contenha estruturas para definir seleção, laços, expressões aritméticas etc.

A Unidade 02 contempla 22,5% da disciplina. Esta centra esforços no estudo da organização de processadores, explorando em alguma profundidade o projeto de uma organização para um estudo de caso de arquitetura. Recomenda-se usar um subconjunto da arquitetura de processador estudada na Unidade 1. O item 2.1 desta Unidade explora classificações de organizações segundo pelo menos os dois critérios listados aqui, mas abordam-se outras. A essência da Unidade reside no item 2.2, que evolui pela transcrição dos conceitos arquiteturais em hardware que implementa estes. Isto ocorre via a apresentação de diagramas de blocos pelo professor, diagramas estes equivalentes a descrições RTL simuláveis disponibilizadas. Partindo de um subconjunto muito restrito de arquitetura, compreende-se o hardware de base. O professor mostra e faz evoluir este hardware para acomodar novas instruções e estruturas para dar suporte a uma versão mais abrangente da arquitetura de base. Reforça-se a importância de identificar e tratar separadamente os fluxos de informação de controle e de dados. O emprego do modelo Bloco de Dados/Bloco de Controle é indicado como chave.

À Unidade 03 reserva-se também 22,5% da disciplina. Após cobrir a parte essencial do curso nas Unidades 1 e 2, esta Unidade traz uma primeira exploração de dois conceitos essenciais para obter processadores modernos com desempenho aceitável. O item 3.1 esboça rudimentos de paralelismo, sobretudo no nível da execução de instruções (*instruction level parallelism* ou ILP), com a apresentação do conceito de *pipelines*. Isto toma cerca de dois terços do tempo da Unidade. O item 3.2 então explora

como acelerar em hardware o processamento numérico, em especial para representações de números racionais. Isto se faz estudando representações de ponto fixo e ponto flutuante, com ênfase nas últimas. O padrão IEEE-754 é apresentado em linhas gerais. A última parte do item consiste em apresentar como trabalhar com números em ponto flutuante em linguagem de montagem manipulando as representações estudadas.

Na Unidade 04 cabem os restantes 25% do tempo da disciplina. Esta Unidade traz conceitos essenciais para a definição da interface entre processadores e o restante do sistema que este sempre integra. O item 4.1 cobre operações de entrada e saída, explorando a interface entre hardware e software que habilita o processador a enviar e receber dados de/para o mundo exterior. A ênfase é nos tipos possíveis de interação processador-mundo exterior. O item 4.2, por outro lado, explora os rudimentos de hierarquias de memória, iniciando com a justificativa de sua existência (a disparidade de velocidades entre o processador e os diversos tipos de meios de armazenamento de informação), seguindo para a noção de memórias "escondidas" (memórias *cache* ou *cache memories*), geridas de forma automática por hardware dedicado. Deve-se enfatizar que o conteúdo desta Unidade, bem como conceitos adicionais de paralelismo são explorados em maior profundidade na disciplina posterior, Arquitetura e Organização de Computadores (98G05-04), obrigatória para alunos da Engenharia de Computação e optativa para alunos da Ciência da Computação.

O Trabalho Discente Efetivo (TDE) deve cobrir um mínimo de 7,5 horas em atividades extra-classe. Estas atividades estarão distribuídas em estudos na bibliografia, e na preparação dos trabalhos práticos da disciplina, incluindo pelo menos 3h de dedicação ao TP1 e pelo menos 4,5h no TP2.

AVALIAÇÃO

Neste semestre em particular, haverá uma modificação dos pesos típicos atribuídos a provas e trabalhos. Os pesos típicos são dados pela fórmula do grau G1 na primeira forma abaixo, onde os itens Px correspondem a notas de provas escritas individuais e TP corresponde ao conjunto de notas de trabalhos práticos realizados em grupo:

$$G1 = 0,3*P1 + 0,3*P2 + 0,4*TP.$$

Em 2022/1, excepcionalmente, empregar-se-á a fórmula alternativa seguinte:

$$**G1 = 0,2*P1 + 0,2*P2 + 0,6*TP.**$$

Esta mudança se justifica pelo retorno às aulas presenciais depois de longo período com aulas na modalidade remota (sem realização de provas ao longo do semestre) e pela necessidade de adaptação paulatina aos processos de avaliação associados ao modelo presencial, com retomada gradual da valorização de avaliações individuais através de provas escritas.

