

Atividade – Aula 22/04

ALUNA: KAREN GIOVANNA
PRAZO DE ENTREGA: 26/04/2021

[Questão – 01] Quais as diferenças de um processador multi ciclo em relação a um uniciclo?

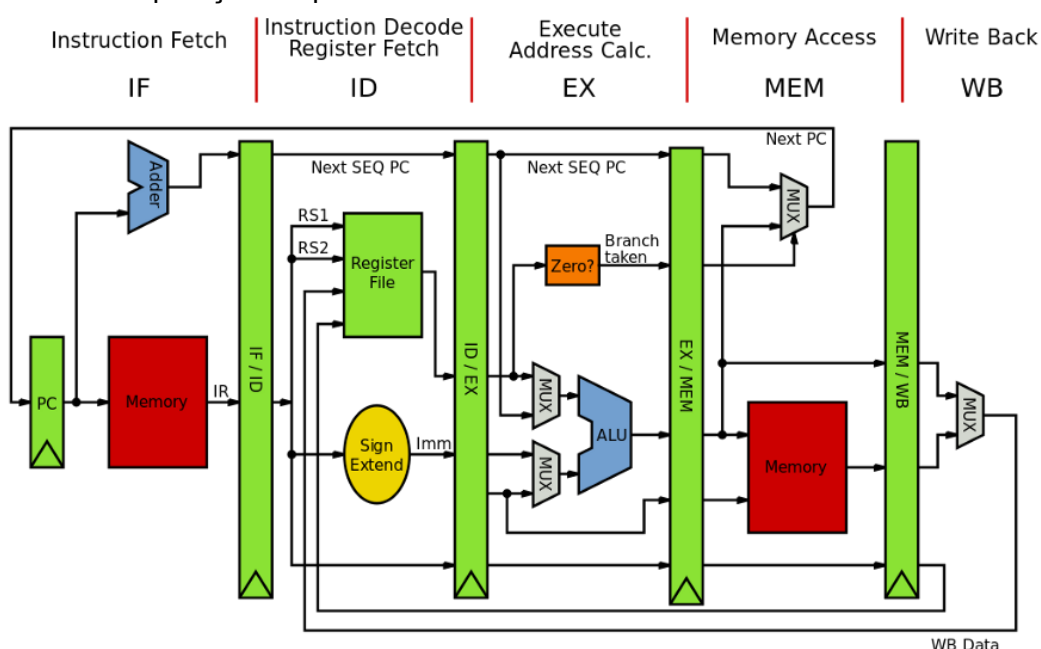
No processador **monociclo** o ciclo de clock tem o mesmo tamanho para todas as instruções implementadas; Este ciclo é definido pela instrução de maior duração (lw – load word); Cada unidade funcional só pode ser usada 1 vez por ciclo, com isso algumas unidades funcionais são duplicadas, aumentando consideravelmente o custo do hardware.

No processador **multiciclo** a execução de cada instrução é quebrada em passos; • Cada passo gasta 1 ciclo de clock; • Com isso, uma unidade funcional pode ser utilizada mais de 1 vez por instrução, desde que em diferentes ciclos.

Fonte: <http://www.ic.uff.br/~bazilio/cursos/arqcomp/cpu-multiciclo.pdf>

[Questão – 02] Quais as modificações necessárias em um processador multi ciclo simples para que se introduza a função de pipeline?

A aplicação do conceito de pipeline na arquitetura de um processador exige, primeiramente, a divisão do processo em “tarefas” menores com suas devidas unidades de processamento, que ocorre como exemplifica a Figura 3. Essa divisão ocorre em arquiteturas multiciclos, as quais podem apresentar números diferentes de ciclos do apresentado pela Figura 3, que apresenta uma separação simples e didática.



A ilustração apresenta cinco ciclos — ou tarefas — cada um com suas unidades funcionais: busca de instruções (IF), decodificação de instruções (ID), execução (EX), acesso de memória (MEM) e escrita no banco de registradores (WB).

A segunda exigência para a implementação do pipeline é a inclusão de recursos que viabilizem a realização desses ciclos de forma simultânea: registradores intermediários.

A ilustração desses registradores na Figura 3 é feita pelas barras verticais entre cada estágio (ciclo) do pipeline. Esses representam o conjunto de registradores necessários para armazenar o resultado de cada estágio, sendo que cada conjunto recebe o rótulo de acordo com os estágios que está conectando: IF/ID, ID/EX, EX/MEM e MEM/WB.

A principal motivação para a inclusão desses registradores é a necessidade de armazenar o resultado de um estágio antes que o próximo acabe. Isso pode ser comparado a necessidade de um local para deixar as roupas lavadas do grupo B enquanto as roupas do grupo A utilizam a secadora.

Fonte: <https://medium.com/computer-architecture/pipeline-e-registradores-intermedi%C3%A1rios-defini%C3%A7%C3%A3o-e-conceitos-c0d1d9797ab>

[Questão – 03] Descreva os seguintes conflitos em pipeline: Estrutural; Dados; e Controle.

Conflitos Estruturais: O Hardware não suporta uma determinada combinação de instruções. Acontece quando algum componente é necessário em mais de 1 estágio. Exemplo: Se a memória de instruções é a mesma de dados.

Conflitos de dados: Acontece quando uma instrução depende do resultado de uma instrução anterior. Exemplo: Uma instrução de sub que depende de um add anterior. O resultado só será escrito no último estágio do add.

Conflitos de controle: Surgem quando são executadas instruções de saltos que alteram o PC. Exemplo: A próxima instrução depende de um branch ser tomado ou não.

[Questão – 04] Dado o seguinte código, apresente quais conflitos podem ocorrer em um processador com pipeline.

ADD R1, R2, R3
SUB R4, R1, R5
AND R6, R1, R7
BEQ R2, R3, SET
ADD R2, R2, R3
SET:

SUB R4, R1, R5

Instruções	Ciclos de Clock									
	1	2	3	4	5	6	7	8	9	10
add r1, r2, r3	IF	ID	EX	WB						
Sub r4, r1, r5		IF	ID	EX	WB					
And r6, r1, r7			IF	ID	EX	WB				
BEQ r2, r3, set				IF	ID	EX	WB			
Add r2, r2, r3					IF	ID	EX	WB		
SET:										
SUB r4, r1, r5						IF	ID	EX	WB	

Busca de instruções (IF),
 decodificação de instruções (ID),
 execução (EX),
 acesso de memória (MEM) e
 escrita no banco de registradores (WB).

- 1) Conflito de dados: ocorre na instrução sub pois ela depende do resultado da instrução add anterior;
- 2) Conflito de controle: Instrução de salto BEQ insere o endereço da instrução set na unidade funcional Program Counter, embora haja a instrução add na espera