

Arquitetura e Organização de Computadores II

Pipeline

Nilton Luiz Queiroz Junior

Ciclo de instruções

Ciclo de instruções

- Uma instrução pode ser composta por várias etapas;
- A quantidade de etapas e o que cada uma faz depende da arquitetura;
 - A arquitetura MIPS por exemplo é dividido em 5 etapas:
 - Busca de Instrução;
 - Decodificação da instrução;
 - Execução;
 - Acesso a memória;
 - Escrita de volta aos registradores;



Ciclo de Instruções

- Para execução da instrução é necessário apontar para ela;
- Assim faz-se o Program Counter (PC) armazenar o endereço de memória no qual a instrução está;
 - Antes de cada instrução é feita uma operação de soma;
 - Essa soma irá resultar na próxima instrução sequencial a ser executada;



Ciclo de instrução

- A decodificação de instrução e busca por operandos requer um banco de registradores, porém, o que vai ser utilizado depende da instrução;
 - Algumas podem utilizar três registradores, outras apenas um, etc...



Ciclo de Instrução

- Por exemplo, na arquitetura MIPS:
 - Instruções Lógicas e aritméticas:
 - Lêem até dois registradores e escrevem em um, usando assim três registradores;
 - Instruções de leitura e escrita:
 - Lêem a memória e um registrador;
 - Instruções de desvio condicional:
 - Lêem até dois registradores;
- Essa etapa também é responsável pela extensão do número de 16 bits para algumas operações;
 - Não é possível operar dois valores de tamanhos diferentes

Ciclo de instrução

- Na arquitetura MIPS a execução da instrução é feita por ULAs, onde se usam unidades de controle e multiplexadores para sua ativação.
 - Cada instrução faz uso de diferentes componentes dentro do processador, por exemplo:
 - Instruções lógicas e aritméticas usam apenas uma ULA e não acessam a memória
 - Instruções de desvio usam uma ULA para comparar os operandos e outra para somar o endereço da próxima instrução ao deslocamento
 - Caso o desvio seja tomado esse é o endereço usado, caso contrario ele é descartado;

Ciclo de instrução

- O acesso a memória na arquitetura MIPS é feito somente por leituras e escritas;
 - Leituras buscam valores na memória de acordo com o endereço especificado e os escrevem em um registrador;
 - Escritas armazenam valores dos registradores na memória;

