

Arquitetura de Computadores

Arquitetura e Funcionamento dos Computadores

Parte 007

MACHINE CYCLE (CICLO DA MÁQUINA)

PIPELINING

O ciclo da máquina consiste num processo contínuo de execução pela CPU de um conjunto de etapas conducente ao processamento das instruções dos programas em execução.

O ciclo da máquina é constituído por quatro fases, nomeadamente ***Fetch Instruction*** (busca da instrução seguinte a ser executada), **Descodificação**, **Execução** e **Armazenamento**.

- **Fetch**

Vai buscar à memória a próxima instrução a ser executada.

Nesta fase, a Unidade de Controle vai buscar à memória RAM (ou cache), a instrução que está armazenada no endereço indicado (apontado) pelo registo Contador do Programa (PC).

- **Descodificação**

Nesta fase, a Unidade de Controle descodifica a instrução no **Registo de Instrução (IR)**, que implica a divisão do campo do operando pelos componentes. A interpretação da instrução requer normalmente o acesso a microcódigo armazenado em micro ROMs que se encontram dentro do Processador.

Pode ser necessário ocorrer mais buscas à memória, no caso de instruções cálculos com dados, sendo assim necessário obter os seus valores (RAM/Cache).

- **Execução**

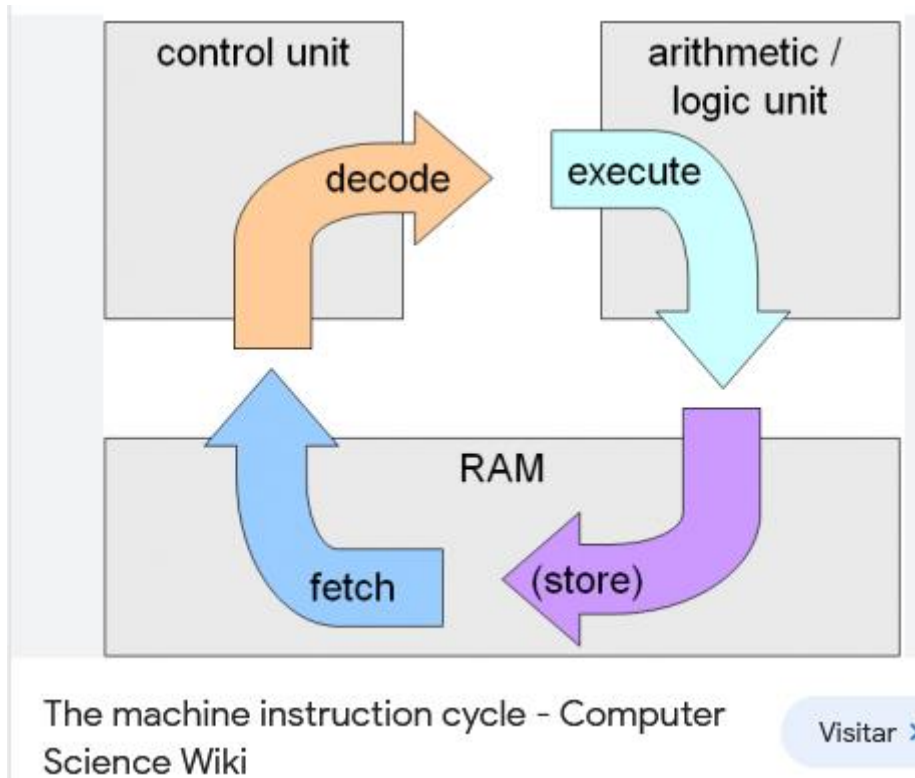
Executa a instrução.

Nesta fase, a instrução é executada pela Unidade Aritmética e Lógica do CPU.

- **Armazenamento**

Armazena/grava os resultados na memória.

A seguinte imagem é ilustrativa do processo das fases do ciclo da máquina:



Notas:

O processo do ciclo da máquina repete-se e assim continuamente.

A execução de uma instrução, é dividida pelos seguintes passos:

- 1 - Cálculo do endereço de memória que contém a instrução
- 2- Busca da informação na RAM
- 3 - Decodificação da instrução
- 4 - Cálculo dos endereços dos operandos (isto se houver dados)
- 5 - Busca de valores dos operandos (dados)
- 6 - Execução da instrução
- 7 - Armazenamento dos resultados

Vejamos um exemplo concreto da parte inicial da Fase Fetch, mais em detalhe:

Descreva, de uma forma concreta e sintética, mas passo a passo, a fase de FETCH, de uma instrução elementar do tipo $X = X + Y$.

Nota:

A parte inicial da fase fetch do Ciclo da Máquina, abaixo descrita, usa nomes de Registos genéricos, tais como PC, MBR, MAR e IR, independentemente da arquitetura de qualquer computador, quer seja organizada ou não por segmentos.

