

Arquitetura de Computadores

Unidade Central de Processamento – CPU

CPU

- Funções realizadas pelo processador:
 - Função controle

Responsável pela busca, interpretação e controle da execução das instruções, bem como do controle da ação dos demais componentes do sistema de computação (memória, entrada/saída).
 - Função processamento

Responsável por realizar as atividades relacionadas com a efetiva execução de uma operação, ou seja, processar, realizar um cálculo aritmético ou lógico, etc.

CPU

- Componentes de uma CPU

- Unidades Funcionais

- São os módulos da CPU responsáveis pela busca e o processamento das instruções.

- Registradores

- É a memória interna da CPU que possui conexão direta com as Unidades Funcionais. São produzidos com a mesma tecnologia da CPU possuindo assim a mesma velocidade.

- Barramento Interno

- Responsável por interconectar os registradores às Unidades Funcionais.

CPU

Os registradores são divididos em dois tipos:

– Uso Geral (dados)

Os registradores de uso geral são usados para armazenar dados da execução do programa, como os valores que serão operados e o resultado das operações.

- Acc (Acumulador) → Este registrador é especial porque ele é normalmente usado como fonte e destino de dados e costuma ser um operando implícito nas instruções.

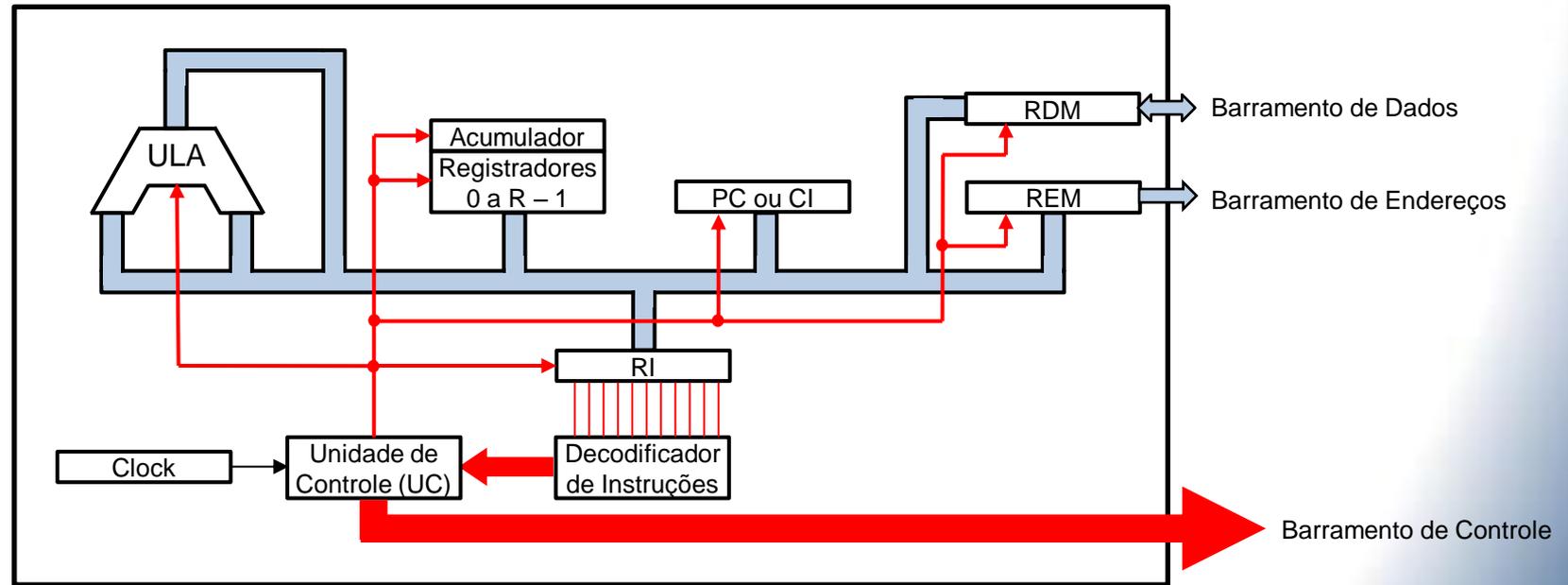
– Uso Específico (informações de uso interno da CPU)

