

Comparativo: Pipeline x Superpipeline x Superescalar

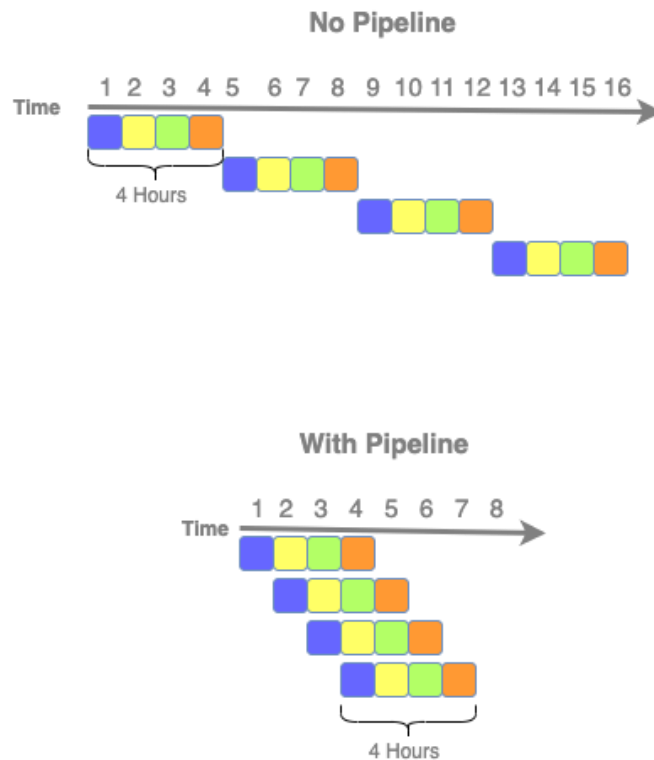
Davi Ventura Cardoso Perdigão

Professor (a): Adriana Dornas

Com a evolução da tecnologia e com o surgimento de aplicativos computacionais cada vez mais complexos, há de se observar que, para que este desenvolvimento fosse possível, foi necessária também a evolução dos processadores, pois são eles os responsáveis por executar tarefas com a maior eficiência, tornando-as viáveis computacionalmente. O presente artigo busca expor um comparativo entre as arquiteturas Pipeline, Superpipeline e Superescalar, buscando citar o que cada arquitetura agregou para a evolução computacional em sua época, seus prós e contras, dentre outros fatores.

Até o 386, os processadores da família x86 eram capazes de processar apenas uma instrução de cada vez. Uma instrução simples podia ser executada em apenas um ciclo de clock, enquanto instruções mais complexas demoravam vários ciclos de clock para serem concluídas. Para melhorar o desempenho dos modelos seguintes, como no caso do 486, a Intel resolveu usar o **pipeline**, uma técnica inicialmente usada em processadores RISC, que consiste em dividir o processador em vários estágios distintos. O 486, como citado anteriormente à caráter de exemplo, possui um pipeline de 5 níveis, ou seja, é dividido em 5 estágios. Quando é carregada uma nova instrução, ela primeiramente passa pelo primeiro estágio, que trabalha nela durante apenas um ciclo de clock, passando-a adiante para o segundo estágio. A instrução continua então sendo processada sucessivamente pelo segundo, terceiro, quarto e quinto estágios do processador. A vantagem desta técnica, é que o primeiro estágio não precisa ficar esperando a instrução passar por todos os demais para carregar a próxima, e sim carregar uma nova instrução assim que se livra da primeira, ou seja, depois do primeiro pulso de clock.

Figura 1 - Demonstrativo do tempo de execução sem Pipeline e com Pipeline.

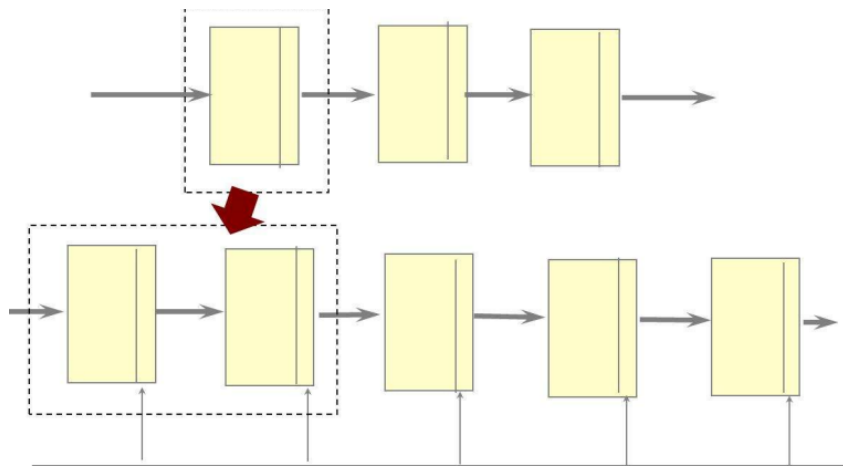


Fonte: PANTUZA, Gustavo. 2020.