A prova P1 versará sobre os conteúdos das Unidades 01 e 02 e a Prova P2 sobre os conteúdos das Unidades 03 e 04. Os trabalhos práticos serão em número de dois, sobre os conteúdos das Unidades 01 e 02. Em princípio, estes trabalhos possuem o mesmo peso (ou seja, cada um valendo 30% do G1, mais que cada uma das provas escritas individuais).

BIBLIOGRAFIA

• BÁSICA:

1. PATTERSON, D.; HENNESSY, J. Organização e projeto de computadores: a interface hardware/software. 4a ed., Elsevier, 2014, 709p.
2. STALLINGS, W. **Arquitetura e organização de computadores**. 8ª ed., Pearson, 2010, 625p.
3. HENNESSY, J; PATTERSON, D. **Arquitetura de computadores: uma abordagem quantitativa**. 5ª ed., Campus, 2014, 744p.

• COMPLEMENTAR:

1. TANENBAUM, A.; AUSTIN, T. **Organização estruturada de computadores**. 6a ed., Capítulo 8 - Arquitetura de Computadores Paralelos, pp. 436-518, Pearson, 2013.
2. BAER, J.-L. **Arquitetura de Microprocessadores – do Simples Pipeline ao Multiprocessador em Chip**. 1ª ed., LTC, 2013, 326p.
3. MONTEIRO, M. **Introdução à organização de computadores**. 5a ed., LTC, 2012, 686p.
4. PEDRONI, V. **Eletrônica digital moderna e VHDL**. 1a ed., Campus, 2010, 648p.
5. PARHAMI, B. **Computer Architecture: From Microprocessors to Supercomputers**. Oxford University Press, 2014, 576p.

REFERÊNCIAS ADICIONAIS

• SOFTWARE DE APOIO:

1. MARS (<http://courses.missouristate.edu/kenvollmar/mars>) ou similar, i.e. ambientes de suporte a captura e simulação de programas escritos em linguagem de montagem de um processador selecionado.
2. Vivado, ModelSim ou similares, i. e. ambientes de captura, simulação e/ou prototipação de circuitos e sistemas digitais descritos em linguagem de descrição de hardware.

CRONOGRAMA DE AULAS

Aula	Data	Dia	Conteúdo
1	02/03	Qua	<p>Aula inaugural: Apresentação da disciplina (conteúdo, datas, avaliação, bibliografia etc.)</p> <p>Transparências1-OAP</p> <p>Unidade 01 - Arquitetura de Processadores: Elementos e Linguagem de Montagem</p> <p>Revisão e complementação de conceitos prévios para Organização de Computadores</p> <ul style="list-style-type: none">• Circuitos combinacionais e sequenciais, oscilador em anel• Definição de sistemas digitais, exemplo do tocador de MP3 (entra MP3 e sai PCM, depois um conversor DA e alto falantes)• Noções fundamentais de um computador: informações de dados e controle, blocos de controle e de dados, programas e instruções