Os registradores de uso específico guardam informações especiais, como por exemplo:

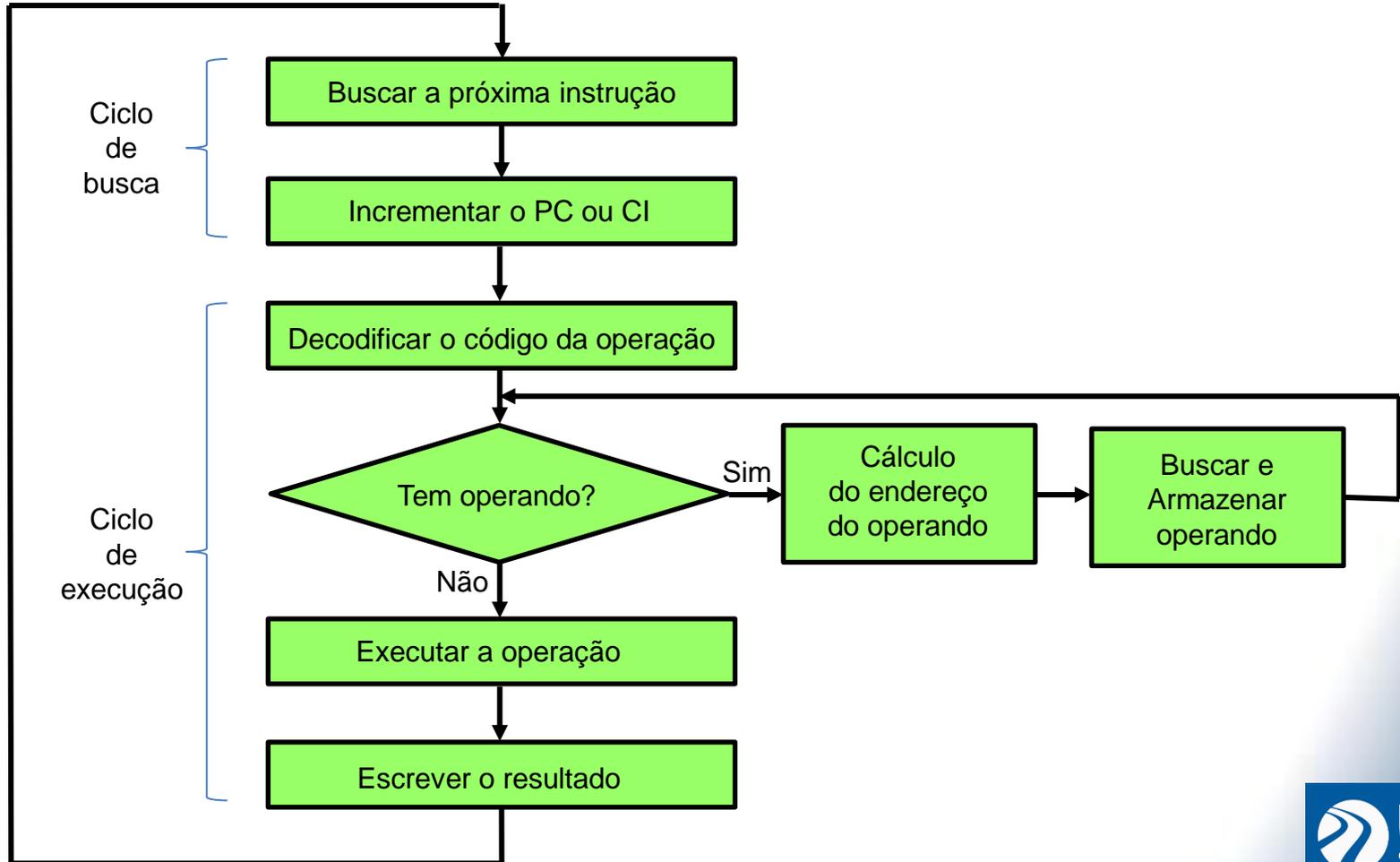
- REM → Guarda o endereço utilizado na comunicação com a memória.
- RDM → Guarda o dado utilizado tanto na leitura como na escrita com a memória.
- PC ou CI → Sigla proveniente de “Program Counter” e “Contador de Instrução” respectivamente. Este registrador guarda o endereço da próxima instrução a ser executada pela CPU.
- RI → Sigla proveniente de “Registrador de Instruções”. Este registrador guarda a instrução que está sendo executada pela CPU.
- FLAGS ou STATUS → Registrador especial para guardar informações referente a última execução realizada e sinais de controle para a CPU realizar controle de fluxo (desvios).