Toda via, há de se observar que o uso dos 5 estágios de pipeline no 486 não chegava a multiplicar por cinco a performance do processador, nem mesmo dobrava, mas o ganho era bastante significativo. A medida em que mais estudos foram realizados, alguns pesquisadores perceberam que, quando se subdividia a pipeline em um maior número de níveis, alguns destes estágios precisavam de muito menos tempo de execução que um ciclo de clock. Assim, propôs-se o uso de um duplicador (ou até triplicador) de clock interno na CPU, de maneira que muitos destes estágios do pipeline pudessem ser finalizados na metade do tempo, acelerando ainda mais o processamento final. Essa é a arquitetura **Superpipeline**: técnica que explora o fato de que alguns estágios de pipeline executam tarefas que requerem menos de meio ciclo de clock, quebrando estágios em sub estágios (estágios menores). Assim, um processador que implemente esta técnica terá a velocidade de clock dobrada, e

permitirá um aumento no desempenho por ter duas tarefas executadas em um único ciclo de clock.

Figura 2 – Pipeline com maior profundidade.



Fonte: Universidade Federal de Pernambuco. 2018.

Em suma, as principais características da **arquitetura superpipelines**:

- Quebra estágio em sub estágios e cada sub estágio faz menos trabalho que o estágio original.
- Estágios menores demandam menos tempo para serem executados (período menor e frequência maior).
- Visa diminuir tempo de execução de um programa.

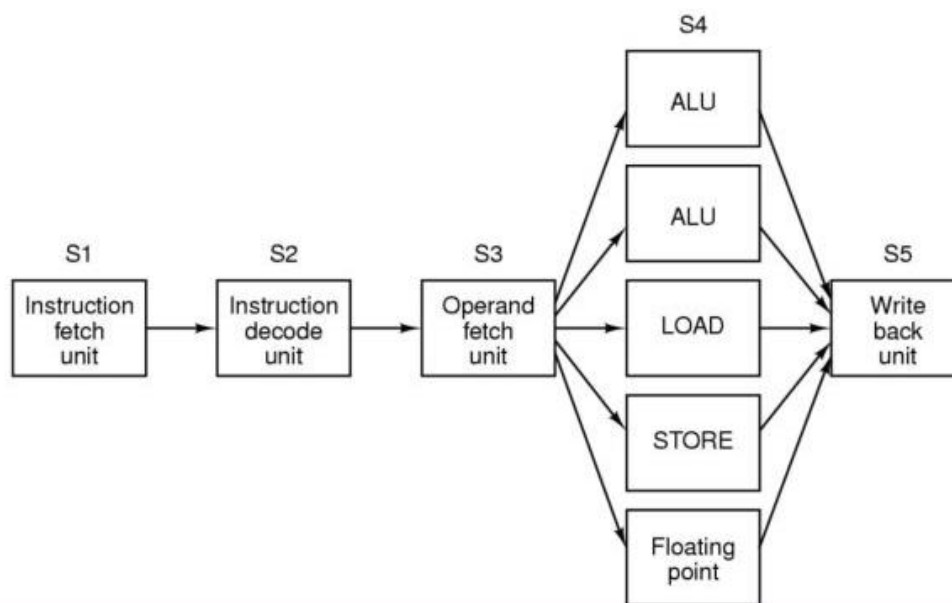
Expandindo a ideia de processar simultaneamente os diferentes estágios de instruções diferentes, foi sugerida a implementação de várias pipelines para a CPU,

de maneira que várias instruções pudessem, de fato, serem executadas ao mesmo tempo, desde que fossem independentes. O princípio é o mesmo da pipeline já visto, mas nessas novas CPUs foram inseridas, em uma mesma CPU, diversas pipelines. Essa estratégia é conhecida como **paralelismo em nível de instruções**, porque ela é capaz de detectar instruções independentes e executá-las simultaneamente. Segundo (Stallings, 1996), numa implementação da arquitetura Superescalar de um processador, instruções comuns como: aritmética inteira e de ponto flutuante, leitura e escrita na memória e desvios condicionais, podem ser inicializadas simultaneamente e executadas independentemente e tal implementação aumenta o grau de complexidade relacionada às instruções pipeline.