2	04/03	Sex	<p>Transparências1-OAP Os seis (6) Elementos Definitórios de uma Arquitetura (1 de 3)</p> <ol style="list-style-type: none"> 1. Conjunto de Registradores Acessíveis (ao programador em linguagem de montagem) 2. Conjunto de Instruções (da arquitetura) 3. Formatos de Instrução (da arquitetura) 4. Modos de Endereçamento (de cada operando de cada instrução) 5. Linguagem de Montagem (da arquitetura) 6. Modelo de Acesso à Memória (pelo processador) <p>Para entender melhor, leiam</p> <ul style="list-style-type: none"> • Sobre tipo, quantidade e tamanho de operandos - HP 2.5 p.77 • Modos de endereçamento principais (há muitos outros) - imediato, direto, base-deslocamento, indexado - PH p.85 • Modelos de acesso a memória - HP 2.3 p.70
3	09/03	Qua	<p>Transparências1-OAP Revisão e complementação de conceitos prévios para Organização de Computadores:</p> <ul style="list-style-type: none"> • Modelo geral de um computador: a interface Processador-Memória • Modelos gerais de organizações de computadores: von Neumann e Harvard <p>Os seis (6) Elementos Definitórios de uma Arquitetura (2 de 3)</p> <ol style="list-style-type: none"> 1. Conjunto de Registradores Acessíveis (ao programador em linguagem de montagem) 2. Conjunto de Instruções (da arquitetura) 3. Formatos de Instrução (da arquitetura) 4. Modos de Endereçamento (de cada operando de cada instrução) 5. Linguagem de Montagem (da arquitetura) 6. Modelo de Acesso à Memória (pelo processador) <p>Para entender melhor, leiam</p> <ul style="list-style-type: none"> • Sobre tipo, quantidade e tamanho de operandos - HP 2.5 p.77 • Modos de endereçamento principais (há muitos outros) - imediato, direto, base-deslocamento, indexado - PH p.85 • Modelos de acesso a memória - HP 2.3 p.70 • Classificação das arquiteturas conforme o conjunto de instruções - HP 2.2 p.67 - máquina de pilha, máquinas com acumulador, máquinas com conjunto de registradores (de uso geral) (subdivididas em registrador-memória e registrador-registrador) • Tipos de operações - HP 2.7 p.79 • Codificação de instruções - HP 2.10 p.86 • A função do compilador - HP 2.11 p.89 <p>Estudo de caso para dominar os 6 elementos <i>definitórios</i> (a arquitetura MIPS 2000)</p> <ul style="list-style-type: none"> • Arquitetura MIPS - características gerais <ul style="list-style-type: none"> ○ Uma <i>instruction set architecture</i> (ISA) de 32 bits <ul style="list-style-type: none"> ▪ A CPU do MIPS possui 32 registradores de 32 bits - \$0 a \$31 ▪ A memória externa é endereçada pelo MIPS por um valor (índice ou endereço) de 32 bits ▪ As operações (lógicas e aritméticas) em sua maioria trabalham sobre dados representados em 32 bits ▪ Toda instrução possui um código objeto (binário) de 32 bits <p>Leiam um pouco sobre o MIPS na Wikipedia Leiam também o Apêndice A do HP Seções A.1, A.2, A.3 e A.10. O link para o material é este: Apêndice A</p>
4	11/03	Sex	<p>Especificação do Trabalho Prático 1 (peso: 20% de G1) Lista de Grupos x Trabalhos Confirmados Material de Apoio para o Trabalho TP1 Exemplo de Trabalho TP1</p> <p>Os seis (6) Elementos Definitórios de uma Arquitetura (3 de 3)</p> <ol style="list-style-type: none"> 1. Conjunto de Registradores Acessíveis (ao programador em linguagem de montagem) 2. Conjunto de Instruções (da arquitetura) 3. Formatos de Instrução (da arquitetura) 4. Modos de Endereçamento (de cada operando de cada instrução) 5. Linguagem de Montagem (da arquitetura) 6. Modelo de Acesso à Memória (pelo processador) <p>Estudo de caso para dominar os 6 elementos <i>definitórios</i> (a arquitetura MIPS 2000)</p> <ul style="list-style-type: none"> • Os modos de endereçamento fundamentais do MIPS <ul style="list-style-type: none"> ○ a registrador ○ imediato ○ base-deslocamento ○ implícito ○ relativo ○ pseudo-absoluto • A linguagem de montagem do MIPS <ul style="list-style-type: none"> ○ sintaxe de uso de instruções em programas (155 instruções - MARS) <ul style="list-style-type: none"> ▪ rótulos, mnemônicos, operandos e comentários ○ pseudo-instruções (388 pseudo-instruções - MARS) ○ diretivas de montagem (20 diretivas - MARS) <ul style="list-style-type: none"> ▪ definição de dados (números e texto; escalares e vetores) ▪ definição das regiões de memória (programas e dados) ▪ alinhamento de memória e reserva de espaço em memória