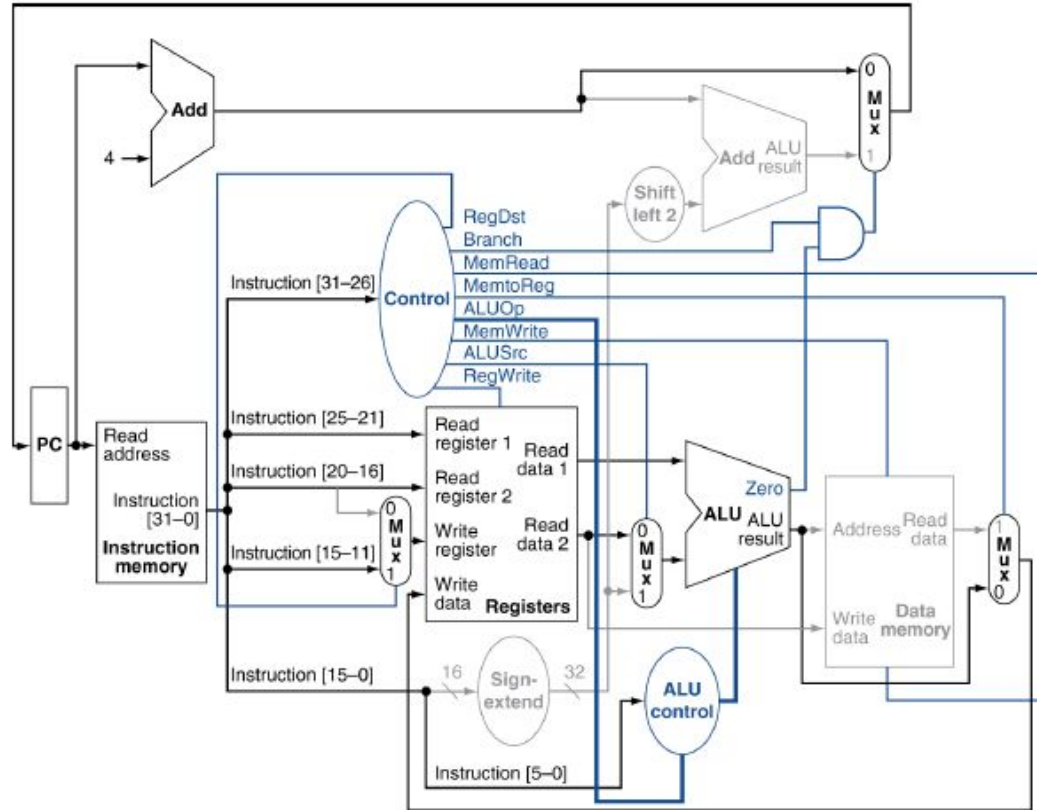


Ciclo de Instrução

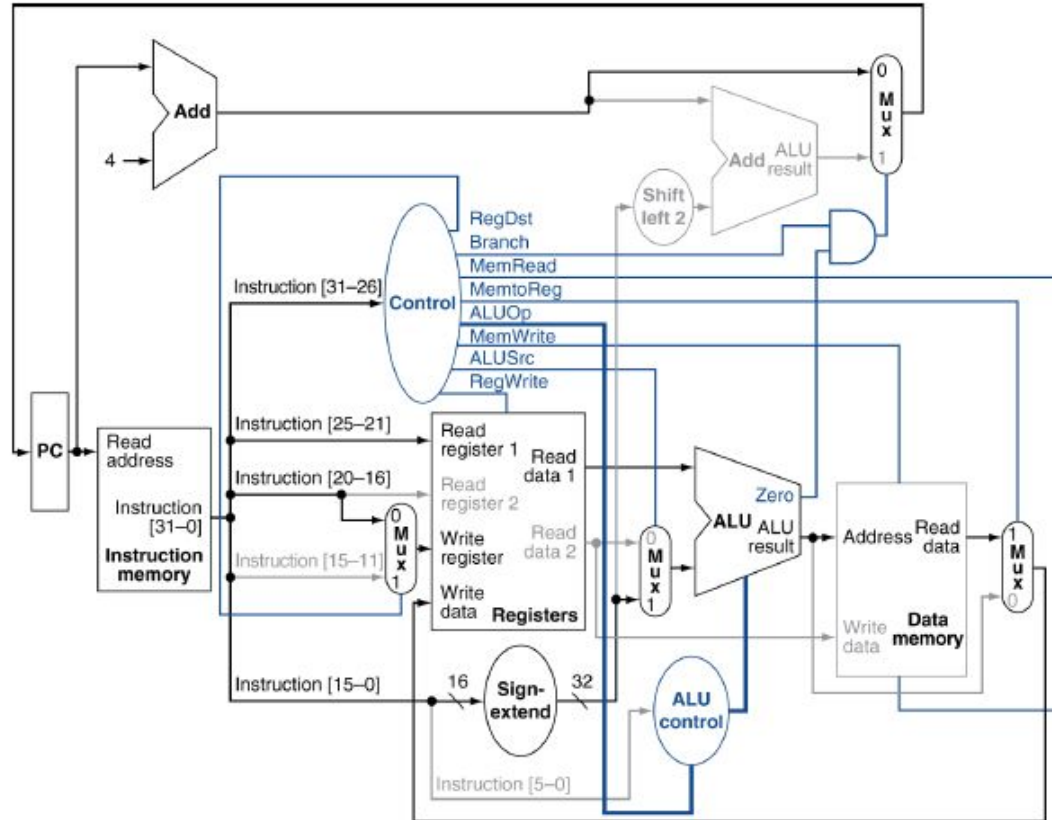
- A escrita de volta aos registradores de uso geral nem sempre é feita;
 - Instruções de desvio por exemplo escrevem somente no PC;
 - Instruções de armazenamento na memória não escrevem o dado no banco de registradores;