CPU

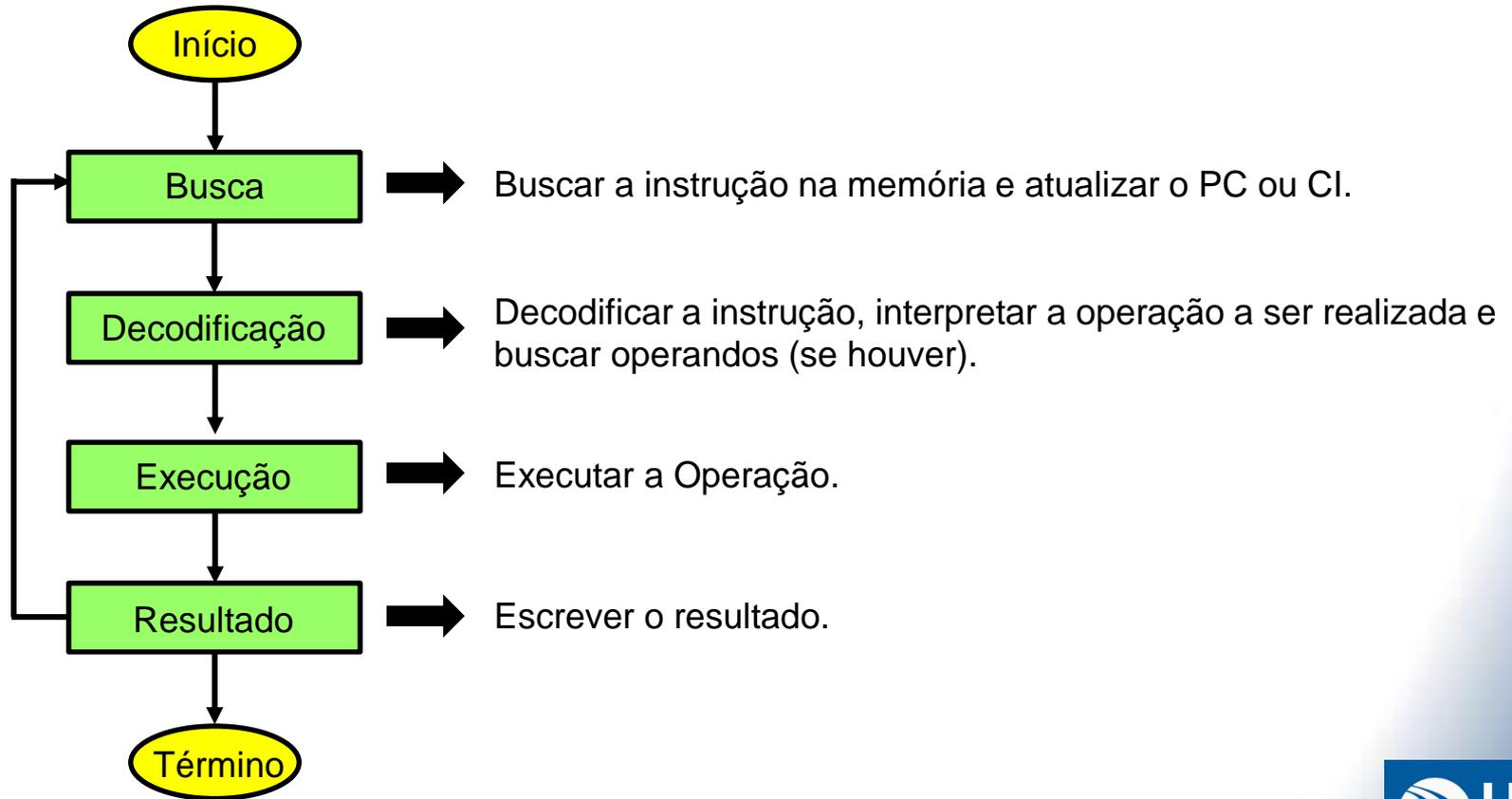
Componentes da CPU



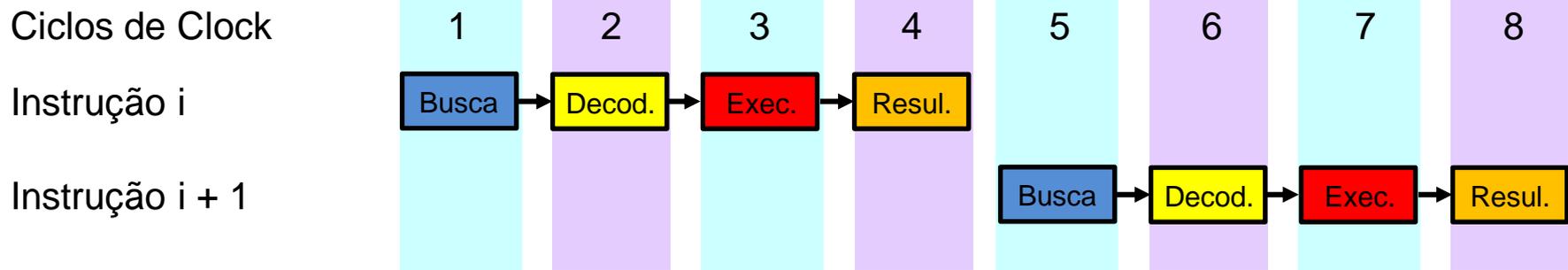
Fluxograma de um Ciclo de Instrução



Fluxograma resumido de um Ciclo de Instrução



Sequência de Execução das Instruções em uma CPU tradicional

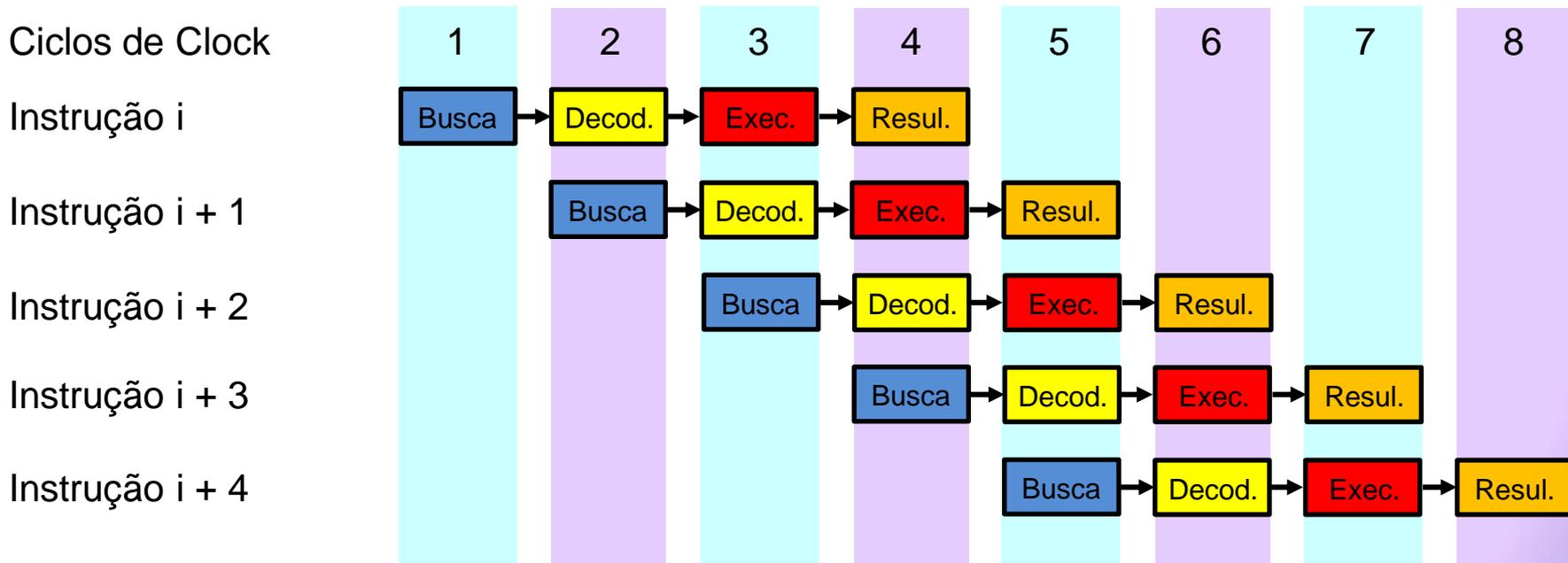


Como podemos observar, teremos uma instrução finalizada a cada 4 ciclos de clock. Isto é, 1 inst/4 ciclos ou 0,25 inst/ciclo.

Aprimoramento da CPU