5	16/03	Qua	<p>Programação em Linguagem de Montagem da arquitetura MIPS (1 de 5)</p> <ul style="list-style-type: none"> • Classes de instruções do MIPS • Tudo que não é instrução em um programa do MIPS pode ser: diretiva do montador, pseudo-instrução ou comentário. Por exemplo, especificação de dados se faz com diretivas • Uso da letra "u" (unsigned) em mnemônicos do MIPS <ul style="list-style-type: none"> ○ gerar interrupções de tratamento de exceções em instruções aritméticas (sem u) ou ignorar tais condições (com u) ○ uso de representações numéricas em complemento de 2 (sem u) ou em binário puro (com u) • Instruções de controle de fluxo: saltos (<i>branches</i> (opcodes 10 e 11), <i>jumps</i> (opcode 4)) • Instruções de comparação (<i>test and set</i>) (opcodes 4 e 8) <i>slts</i> (set if less than) <p>Simulador MARS - documentação e tutorial</p> <ul style="list-style-type: none"> • Simulador MARS - versão para Windows - Mars4_5.jar Instalação e uso • Apêndice A de PH - documentação do processador MIPS • Exemplos a serem apresentados em aula <ul style="list-style-type: none"> ○ Meu primeiro programa MIPS ○ Meu segundo programa MIPS ○ Soma uma constante a cada elemento de um vetor
6	18/03	Sex	<p>Programação em Linguagem de Montagem da arquitetura MIPS (2 de 5)</p> <ul style="list-style-type: none"> • Instruções de comparação (test and set) (opcodes 4 e 8) <i>slts</i> (set if less than) • Especificação dos três exercícios a serem feitos na aula <ul style="list-style-type: none"> ▪ Solução dos três exercícios • Como trabalhar com cadeias de caracteres ou strings: programa toupper.asm e tolower_acent.asm <ul style="list-style-type: none"> ○ Diretivas <code>.ascii</code> e <code>.asciiz</code> ○ Instruções de leitura de byte da memória (<code>lb</code> e <code>lbu</code>) ○ Instrução para escrita de bytes na memória (<code>sb</code>) ○ Grupos de caracteres ASCII (letras maiúsculas, letras minúsculas, caracteres de controle e dígitos decimais)
7	23/03	Qua	<p>Programação em Linguagem de Montagem da arquitetura MIPS (3 de 5) Procedimentos: sub-rotinas e funções - uso de estruturas de pilha no MIPS</p> <ul style="list-style-type: none"> • Passagem de argumentos e retorno de valores <ul style="list-style-type: none"> ○ Via registradores ○ Via pilha ○ Métodos mistos • As instruções <code>JAL</code> e <code>JR</code> (para chamada e retorno de procedimentos) • Chamadas não aninhadas: subrotinas folha - o programa ex_func_call.asm • Chamadas aninhadas - o programa rotinas_aninhadas_com_uso_da_pilha • Chamadas recursivas - caso especial de chamadas aninhadas <ul style="list-style-type: none"> ○ Exemplo de programa com chamadas recursivas de subrotina: o programa fatorial ○ Exercício de recursão: o programa divisão_por_dois_recursoivo <ul style="list-style-type: none"> ▪ Solução: - programa de divisão por 2 recursivo
8	25/03	Sex	<p>Programação em Linguagem de Montagem da arquitetura MIPS (4 de 5) Procedimentos: sub-rotinas e funções - uso de estruturas de pilha no MIPS</p> <ul style="list-style-type: none"> • Chamadas não aninhadas: subrotinas folha - o programa ex_func_call.asm
9	30/03	Qua	<p>Data de Entrega do TP1 Programação em Linguagem de Montagem da arquitetura MIPS (5 de 5) Procedimentos: sub-rotinas e funções - uso de estruturas de pilha no MIPS</p> <ul style="list-style-type: none"> • Chamadas não aninhadas: subrotinas folha - o programa ex_func_call.asm • Chamadas aninhadas - o programa rotinas_aninhadas_com_uso_da_pilha • Chamadas recursivas - caso especial de chamadas aninhadas <ul style="list-style-type: none"> ○ Exemplo de programa com chamadas recursivas de subrotina: o programa fatorial ○ Exercício de recursão: o programa divisão_por_dois_recursoivo <ul style="list-style-type: none"> ▪ Solução: - programa de divisão por 2 recursivo
10	1ro/04	Sex	<p>Disponibilização dos TP1s para avaliação pelos colegas Aula 1 de 7 - Unidade 02 Unidade 02 - Organização de Processadores: Fundamentos e Estudo de Caso</p> <ul style="list-style-type: none"> • Um exemplo de Circuito - o <code>Fibonacci</code>, um gerador dos "n" primeiros elementos da sequência de Fibonacci <ul style="list-style-type: none"> ○ Especificação para uma implementação síncrona - O circuito recebe duas entradas do seu ambiente: (1) uma entrada de 8 bits denominada <code>nro_elementos</code>, que designa quantos dos primeiros elementos da sequência de Fibonacci o circuito deve produzir antes de automaticamente parar; (2) uma entrada de controle de 1 bit denominada <code>anda_para</code>, que diz quando em '0' que o processo de geração de elementos da série deve ser/estar suspenso, e quando em '1' determina que a cada borda de subida da entrada de <code>clock</code> seja gerado um novo elemento da série, caso ainda haja valores a gerar. As entradas de controle da operação são o já mencionado <code>clock</code> e o sinal de <code>reset</code>. A única saída do circuito é denominada <code>Saida</code>, e mostra, em 8 bits, o último elemento gerado da Série de Fibonacci. A interface assim definida do circuito aparece neste link.

