

**CURSO DE SISTEMAS DE INFORMAÇÃO**

**FÁBIO DINIZ**

**FRUTAL (MG)**

**2015**

## **SUMÁRIO**

|                            |          |
|----------------------------|----------|
| <b>INTRODUÇÃO</b>          | <b>3</b> |
| <b>CARACTERÍSTICAS</b>     | <b>3</b> |
| <b>EXEMPLO</b>             | <b>6</b> |
| <b>PROBLEMAS RELATADOS</b> | <b>7</b> |
| <b>VANTAGENS</b>           | <b>7</b> |
| <b>DESVANTAGENS</b>        | <b>8</b> |
| <b>CONCLUSÃO</b>           | <b>8</b> |
| <b>REFERÊNCIAS</b>         | <b>9</b> |

## INTRODUÇÃO

Atualmente, a grande maioria dos sistemas de computação é fabricado em torno de processadores que buscam maior velocidade na realização de suas atividades.

Um dos processos mais usados é o que conhecemos na indústria como linha de montagem ("pipeline"), no qual a UCP se divide em várias partes funcionais distintas (estágios), cada uma correspondendo a uma determinada atividade.

A ideia básica num pipeline de instruções é a de novas entradas serem aceitas, antes que as entradas aceitas previamente tenham terminado. Várias instruções são realizadas de forma simultânea, embora em estágios diferentes.

## CARACTERÍSTICAS

Tradicionalmente, as instruções do MIPS são executadas em até 5 passos:

1. Busca da instrução na memória;
2. Leitura dos registradores enquanto uma instrução é decodificada, (o formato das instruções do MIPS permite que a leitura e a codificação ocorram simultaneamente);
3. Execução de uma operação ou cálculo de um endereço;
4. Acesso a um operando de memória;
5. Escrita do resultado em um registrador;

As instruções e dados se movimentam geralmente da esquerda para a direita através de cada um dos cinco estágios. Existem, no entanto, duas exceções para este fluxo de instruções da esquerda para a direita:

- O Estágio de escrita no banco de registradores, que coloca o resultado em um dos registradores do banco, que está fisicamente situado no meio do caminho de dados.
- A seleção do próximo valor do PC, escolhido entre o PC incrementado e o endereço de desvio condicional que vem do estágio MEM.

Para reter o valor de determinada instrução durante seus outros quatro estágios, é necessário salvar em um registrador o valor lido da memória de instruções; portanto, é necessário que se coloque um registrador em cada uma das fronteiras entre os estágios de pipeline.

### **load Word (lw)**

1. Busca de instrução: A instrução é lida da memória, usando, para isso, o endereço armazenado no PC. O resultado da leitura é armazenado no registrador BI/DI. O endereço do PC é incrementado e armazenado de volta no PC. Esse valor do PC incrementado também é guardado em BI/DI;
2. Decodificação da instrução e leitura do banco de registradores: Os 16 bits menos significativos da instrução são colocados na entrada do módulo de extensão de sinal, além dos números dos registradores a serem lidos sendo colocados nas entradas correspondentes do banco de registradores. Todos os três valores resultantes dessas operações são armazenados no registrador DI/EX, junto com o valor incrementado do PC, que veio do estágio anterior.
3. Execução ou cálculo do endereço: O conteúdo do registrador 1 é lido do registrador DI/EX e o resultado da extensão de sinal também, adicionando-os, usando a ULA. O resultado dessa soma é colocado no registrador EX/MEM.
4. Acesso à memória: A memória de dados é lida a partir do endereço armazenado em EX/MEM e o dado resultante é carregado no registrador MEM/ER.