- Enquanto a instrução encontra-se em um determinado estágio, nota-se claramente a ociosidade dos demais estágios da CPU.
- Um aprimoramento realizado no projeto da CPU foi de minimizar estas ociosidades implementando a possibilidade de executar várias instruções ao mesmo tempo, porém em estágios diferentes.
- A este tipo de funcionalidade da CPU dá-se o nome de **Pipeline**.

Sequência de Execução das Instruções em uma CPU com Pipeline

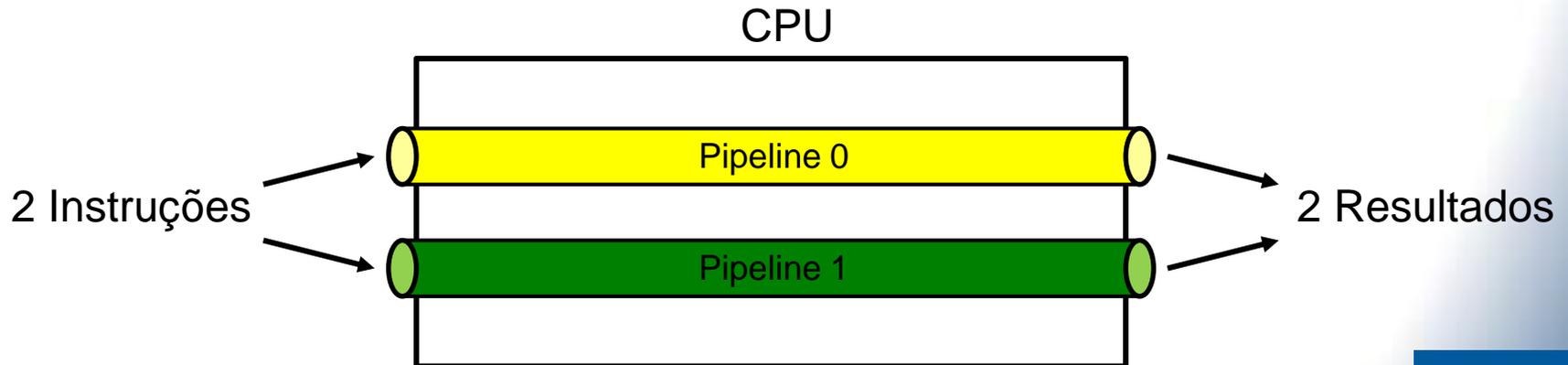


Como podemos observar, após o preenchimento do pipeline, teremos uma instrução finalizada a cada ciclo de clock, logo temos, 1 inst/ciclo.

O aumento de desempenho foi de 4. Isto é, o aumento de desempenho é equivalente ao número de estágios do pipeline quando o pipeline estiver cheio.

Aprimoramento da CPU

- Um novo aprimoramento realizado foi de implementar mais de um pipeline, possibilitando a execução de mais de uma instrução simultaneamente no mesmo estágio.
- A este tipo de CPU dá-se o nome de **Superescalar**.

