

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES II

EDMILSON LINO CORDEIRO
DAVI VENTURA CARDOSO PERDIGÃO
JOÃO PAULO FERNANDES ROCHA



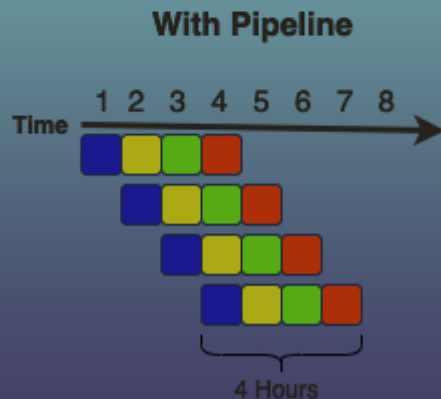
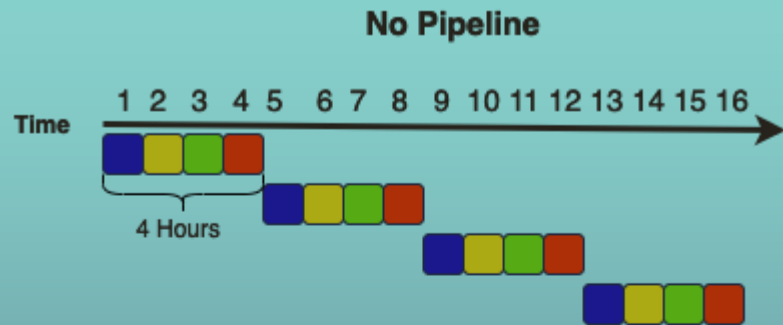


Estrutura e Função do Processador

Pipeline

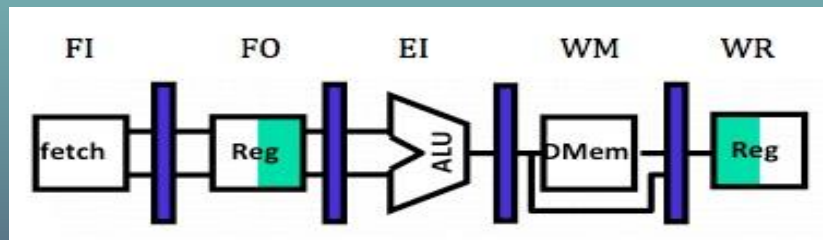
DEFINIÇÃO

Pipeline, muitas vezes traduzido para português como paralelismo, é uma técnica que permite os processadores executarem tarefas diferentes ao mesmo tempo. Essas tarefas são colocadas em uma fila de memória dentro do processador (CPU) onde aguardam o momento de serem executadas: assim que uma instrução termina o primeiro estágio e parte para o segundo, a próxima instrução já ocupa o primeiro estágio. Essa técnica aumenta o desempenho do processador e reduz o tempo de execução global de tarefas.



Arquitetura Básica Pipeline em um Computador

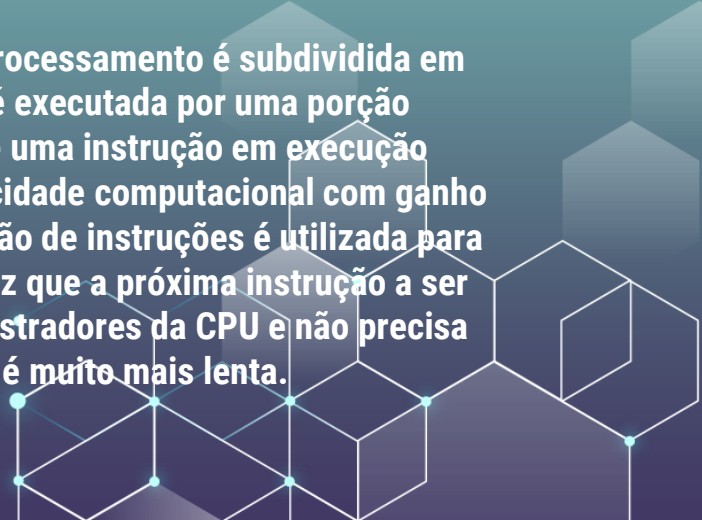
- A primeira mudança necessária é a separação da memória em duas partes independentes (ou duas memórias). Essa mudança vai contra o que foi projetado na Arquitetura de von Neumann, e considerado um grande avanço. Ela foi batizada de Arquitetura Harvard.
- Outra mudança importante foi a adição de memórias intermediárias entre cada etapa. Outra mudança importante foi a adição de memórias intermediárias entre cada etapa.