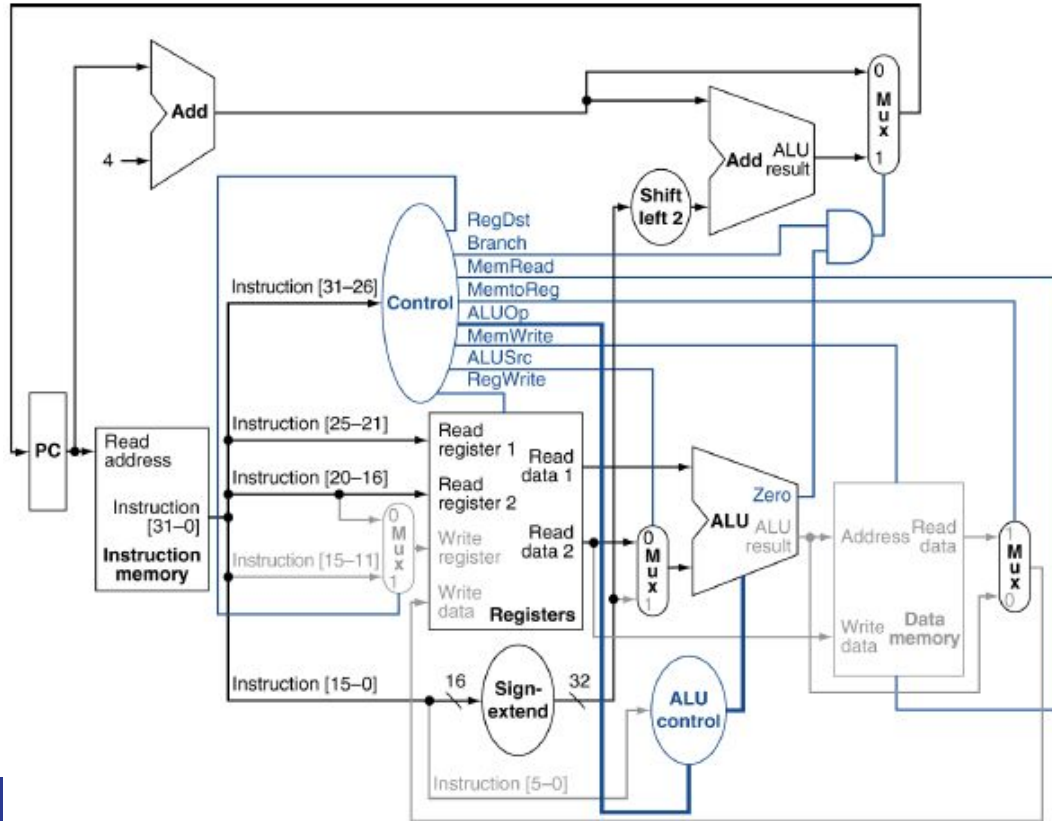
Ciclo para instruções lógicas e aritméticas



Ciclo para instruções de leitura



Ciclo para instruções de desvio



Pipeline

Visão geral de Pipeline

- O que é Pipelining?
 - Técnica para paralelismo em nível de instrução;
 - Estágios da execução ocorrem simultaneamente;
 - Transparente a nível de software;
- Ocorre também em processos fora da computação:
 - Lavar roupas;
 - Fabricar carros;



Visão geral de Pipeline

- Utiliza os recursos existentes no processador de maneira mais eficiente;
- Não reduz o tempo da execução de uma instrução;
- Reduz o intervalo de “saída” entre as instruções;
- Em condições ideais:
 - Pipeline de N estágios produz valor próximo a N vezes a execução sequencial;