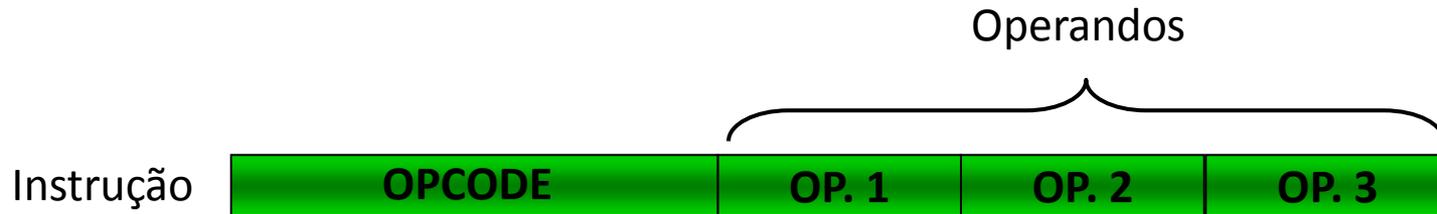


Formato das Instruções de Máquina

- Como será que é formada a instrução de máquina?
- Devemos ter em mente que a CPU possui um circuito para cada tarefa definida pelo fabricante.
- Para acessar estes circuitos e então ter tal operação realizada, eles foram identificados com um código numérico.
- Sempre que se desejar executar uma determinada operação, deve-se indicar o código do circuito em questão.
- Este código é conhecido como **opcode**.

Formato das Instruções de Máquina

- Além de indicar a operação, a maioria delas necessita de parâmetros (operandos). Então, além do opcode, a instrução possui campos com os operandos da operação em questão.
- Nas arquiteturas mais comuns, as instruções podem ir desde apenas o opcode, zero operando, até três operandos.



Modos de Endereçamentos

- Colocar o dado diretamente na instrução muitas das vezes é insuficiente para algumas estruturas de dados. Sendo assim, existem outras formas de informar o local onde os dados ou instruções estão guardados.
- Dentre os diversos modos de endereçamentos desenvolvidos para processadores, os principais são:
 - Imediato;
 - Direto;
 - Indireto;
 - Por registrador;
 - Indexado;
 - Base mais deslocamento.
- Nem todos estes modos estão incluídos nos conjuntos de instruções dos processadores contemporâneos, mas devem ser descritos não só por razões históricas mas para percepção do processo evolutivo da tecnologia.

Modos de Endereçamentos

Modo Imediato

- O método mais simples e rápido de obter um dado é indicar seu próprio valor no campo operando da instrução, em vez de buscá-lo na memória.
- A vantagem desse método reside no curto tempo de execução da instrução, pois não gasta ciclo de memória para a busca do dado, exceto o único requerido para a busca da instrução.
- Assim, o dado é transferido da memória juntamente com a instrução (para o RI), visto estar contido no campo operando da instrução.
- Este modo é útil:
 - Para inicialização de contadores (um valor sempre fixo em toda execução do mesmo programa);
 - Na operação com constantes matemáticas;
 - Para armazenamento de ponteiros em registradores do processador;
- Duas de suas desvantagens consistem na limitação do tamanho do campo operando das instruções o que reduz o valor máximo do dado a ser manipulado e o fato dos valores serem constantes, já que as instruções não podem ser modificadas ao longo da execução.

Modos de Endereçamentos

Modo Imediato

Operando

Instrução



Registrador
ou ULA



O dado é transferido diretamente da instrução para o Registrador ou ULA.