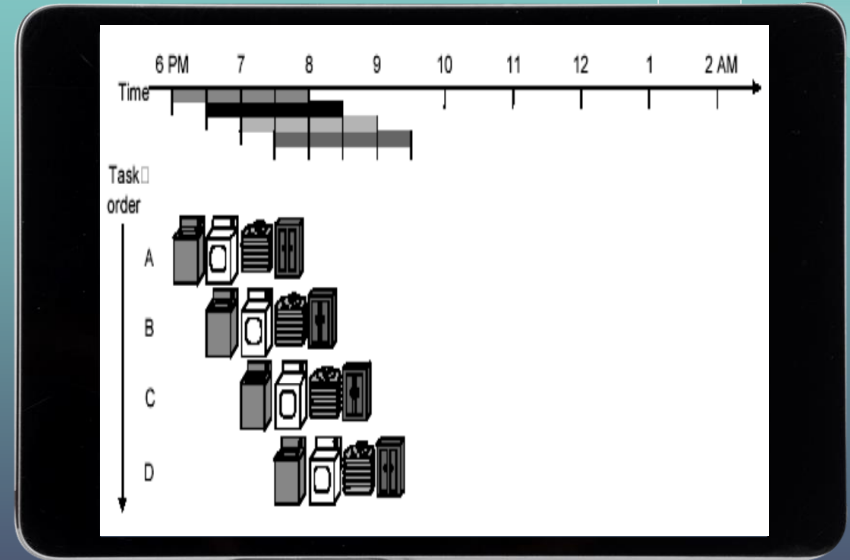
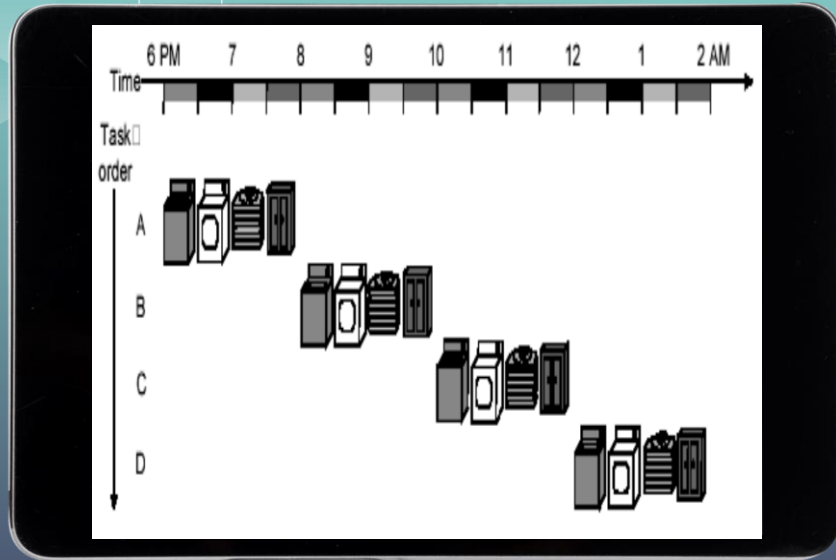


VISÃO GERAL PIPELINE

Em resumo, é o processo pelo qual uma instrução de processamento é subdividida em etapas, uma vez que cada uma destas etapas é executada por uma porção especializada da CPU, podendo colocar mais de uma instrução em execução simultânea. Isto traz um uso mais racional da capacidade computacional com ganho substancial de velocidade. A técnica de segmentação de instruções é utilizada para acelerar a velocidade de operação da CPU, uma vez que a próxima instrução a ser executada está normalmente armazenada nos registradores da CPU e não precisa ser buscada da memória principal que é muito mais lenta.