Visão geral de Pipeline

- Pipelines ideais requerem que:
 - Todos objetos passem pelos mesmos estágios;
 - Não exista recursos compartilhados entre estágios;
 - Estágios tenham mesmo tempo;
 - Estágios não necessitem esperar por etapas ainda não concluídas;
- Quase impossíveis em arquiteturas reais;



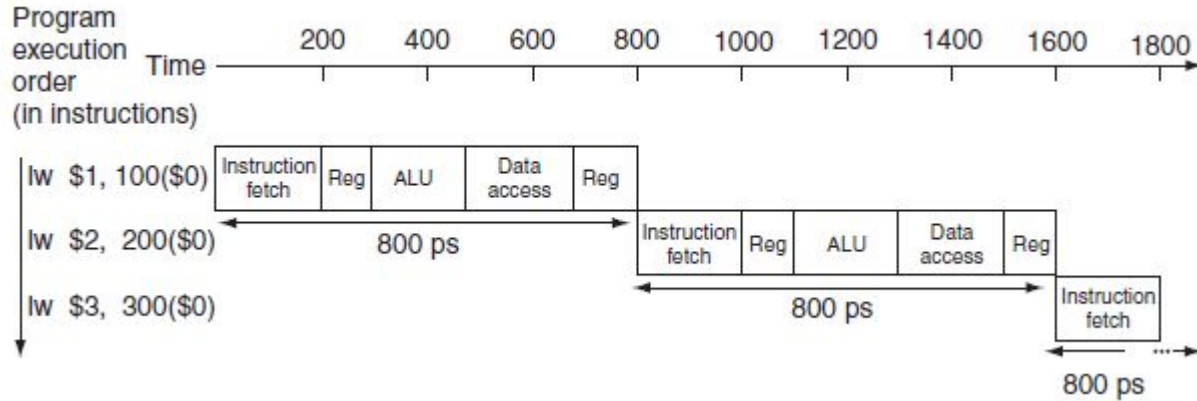
Pipeline na arquitetura MIPS

- O Pipeline da arquitetura MIPS tem 5 estágios;
 - Os 5 estágios normalmente exigidos nas instruções;