5. Escrita no banco de registradores: O dado lido do registrador MEM/EX é escrito no banco de registradores.

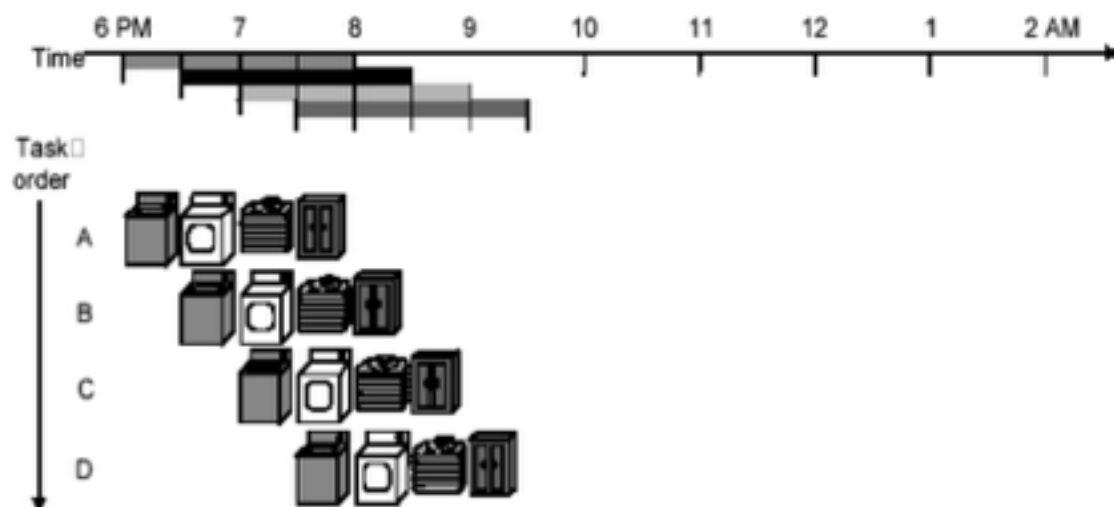
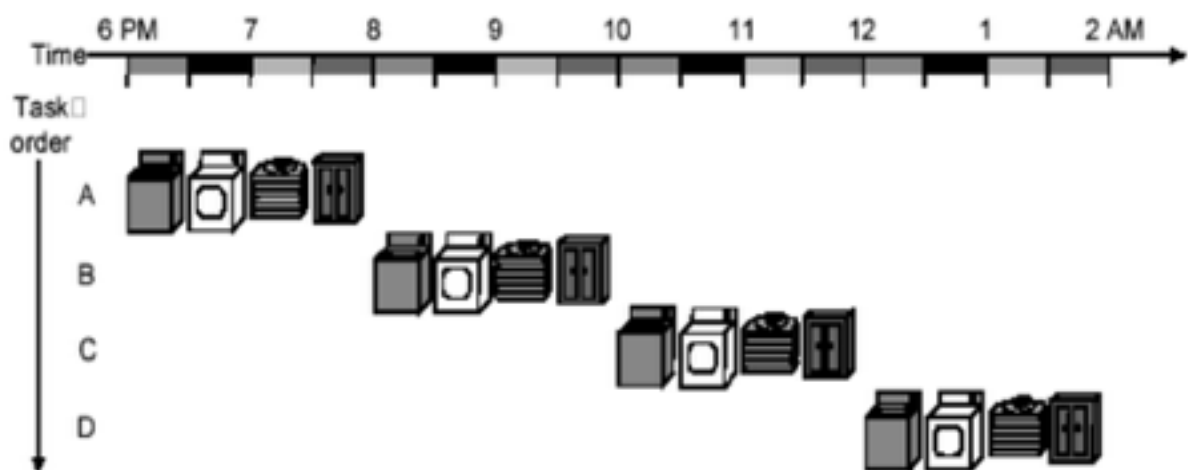
### **Controle do Processador Pipeline**

O PC é escrito em cada um dos ciclos de clock, de modo que não há necessidade de se ter um sinal de controle para a escrita nele. Da mesma forma, também não há necessidade de sinais de escrita para os registradores do pipeline.

1. Busca de instruções: os sinais de controle para ler a memória de instruções e para escrever no PC estão sempre ativos.
2. Decodificação da instrução / Leitura do banco de registradores: não há uma linha de controle adicional a ser ativada.
3. Execução / Cálculo do endereço: os sinais a serem considerados são RegDst, ULAOp, ULAFonte. Tais sinais selecionam o registrador no qual o dado será escrito, a operação da ULA e a segunda entrada da ULA, entre o dado presente na saída 2 do banco de registradores e o resultado da extensão do sinal do campo imediato da instrução.
4. Acesso à memória: as linhas de controle a serem consideradas neste estágio são DvC, LerMem e EscMem. Tais sinais são necessários às instruções de desvio se igual, load word e store Word, respectivamente. O sinal FontePC seleciona o próximo endereço na sequência de execução, a não ser que o controle tenha ativado o sinal DvC e que o resultado da ULA seja zero.
5. Escrita no banco de registradores: as duas linhas de controle a serem consideradas neste estágio são MemparaReg, que decide entre enviar para o banco de registradores o resultado da ULA ou o valor lido da memória de dados, e o EscReg, que comanda a escrita do valor escolhido.