Em suma, a **arquitetura Superescalar** é uma técnica de implementação que permite que várias instruções sejam iniciadas simultaneamente e são executadas independentemente uma das outras. Isso ocorre, pois esta arquitetura replica os componentes internos do processador, permitindo assim colocar várias instruções em cada estágio do pipeline. As principais características dessa arquitetura são:

- Possibilita a execução de instruções em pipelines paralelos;
- Pipelines paralelos são possíveis replicando unidades funcionais;
- Utilizam uma combinação de otimização de compiladores e técnicas de hardware para aumentar o nível de paralelismo;

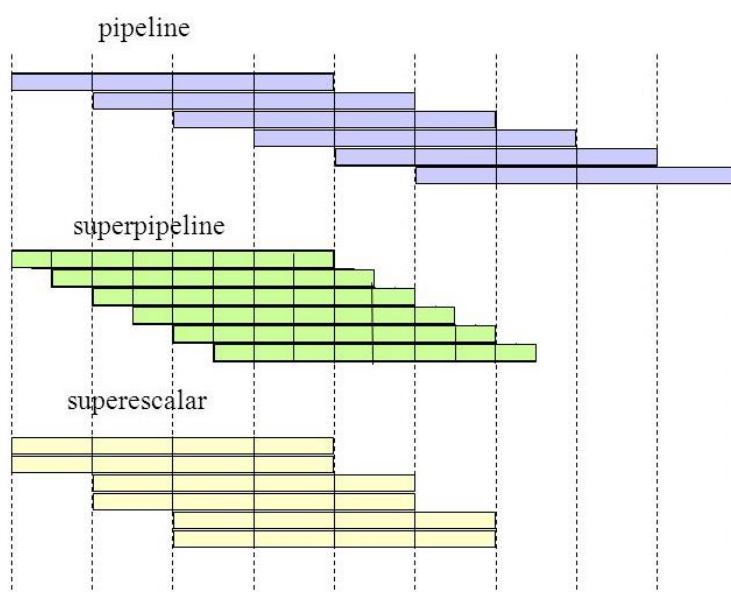
Figura 3 – Processador Superescalar - Pentium.



Fonte: NATLIA, Larissa. 2009.

O gráfico a seguir mostra o processamento Superescalar comparado com o processamento em pipeline e Superpipeline:

Figura 4 – Comparativo: Pipeline x Superpipeline x Superescalar.



Fonte: MADRUGA, Stéphanie. 2015.

Observe que, diferentemente das arquiteturas pipeline e Superpipeline, na arquitetura Superescalar temos, de fato, mais de uma instrução sendo processada ao mesmo tempo. Isso é muito útil para processamento dados como imagens e outros tipos de matrizes, onde é possível processar diversos dados diferentes ao mesmo tempo pois, em geral, o processamento de cada pixel ou elemento da matriz é independente do processamento dos outros pixels/elementos da matriz.

A execução de programas em arquiteturas Superescalar e Superpipeline apresentam um aumento de throughput (número de tarefas concluídas por unidade de tempo) e com isto pode existir uma melhora de desempenho. Mas nem sempre isto é real, pois necessita que exista uma execução ideal das sequencias de instruções. Então a aplicação dos conceitos dessas arquiteturas nos processadores deve ser feita com cautela para minimizar os erros. É de extrema importância estarmos atentos às novas demandas computacionais, com base nas arquiteturas que temos hoje, para observarmos novas potenciais melhorias que virão à ser implementadas em futuros processadores.

Referências Bibliográficas:

- 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.
- STALLINGS, William. **Arquitetura e Organização de Computadores**. Tradução da 8a edição. Editora Prentice Hall Brasil, 2002.
- MURDOCCA, M. J; HEURING, V.P. **Introdução à arquitetura de computadores**. S.I.: Ed. Campus, 2000.
- TANENBAUM, Andrew S. **Organização Estruturada de Computadores**. Tradução da 5a edição. Editora Prentice Hall Brasil, 2007.
- VASCONCELOS, Laércio. **Hardware Total**. 1ª Edição. Editora Makron Books, 2002