			<ul style="list-style-type: none"> ○ Operação resumida da versão síncrona do circuito Fibonacci: O circuito pode gerar um subconjunto de elementos da sequência de Fibonacci, limitado a sequências com até 255 elementos (pois nro_elementos é de 8 bits, o que permite gerar números entre 0 e 255 apenas). Destes, no máximo os 14 primeiros podem estar corretos (pois Saída é um sinal de apenas 8 bits, e o décimo quarto elemento da sequência de Fibonacci já é 233). Dentro destas limitações, a operação do circuito é a seguinte: (1) Tudo inicia com o sinal de reset (assíncrono) sendo ativado (reset=1), o que diz que o valor da entrada nro_elementos deve ser amostrado e guardado internamente. Como efeito secundário, Saída deve receber o valor 0; (2) Após o reset ser desativado (reset=0), a cada borda de subida do sinal clock, se o sinal externo anda_para estiver em '1' gera-se em Saída o próximo elemento da sequência (0, 1, 1, 2, 3, 5, 8, ...). Se, e enquanto anda_para estiver em '0', nada muda na saída e o processo de geração de elementos da sequência fica suspenso; (3) O processo de geração para automaticamente depois de gerar o nro_elementos-ésimo elemento da série. O processo pode se repetir com uma nova ativação do sinal de reset (passo (1) acima) ○ Questão a ser respondida: Como implementar o circuito Fibonacci? <ul style="list-style-type: none"> ▪ Princípios gerais da operação síncrona (Pressupostos: sensível apenas à borda de subida do clock e com reset assíncrono) - <ol style="list-style-type: none"> 1. O Clock é um sinal periódico com frequência e período fixos 2. Quando o clock executa uma borda de subida, nenhum sinal por ele amostrado (entradas do circuito, externas ou internas) pode estar mudando 3. O reset, sendo assíncrono, inicializa todo o circuito internamente e eventualmente parte ou todas as saídas 4. Todos os módulos internos que guardam alguma informação são controlados pelo <i>clock</i> e pelo <i>reset</i> ▪ A divisão entre informações úteis (dados) e informações de controle (internas e externas) ▪ Dividindo a implementação do circuito em Bloco de Dados e Bloco de Controle ▪ Aplicando o processo de projeto acima ao Fibonacci ▪ Arquivos de apoio ao projeto lógico do Fibonacci (Para o software Logisim) <p>* Para quem quiser usar o ambiente Xilinx Vivado, veja no Box de Material de Apoio da disciplina as instruções para baixar e instalar este software na sua versão free. Este é muito útil para exercitar as descrições executáveis desta disciplina.</p>
11	06/04	Qua	<p>Aula 2 de 7 - Unidade 02</p> <ul style="list-style-type: none"> • Tipos de Organizações Sequenciais de Processadores <ul style="list-style-type: none"> ○ Organizações monociclo versus organizações multiciclo ○ Organizações Harvard e von Neumann ○ Organizações híbridas • Uma especificação para uma organização - MIPS-S <ul style="list-style-type: none"> • Princípios básicos • Os 6 elementos definitórios da MIPS-S • O diagrama de blocos da MIPS-S • Instruções tipo R <ul style="list-style-type: none"> ○ O formato R de instruções do MIPS ○ Execução da instrução ADDU ilustrada com o Diagrama de Blocos da MIPS
12	08/04	Sex	<p>Aula 3 de 7 - Unidade 02</p> <ul style="list-style-type: none"> • Memórias e Hierarquias de Memória em Organização e Arquitetura de Processadores <ul style="list-style-type: none"> ▪ Diferentes tipos de memória usadas em processadores: registradores, caches, memória principal, etc. • Temas acessórios em Organização e Arquitetura de Computadores (outra disciplina) <ul style="list-style-type: none"> ○ Entrada e Saída em sistemas computacionais ○ Hierarquias de Memória ○ Paralelismo • O formato I de instruções do MIPS <ul style="list-style-type: none"> ○ A instrução BEQ -especificação e execução ○ Limites determinados pelo campo de offset ○ Execução e relação com o Diagrama de Blocos da MIPS-S • O formato J de instruções do MIPS e a instrução JR <ul style="list-style-type: none"> ○ Instruções J e JAL ○ A instrução JR
13	13/04	Qua	<p>Entrega das Avaliações de TP1 dos colegas</p> <p>Aula 4 de 7 – Unidade 02</p> <ul style="list-style-type: none"> • Revisão/Detailhamento da execução de instruções de salto <ul style="list-style-type: none"> ▪ Instruções LW, LBU e o Diagrama de Blocos da MIPS-S ▪ Instruções SW, SB e o Diagrama de Blocos da MIPS-S • A estrutura geral da implementação VHDL da MIPS-S <ul style="list-style-type: none"> ▪ Os módulos principais: BD, BC, registrador de 32b, ALU, banco de registradores, FSM ▪ Hierarquia da implementação ▪ Um pouco sobre o testbench ▪ Simulação da MIPS-S
14	20/04	Qua	<p>Especificação do Trabalho Prático 2 (Peso: 20% de G1)</p> <p>Aula 5 de 7 - Unidade 02</p> <ul style="list-style-type: none"> • Os módulos de multiplicação e divisão da MIPS-S • Princípios do projeto dos módulos: Hw mais simples possível <ul style="list-style-type: none"> ○ Módulos síncronos ○ Estrutura similar a algoritmos lápis e papel