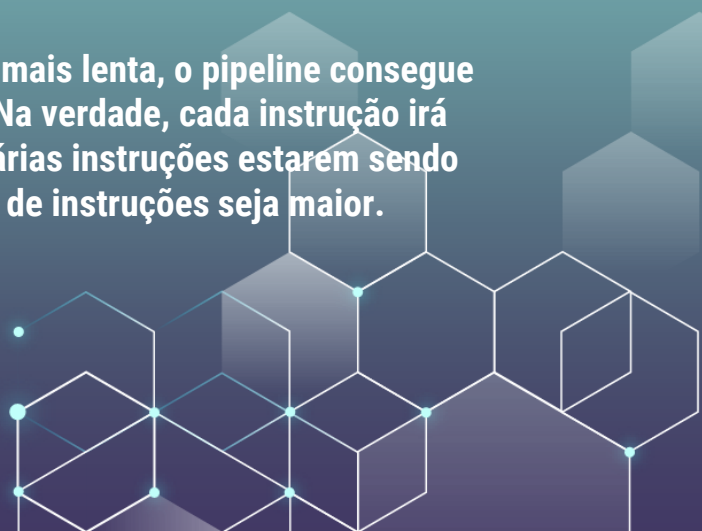
ANALOGIA “LAVANDERIA”





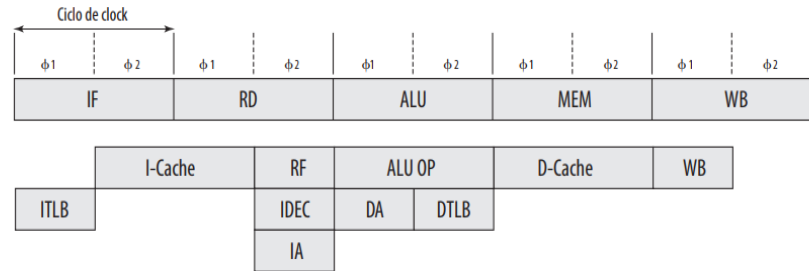
PRINCÍPIO BÁSICO

Por mais que tenhamos um clock baseado na instrução mais lenta, o pipeline consegue ter um tempo de execução por instrução menor. Na verdade, cada instrução irá demorar o tempo de clock, no entanto o fato de várias instruções estarem sendo executadas em paralelo faz com que a vazão de instruções seja maior.



INSTRUÇÕES DO MIPS

Como já foi estudado, os processadores são componentes desenvolvidos metodicamente e seguem sempre um padrão de funcionamento e realização das tarefas, portanto na arquitetura MIPS tal fato não seria diferente. Em relação ao Pipeline, no MIPS, o processo de leitura, execução e atualização das instruções é dividido em cinco etapas:



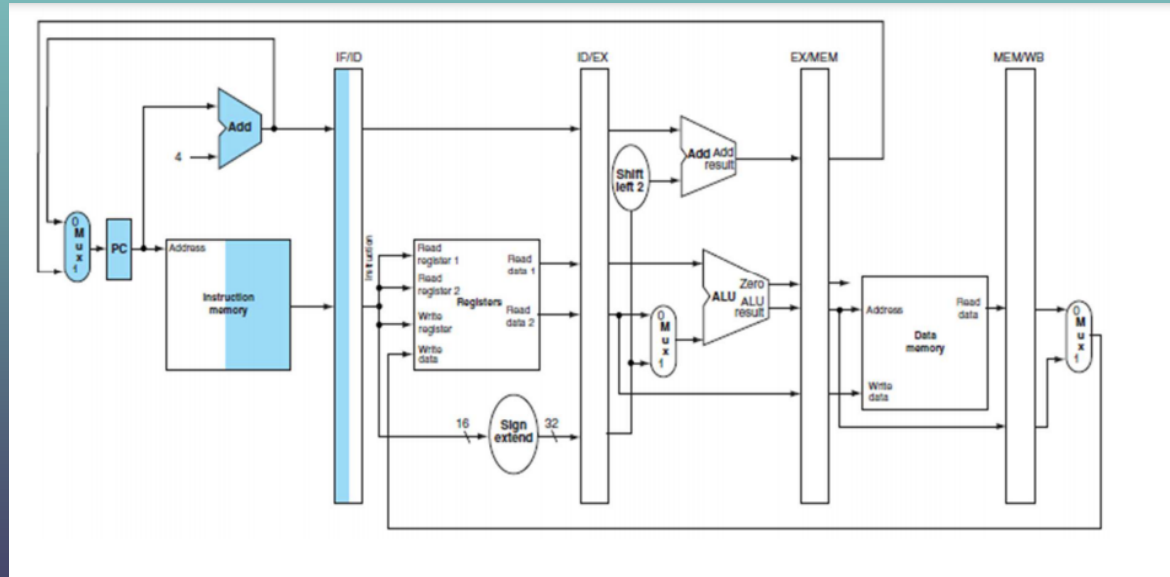
IF	=	busca da instrução
RD	=	leitura
MEM	=	acesso à memória
WB	=	atualizar
I-Cache	=	acesso à cache de instruções
RF	=	busca do operando do registrador
D-Cache	=	acesso à cache de dados
ITLB	=	tradução endereço da instrução
IDEC	=	decodificação da instrução
IA	=	calcular endereço da instrução
DA	=	calcular endereço virtual de dados
DTLB	=	traduzir endereço de dados
TC	=	verificar rótulo de cache de dados

AS 5 ETAPAS:



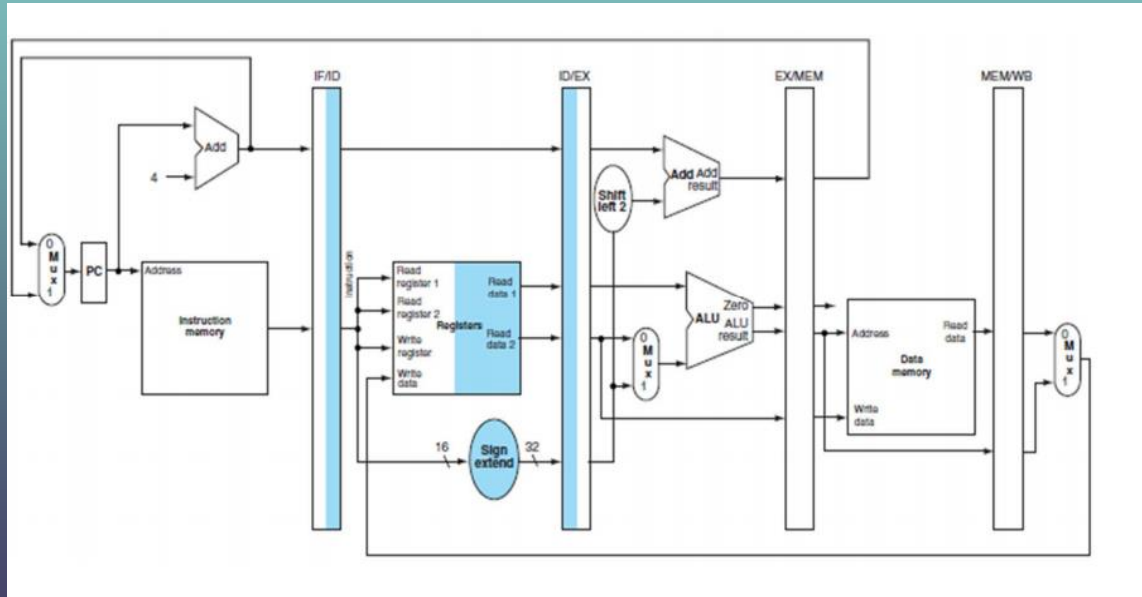
Iª ETAPA:

IF



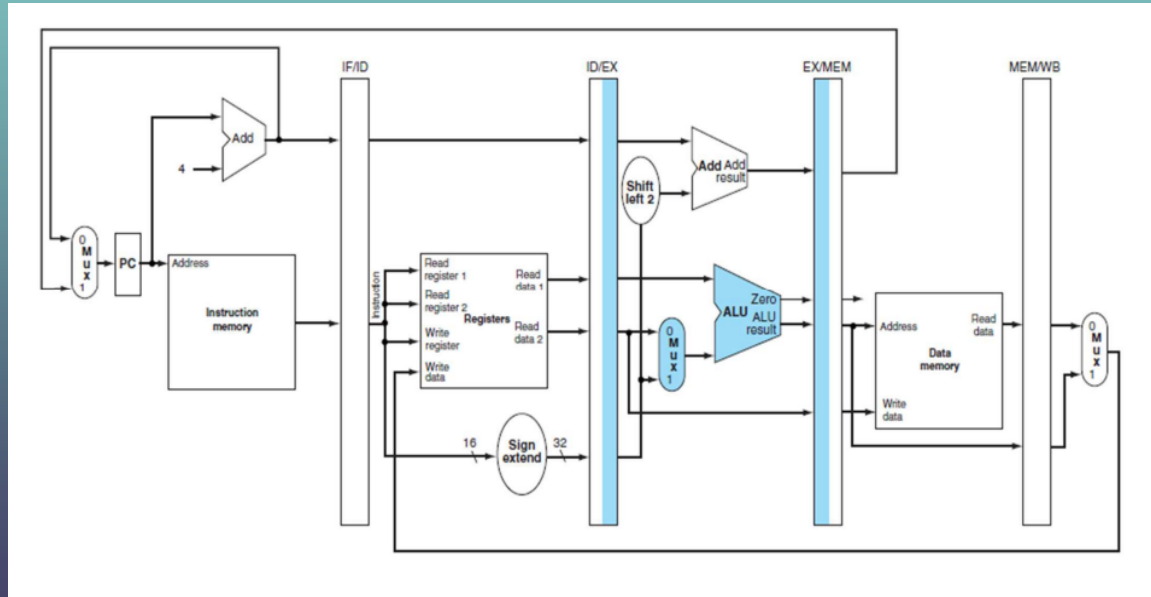
2ª ETAPA:

RD



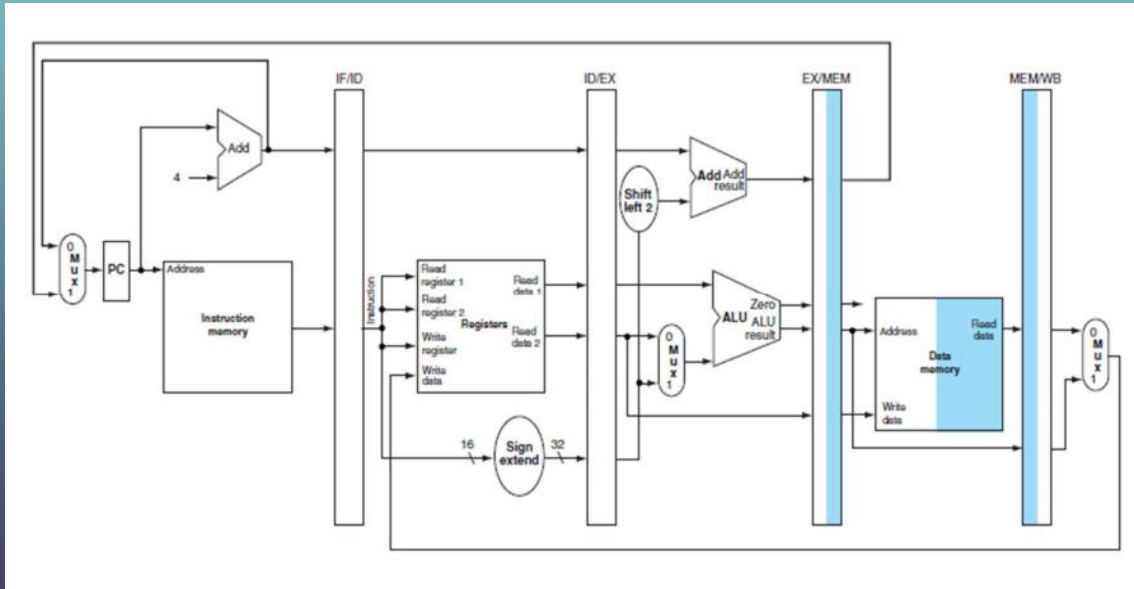
3ª ETAPA:

ALU



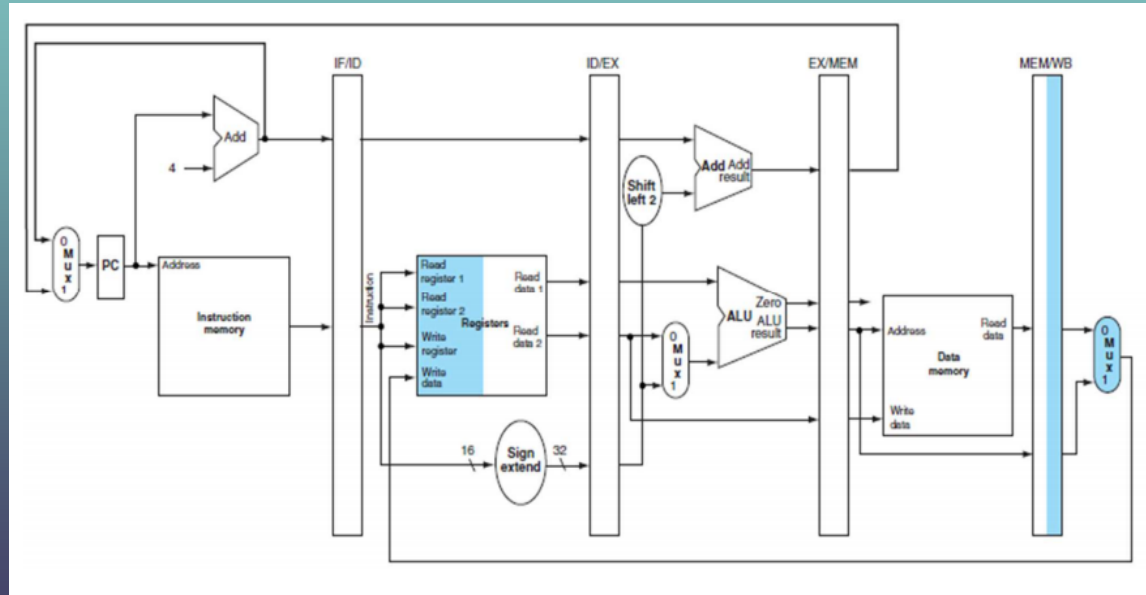
4ª ETAPA:

mem



5ª ETAPA:

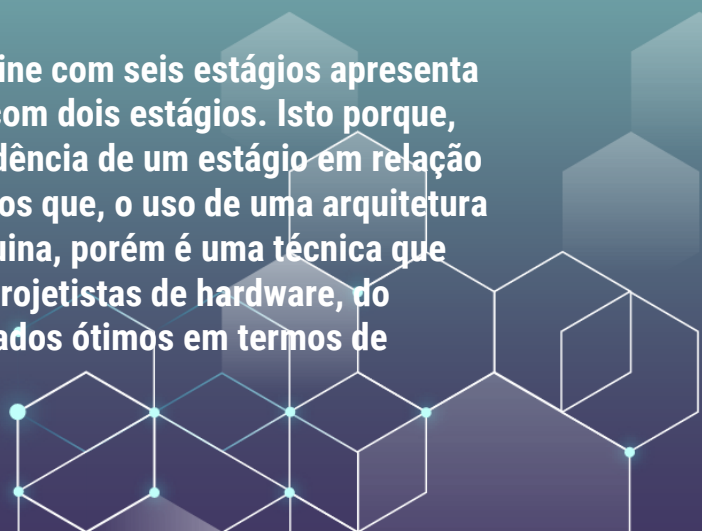
WB





CONCLUSÃO

Apesar do ganho de desempenho, uma arquitetura pipeline com seis estágios apresenta problemas que não ocorrem em uma arquitetura com dois estágios. Isto porque, quanto maior o número de estágios, maior a dependência de um estágio em relação ao outro. Isso é apenas um exemplo para concluirmos que, o uso de uma arquitetura pipeline traz ganhos de performance para a máquina, porém é uma técnica que requer um projeto cuidadoso (por parte dos projetistas de hardware, do processador) para que possa alcançar resultados ótimos em termos de desempenho.



REFERÊNCIAS (IMAGENS)

1. **Figura 1 - Demonstrativo do tempo de execução sem Pipeline e com Pipeline. Fonte: PANTUZA, Gustavo. 2020.**
2. **Figura 2 - Processador adaptado para trabalhar com Pipeline de cinco estágios. Fonte: BRITO, Alisson. Introdução a Arquitetura de Computadores.**
3. **Figura 3 - Analogia com lavanderia, com e sem Pipeline. Fonte: SILVA, Gabriel. Arquitetura de Computadores II.**
4. **Figura 4 - Divisão em cinco etapas. Fonte: STALLINGS, William. Pag. 417. 2009.**
5. **Figura 5 - Exemplo Etapa IF. Fonte: UFF (Universidade Federal Fluminense), Cap. 5. 2017**
6. **Figura 6 - Exemplo Etapa RD. Fonte: UFF (Universidade Federal Fluminense), Cap. 5. 2017**
7. **Figura 7 - Exemplo Etapa ALU. Fonte: UFF (Universidade Federal Fluminense), Cap. 5. 2017**
8. **Figura 8 - Exemplo Etapa MEM. Fonte: UFF (Universidade Federal Fluminense), Cap. 5. 2017**
9. **Figura 9 - Exemplo Etapa WB. Fonte: UFF (Universidade Federal Fluminense), Cap. 5. 2017**

REFERÊNCIAS BIBLIOGRÁFICAS

1. PANTUZA, Gustavo. Organização e Arquitetura de Computadores – Pipeline em Processadores. Disponível em: <<https://blog.pantuza.com/artigos/organizacao-e-arquitetura-de-computadores-pipeline-em-processadores>>. Acesso em: 12 de março de 2021.
2. SÉRGIO, Luiz. Arquitetura e Organização de Computadores. Disponível em: <https://educapes.capes.gov.br/bitstream/capes/206151/2/apostila%20de%20AOC_Luiz%20S%C3%A9rgio.pdf>. Acesso em: 01 de março de 2021.
3. SILVA, Gabriel. Arquitetura de Computadores II – Pipeline. Disponível em: <<https://dcc.ufrj.br/~gabriel/arqcomp2/Pipeline.pdf>>. Acesso em: 11 de março de 2021.
4. STALLINGS, William. Arquitetura e Organização de Computadores. Tradução da 8ª edição. Editora Prentice Hall Brasil, 2002.
5. TANENBAUM, Andrew S. Organização Estruturada de Computadores. Tradução da 5ª edição. Editora Prentice Hall Brasil, 2007.
6. UFF - Universidade Federal Fluminense. Capítulo 5 Pipeline. Disponível em: <<http://www.professores.uff.br/lbertini/wp-content/uploads/sites/108/2017/08/Cap-5-Pipeline.pdf>>. Acesso em: 10 de março de 2021.
7. VASCONCELOS, Laércio. Hardware Total. 1ª Edição. Editora Makron Books, 2002