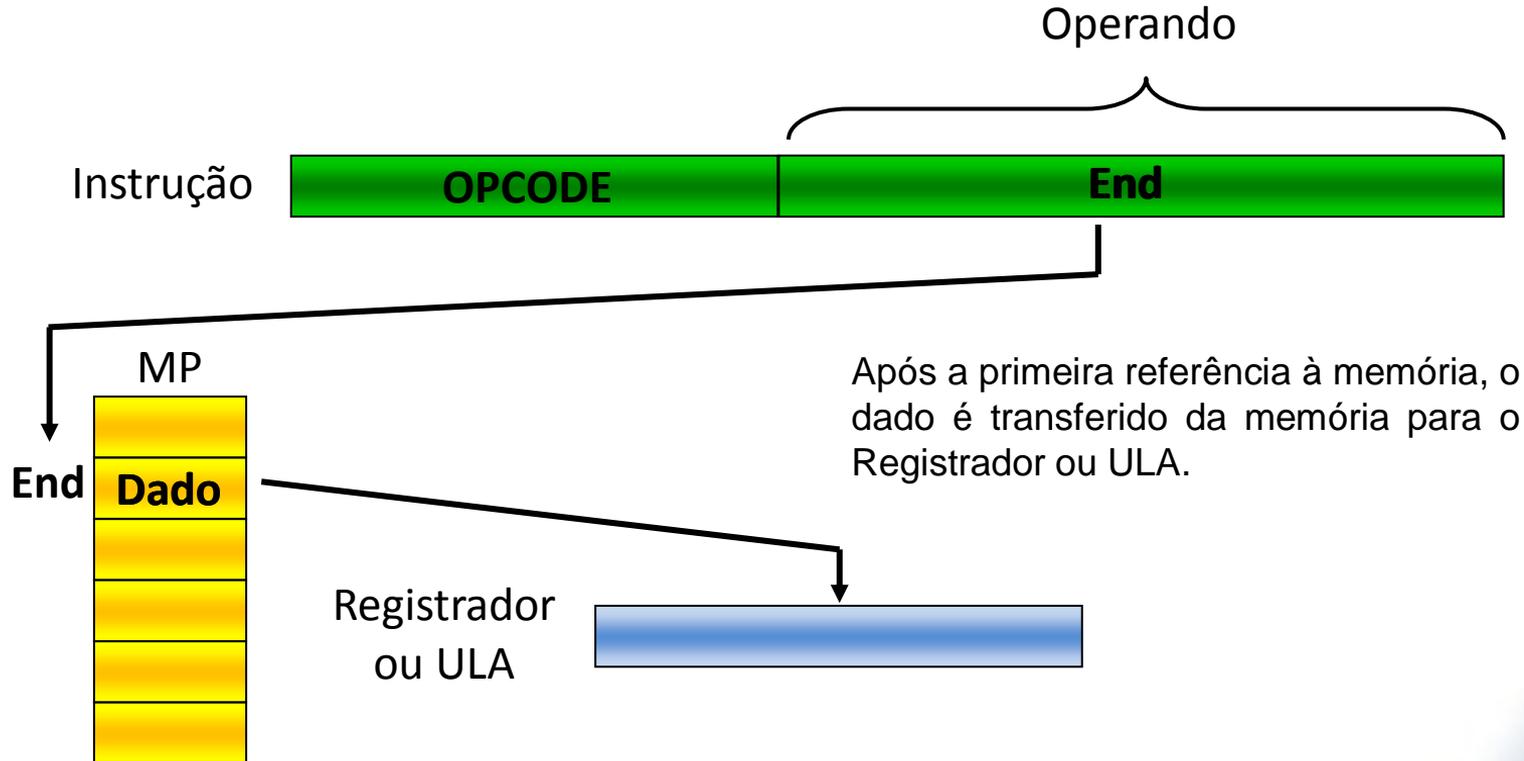
Modos de Endereçamentos

Modo Direto

- Para contornar as limitações do modo de endereçamento imediato, o modo direto guarda o dado na Memória Principal.
- Sendo assim, o valor binário contido no campo operando da instrução indica o endereço de memória onde se localiza o dado.
- O endereço pode ser o de uma célula onde o dado está inteiramente contido ou a célula onde começa o armazenamento do dado no caso dele ocupar mais de uma célula.
- É também um modo simples de acesso, pois requer apenas uma referência à MP para buscar o dado, sendo, porém mais lento que o modo imediato devido à referência à memória.
- Quando um dado for variar de valor ao longo da execução do programa, a melhor maneira de utilizá-lo é, inicialmente, armazená-lo na MP. O programa, então, usa o dado através do modo direto, onde a instrução indica apenas o endereço onde ele se localiza. Esta é a característica de uma variável de programa: representar o endereço de uma variável.
- Uma possível desvantagem desse processo está na limitação da memória a ser usada, conforme o tamanho do campo operando. Isto é, se o campo tiver um tamanho, por exemplo, de 12 bits, com o emprego do modo direto somente se pode acessar as células de endereço na faixa de 0 a $(4.095)_{10}$.