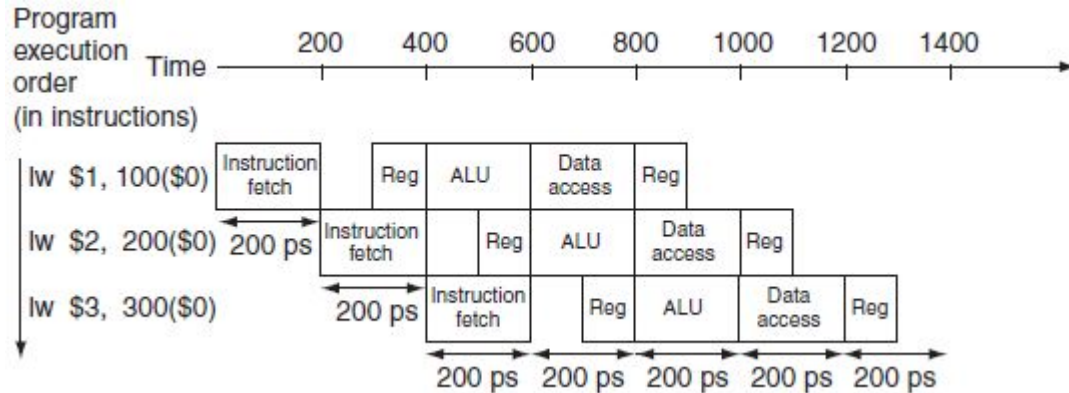
Pipelining na arquitetura MIPS

- Execução de 3 instruções load sem pipelining



Pipelining na arquitetura MIPS

- Execução de 3 instruções load com pipelining



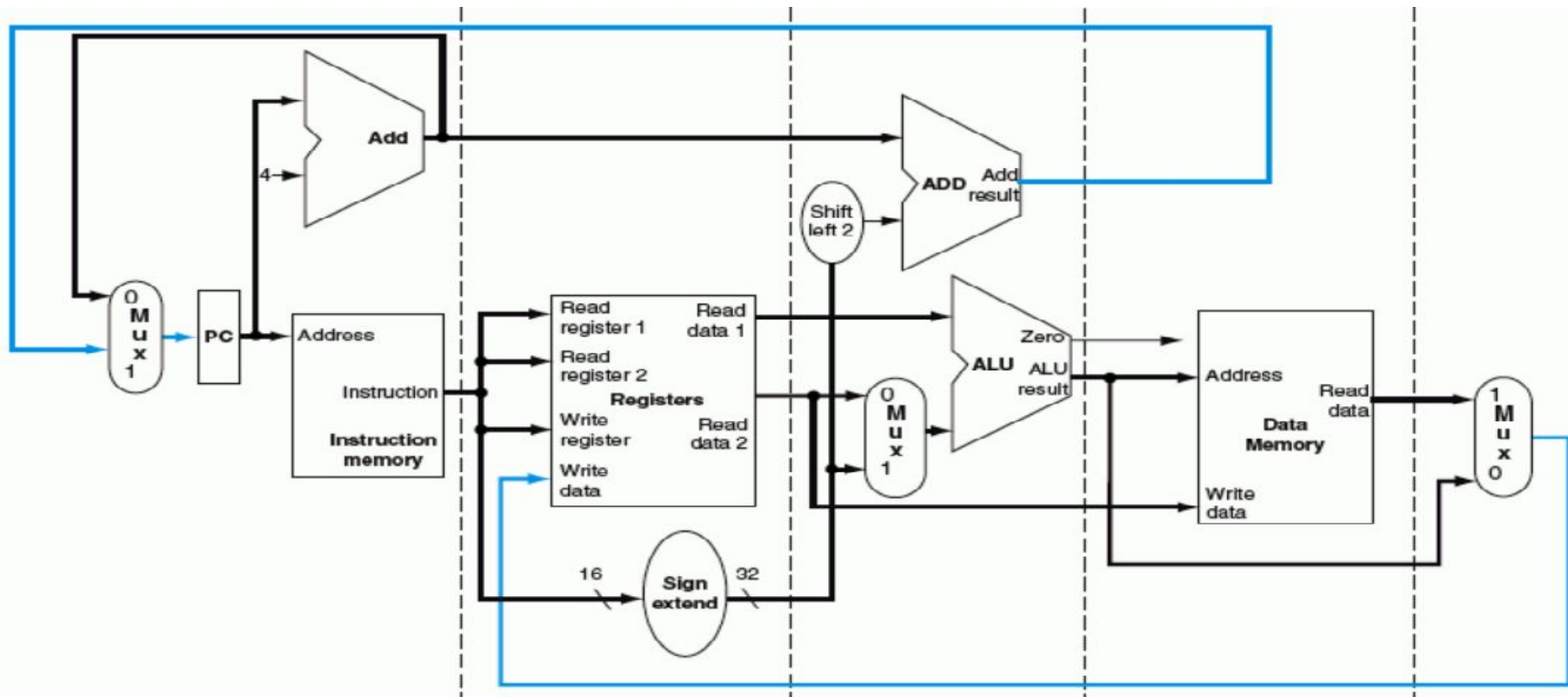
Obs: note que algumas etapas de uma instrução são mais rápidas