- O registo PC (Program Counter) é carregado com o endereço da instrução a ser trazida da RAM para o Processador para decodificação.
- O valor do registo PC é então carregado no registo de endereços MAR (Memory Address Register).
- O valor do registo MAR é carregado (os sinais que compõem o endereço da instrução) na Address Bus (via de endereços).
- O Processador emite o Sinal de Controle Read (para leitura da Memória).
- A localização da RAM (onde se encontra a instrução) é acedida e dá-se a transferência do código da instrução ($X = X + Y$) para o Data Bus (via de dados).
- O valor do código da instrução que se encontra no data bus é carregado no registo MBR (Memory Buffer Register).
- O valor do MBR é transferido para o registo IR (Instruction Register), para decodificação da instrução, dando assim por terminada a fase *fetch* de uma instrução do ciclo da máquina e início da fase Interpretação da Instrução.

Machine Cycle em Pipelining

“A **segmentação de instruções** (em inglês, *pipeline*) é uma técnica *hardware* que permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada. Estas instruções 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.

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 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”.

Vejamos a ilustração abaixo:

Let us consider the following decomposition of the instruction processing.

- **Fetch instruction (FI):** Read the next expected instruction into a buffer.
- **Decode instruction (DI):** Determine the opcode and the operand specifiers.
- **Calculate operands (CO):** Calculate the effective address of each source operand. This may involve displacement, register indirect, indirect, or other forms of address calculation.
- **Fetch operands (FO):** Fetch each operand from memory. Operands in registers need not be fetched.
- **Execute instruction (EI):** Perform the indicated operation and store the result, if any, in the specified destination operand location.
- **Write operand (WO):** Store the result in memory.

No exemplo abaixo, pode-se observar o seguinte:

Tempo 1:

A instrução 1 é buscada da Memória (**FI**).

Tempo 2:

Enquanto a instrução 1 está a ser decodificada (**DI**) a instrução 2 é trazida da Memória (**FI**).

Tempo 3:

Enquanto a instrução 1 encontra-se no Cálculo dos Endereços dos Operandos (**CO**) para serem trazidos para o Processador os valores de **X** e **Y**, a instrução 2 está a ser

descodificada (**DI**) e a instrução 3 está a ser trazida da memória para o Processador (**FI**).

Tempo 4:

Enquanto a instrução 1 está a trazer para o Processador os valores dos operandos **X** e **Y**, a instrução 2 encontra-se no **Cálculo dos endereços dos Operandos (CO)** para serem trazidos para o Processador os valores da instrução, a instrução 3 está a ser descodificada (**DI**) e a instrução 4 está a ser trazida da memória para o Processador (**FI**).

E assim sucessivamente:

No **tempo 5** dá-se a execução da instrução (**EI**) em paralelo com outras ocorrências relativas a outras instruções.

Finalmente relativamente à instrução 1, no **tempo 6**, dá-se a escrita (**WO**) do resultado da sua execução, terminando assim a resolução do ciclo da máquina para a instrução 1.

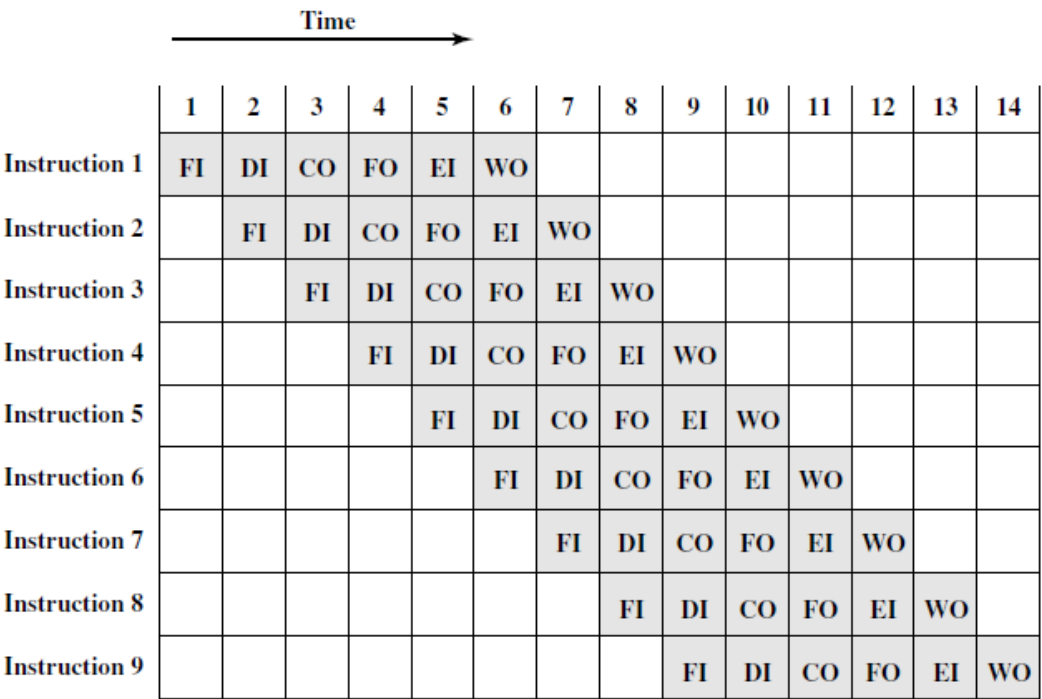


Figure 12.10 Timing Diagram for Instruction Pipeline Operation