			<ul style="list-style-type: none"> ○ Multiplicação - geração de produtos parciais e soma destes ○ Divisão - subtrações sucessivas do divisor do dividendo e teste ● Princípios de simulação da MIPS <ul style="list-style-type: none"> ● Testbench capaz de carregar programas e dados em memórias externas ● Preparação do arquivo de entrada (programa e dados gerados no MARS) ● Execução clock a clock de um programa via simulação ● Estrutura do Banco de Registradores - Ideia geral
15	22/04	Sex	<p>Aula 6 de 7 - Unidade 02</p> <ul style="list-style-type: none"> ● O Banco de Registradores - detalhamento do código <ul style="list-style-type: none"> ○ Comandos for-generate e if-generate ○ Instanciação dos registradores do Banco ○ Criação do decodificador - a porta de escrita do banco ○ Os multiplexadores que produzem as portas de leitura do banco ● A Unidade Lógico-Aritmética <ul style="list-style-type: none"> ○ Tabela de operações (Tabela 3 da especificação do MIPS-S) <ul style="list-style-type: none"> ■ Operações básicas - soma, subtração, and, or, xor, nor ■ Operações especiais - suporte a LUI, J/JAL etc.
16	27/04	Qua	<p>Aula 7 de 7 - Unidade 02</p> <ul style="list-style-type: none"> ● A estrutura geral da implementação da organização MIPS-S <ul style="list-style-type: none"> ○ Os 6 pares Entidade/Arquitetura - Registrador Básico, ALU, Banco de Registradores, o Bloco de Dados, o Bloco de Controle, a MIPS-S ○ O package p_MIPS_MCS ○ O Bloco de Dados <ul style="list-style-type: none"> ■ Detectores de instruções R/I/branch ■ Estágios 2/3/4/5 da organização ■ Os multiplexadores M3 a M11 ■ O comparador ■ Instanciação do multiplicador e do divisor ■ Os registradores Hi e Lo ○ O Bloco de Controle ○ Microinstruções, microoperações ○ O primeiro estágio da organização - PC/IR/NPC ○ O Testbench da MIPS-S ● Estrutura geral do VHDL da MIPS-S - O Bloco de Controle <ul style="list-style-type: none"> ○ O PC, seu incrementador e o NPC ○ A decodificação de instruções ○ A máquina de estados de controle ○ O registrador IR <p>Entrega dos Resultados do TP1</p>
17	29/04	Sex	<p>Prova P1 - Unidades 01 e 02 (peso: 20% de G1)</p>
18	04/05	Qua	<p>Unidade 03 - Organização de Processadores: Paralelismo Básico e Aceleradores</p> <ul style="list-style-type: none"> ● Introdução - Pipelines e Aceleradores Aritméticos ● Pipelines - Aula 1 de 3
19	06/05	Sex	<ul style="list-style-type: none"> ● Pipelines e Hazards - Aula 2 de 3 <ul style="list-style-type: none"> ○ Ideia fundamental - paralelismo temporal ○ Problemas para pipelines em sistemas computacionais e bolhas ○ Exercícios
20	11/05	Qua	<ul style="list-style-type: none"> ● Pipelines e Hazards - Aula 3 de 3 <ul style="list-style-type: none"> ○ Hazards - definição e classificação ○ Soluções para Hazards de dado, de controle ○ Otimizações em software - o papel do compilador ○ Exercícios
21	13/05	Sex	<p>Data de Entrega do TP2</p> <ul style="list-style-type: none"> ● Aceleradores de Hardware - Aula 1 de 4 <ul style="list-style-type: none"> ○ Aritmética Computacional <ul style="list-style-type: none"> ■ Aritmética com Naturais (N) ■ Aritmética com Inteiros (Z) ■ Aritmética com Racionais (Q) - o padrão IEEE-754
22	18/05	Qua	<ul style="list-style-type: none"> ● Exercícios de Pipeline - Lista com soluções ● Aceleradores de Hardware - Aula 2 de 4 <ul style="list-style-type: none"> ○ Aritmética Computacional <ul style="list-style-type: none"> ■ Aritmética com Naturais (N) ■ Aritmética com Inteiros (Z) ■ Aritmética com Racionais (Q) - o padrão IEEE-754 <p>Entrega dos Resultados da P1</p>