Caminho de dados

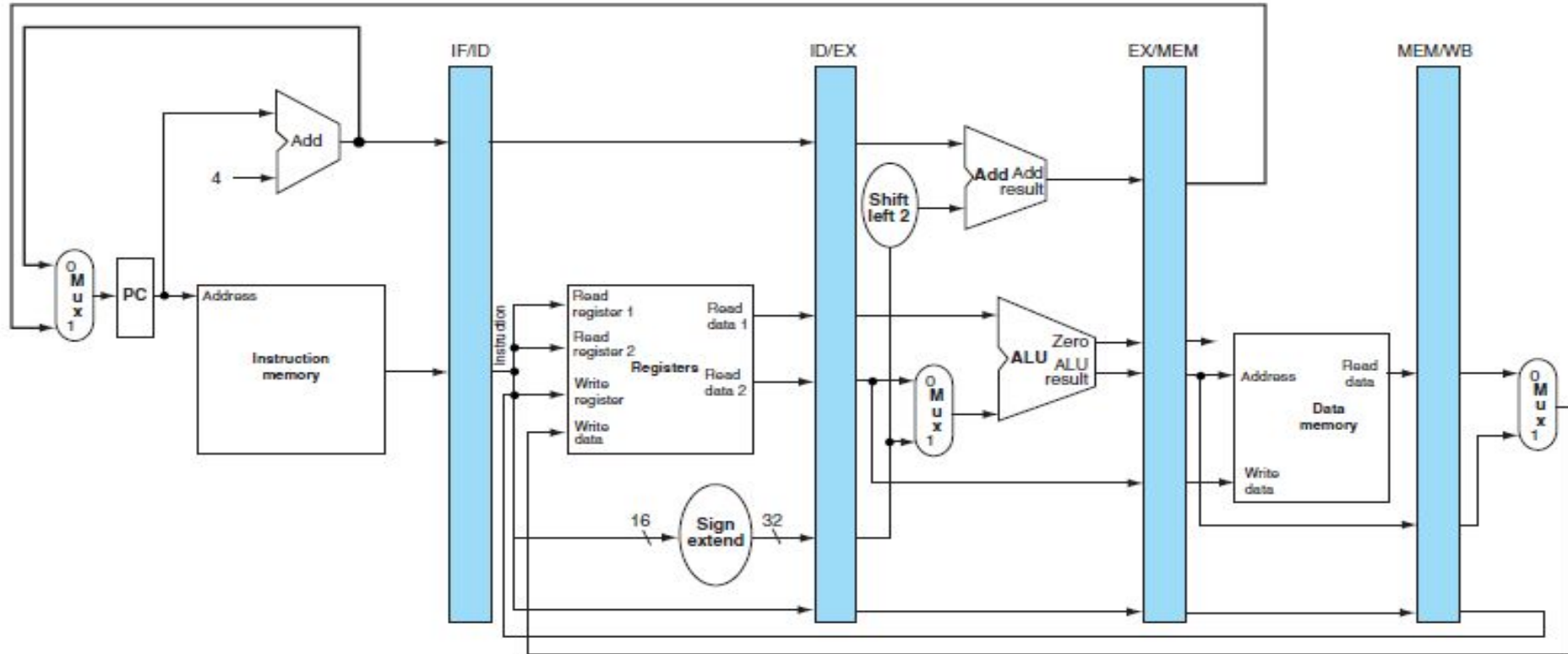
- O caminho de dados de um pipeline requer algumas alterações:
 - São necessários bancos de registradores;
 - Esses são chamados de registradores de pipeline;
 - Atualizam a cada ciclo;
 - O registrador de escrita na operação de load deve “trafegar” junto com o fluxo da instrução;
 - Se isso não ocorrer, o registrador errado será atualizado;