## EXEMPLO

1. Vamos supor uma lavanderia, em que cada etapa possa ser realizada em 30 minutos:
2. Colocar a roupa na máquina de lavar
3. Depois de lavada, colocá-la na máquina de secar roupa
4. Depois de seca, passar a ferro
5. Depois de passada, arrumá-la no armário



- Supondo-se que cada uma destas etapas leve 30 minutos para ser realizada, a lavagem de um cesto de roupas continuará levando 2 horas para ser realizada.
- Entretanto, podemos iniciar a lavagem de um cesto de roupas a cada 30 minutos, até que tenhamos 4 cestos sendo lavados simultaneamente, um em cada etapa do “pipeline”.
- Depois das primeiras 2 horas, teremos um cesto de roupa lavada a cada 30 minutos. Ao final do dia teremos lavado muito mais cestos de roupa do que sem o uso de pipeline.

## PROBLEMAS RELATADOS

- Conflito estrutural: a infra-estrutura não suporta a “combinação” de (fases de) instruções que se pretende executar no mesmo ciclo. Exemplo: acesso a memória.
- Conflito de dados: uma (fase de) instrução depende dos resultados de outra. Exemplo: duas operações tipo R, em que a segunda usa os resultados da primeira. Solução: forwarding
- Conflito de controle: execução de uma instrução depende de uma decisão ainda não tomada. Soluções: protelar, prever, delayed decision.

## VANTAGENS

O uso de pipeline permite que todos os recursos envolvidos operem em paralelo, de maneira que mais instruções possam ser processadas na unidade de tempo considerada.

No multiciclo as instruções não são executadas na mesma quantidade de tempo, portanto, as instruções são executadas no seu tempo característico ( e não no tempo da instrução mais longa). Também não há ocorrência de conflitos.

## **DESVANTAGENS**

No multíciclo não é possível que várias instruções sejam processadas ao mesmo tempo. É preciso que uma instrução seja totalmente processada para que a próxima instrução comece sua execução. Isto torna mais lenta a execução de um grupo de instruções, pois há recursos não utilizados todo o tempo.

No pipeline, para haver um desempenho mais notável é necessário uma grande quantidade de instruções. Enquanto o pipeline não está totalmente cheio, a performance não é ideal. Além disso, há o fato do balanceamento imperfeito, o ciclo de clock deve ser de acordo com a instrução mais longa. Problemas de conflitos de dados, estruturais e de controle que necessitam de instruções que exigem acréscimo de hardware para sua solução que nem sempre são solucionadas completamente, havendo perda de efetividade do mesmo jeito.

## **CONCLUSÃO**

Em resumo, é o processo pelo qual uma instrução de processamento é subdividido 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. Entre os problemas enfrentados estão a dependência de instruções anteriores e desvios que dificultam o processo, bem como a diferença de complexidade de instruções que fazem com que as mesmas possam levar um tempo variável para execução.

A técnica de pipeline 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.



## REFERÊNCIAS

IFRN - Prof. Leandro Coelho. Disponível em: <<http://docente.ifrn.edu.br/moisessouto/disciplinas/organizacao-e-manutencao-de-computadores-i/oc-05-material-extre-sobre-pipeline>>. Acesso em: 18 jun. 2015 às 20h00.

UFPE. Disponível em: <<http://www.cin.ufpe.br/~jvwr/RESUMO%20pipeline.doc>>. Acesso em: 18 jun. 2015 às 20h00.

JCF, 2004 - ASPD (FEUP/LEEC) - João Canas Ferreira. Disponível em: <<http://paginas.fe.up.pt/~jca/feup/aspd/slides/aspd-slides-cpu-pipelining-1-52.pdf>>. Acesso em: 18 jun. 2015 às 20h00.