23	20/05	Sex	<ul style="list-style-type: none"> ● Aceleradores de Hardware - Aula 3 de 4 <ul style="list-style-type: none"> ○ Aritmética Computacional <ul style="list-style-type: none"> ■ Aritmética com Naturais (N) ■ Aritmética com Inteiros (Z) ■ Aritmética com Racionais (Q) - o padrão IEEE-754
24	25/05	Qua	<ul style="list-style-type: none"> ● Aceleradores de Hardware - Aula 4 de 4 <ul style="list-style-type: none"> ○ Aritmética Computacional <ul style="list-style-type: none"> ■ Aritmética com Naturais (N) ■ Aritmética com Inteiros (Z) ■ Aritmética com Racionais (Q) - o padrão IEEE-754
25	27/05	Sex	<p>Unidade 04 - Arquitetura de Processadores: Tópicos Fundamentais Adicionais</p> <ul style="list-style-type: none"> ● Entrada e Saída em Processadores - Aula 1 de 4 <p>Entrega dos Resultados do TP2</p>
26	1ro/06	Qua	<ul style="list-style-type: none"> ● Entrada e Saída em Processadores - Aula 2 de 4
27	03/06	Sex	<ul style="list-style-type: none"> ● Entrada e Saída em Processadores - Aula 3 de 4
28	08/06	Qua	<ul style="list-style-type: none"> ● Entrada e Saída em Processadores - Aula 3 de 4
29	10/06	Sex	<ul style="list-style-type: none"> ● Hierarquia de Memória em Processadores - Aula 1 de 3
30	15/06	Qua	<ul style="list-style-type: none"> ● Hierarquia de Memória em Processadores - Aula 2 de 3
31	17/06	Sex	<ul style="list-style-type: none"> ● Hierarquia de Memória em Processadores - Aula 3 de 3
32	22/06	Qua	Prova P2 - Unidades 03 e 04 (peso: 20% de G1)
33	24/06	Sex	Aula de Dúvidas: PS e G2
34	29/06	Qua	Prova PS - Inclui Conteúdos de todas as Unidades
35	1ro/07	Sex	Aula de Dúvidas: G2
--	06/07	Qua	Prova de G2 - Inclui Conteúdos de todas as Unidades