Caminho de dados - execução sequencial



Caminho de dados - Pipeline



Controle

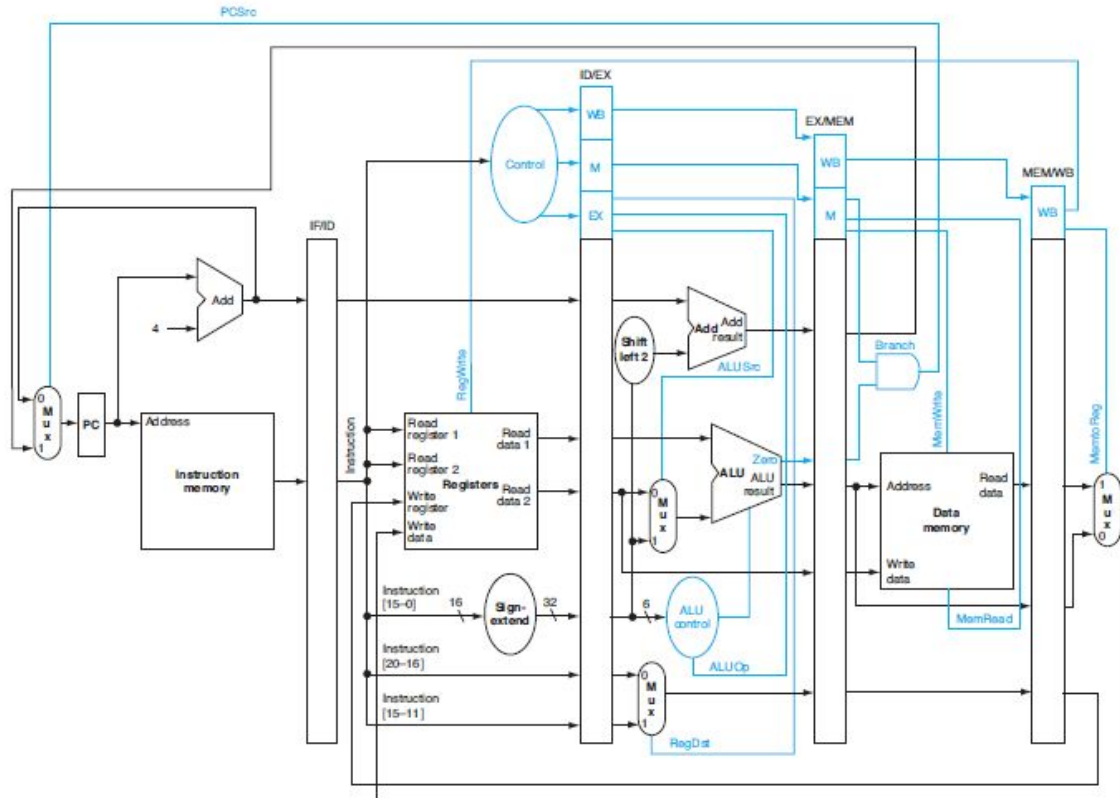
- Existirão diversas instruções ao mesmo tempo no processador;
- Nem todos sinais de controle são usados na execução;
 - Muitas vezes é necessário saber se uma instrução irá escrever ou ler da memória;
 - Em caso de leitura, em qual registrador será escrito;
- Apenas um OP-CODE não consegue armazenar todos sinais de controle das instruções que estão em execução;
 - Note que as duas primeiras etapas não precisam de seus sinais armazenados;

Controle

- É necessário um sinal de controle para cada estágio do pipeline;
 - Busca e decodificação de instruções: Sinais sempre ativos;
 - Execução: Sinais selecionam registrador destino, operação ou extensão;
 - Acesso a memória: Linhas de controle para desvio, leitura e escrita;
 - Escrita do resultado: Linha de controle decide se o valor escrito no registrador vem da ALU ou da memória;



Controle



Representação gráfica de Pipeline

- É possível representar os pipelines graficamente de diversas maneiras, entre elas tem-se:
 - Tradicional:
 - Instruções x Tempo;
 - Recursos físicos;



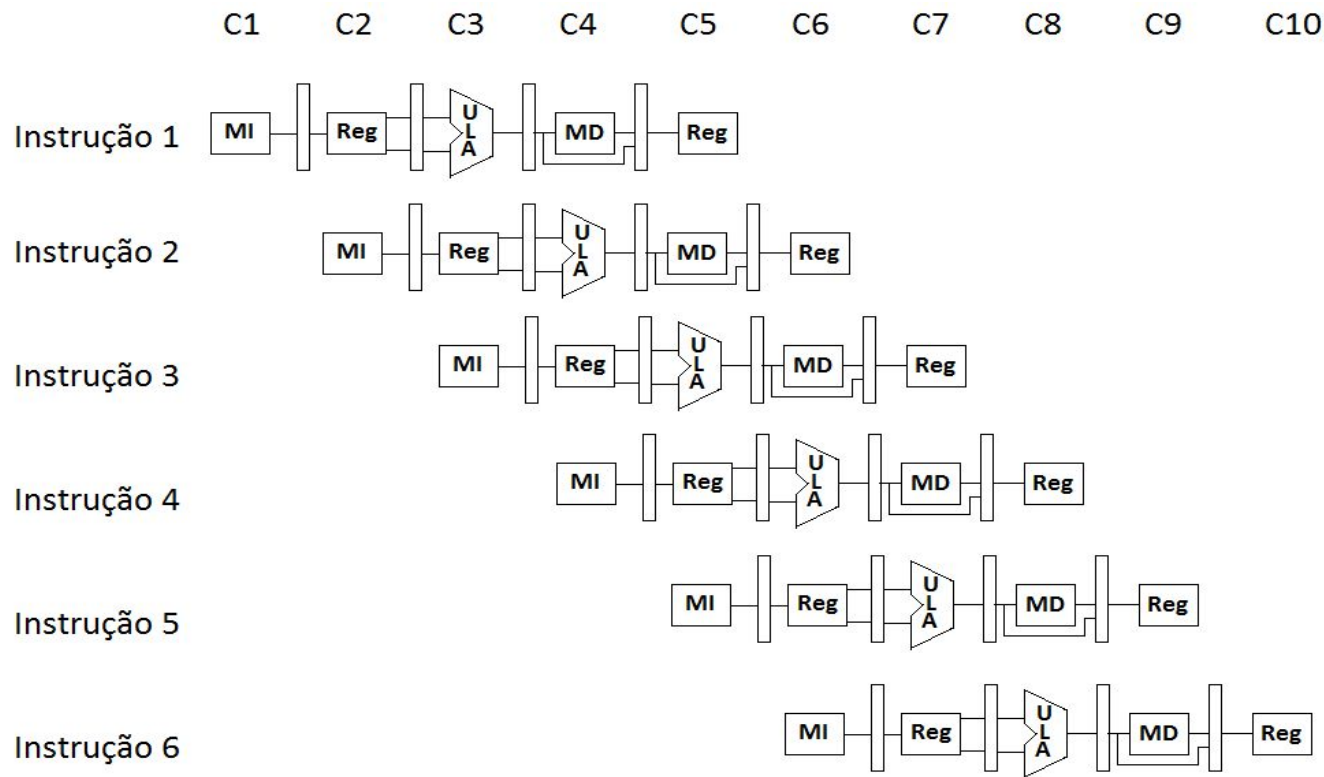
Representação gráfica de Pipeline

- Estágios

	ciclo 1	ciclo 2	ciclo 3	ciclo 4	ciclo 5	ciclo 6	ciclo 7	ciclo 8	ciclo 9	ciclo 10
Instrução 1	Busca de Instrução	Decodificação de instrução	Execução	Acesso aos dados	Escrita do resultado					
Instrução 2		Busca de Instrução	Decodificação de instrução	Execução	Acesso aos dados	Escrita do resultado				
Instrução 3			Busca de Instrução	Decodificação de instrução	Execução	Acesso aos dados	Escrita do resultado			
Instrução 4				Busca de Instrução	Decodificação de instrução	Execução	Acesso aos dados	Escrita do resultado		
Instrução 5					Busca de Instrução	Decodificação de instrução	Execução	Acesso aos dados	Escrita do resultado	
Instrução 6						Busca de Instrução	Decodificação de instrução	Execução	Acesso aos dados	Escrita do resultado

Representação gráfica de Pipeline

- Recursos físicos



Referências

PATTERSON, D. A.; HENNESSY, J. L. Computer Organization and Design: The Hardware/Software Interface. Fourth edition.