Modos de Endereçamentos

Modo Direto



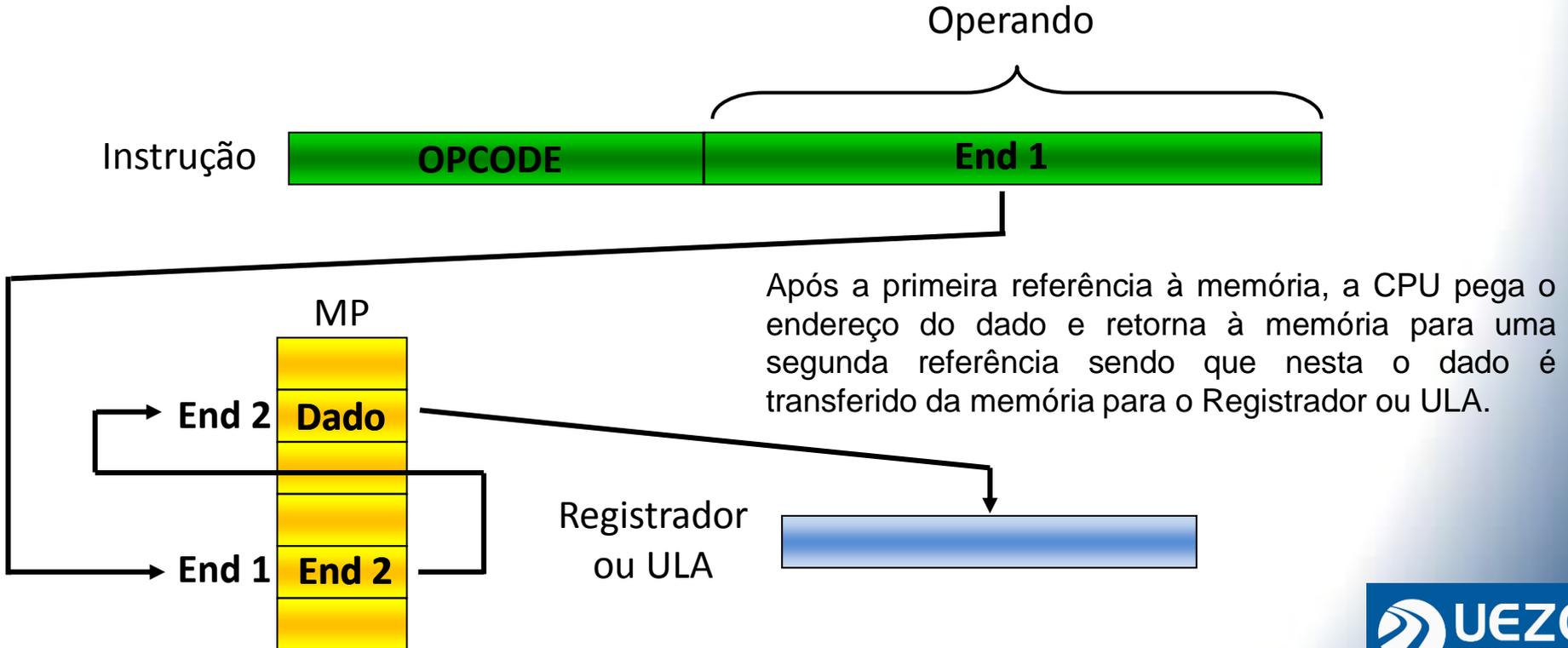
Modos de Endereçamentos

Modo Indireto

- Para contornar as limitações do modo de endereçamento direto, o modo indireto guarda o endereço do dado na Memória Principal.
- Nesse método, o valor binário contido no campo operando da instrução representa o endereço onde está armazenado o endereço do dado.
- Assim, há um duplo endereçamento para o acesso a um dado e, conseqüentemente, mais ciclos de memória para buscar o dado, comparativamente com os métodos já apresentados.
- O endereço intermediário (conteúdo da célula endereçado pelo valor do campo operando) é conhecido como ponteiro, pois indica a localização do dado.
- Com esse processo, elimina-se o problema do modo direto, delimitação do valor do endereço do dado, pois estando o endereço armazenado na memória (podendo ocupar uma ou mais células), este se estenderá ao tamanho necessário à representação do maior endereço da MP do sistema de computação em uso além de poder ser modificado ao longo da execução do programa.

Modos de Endereçamentos

Modo Indireto



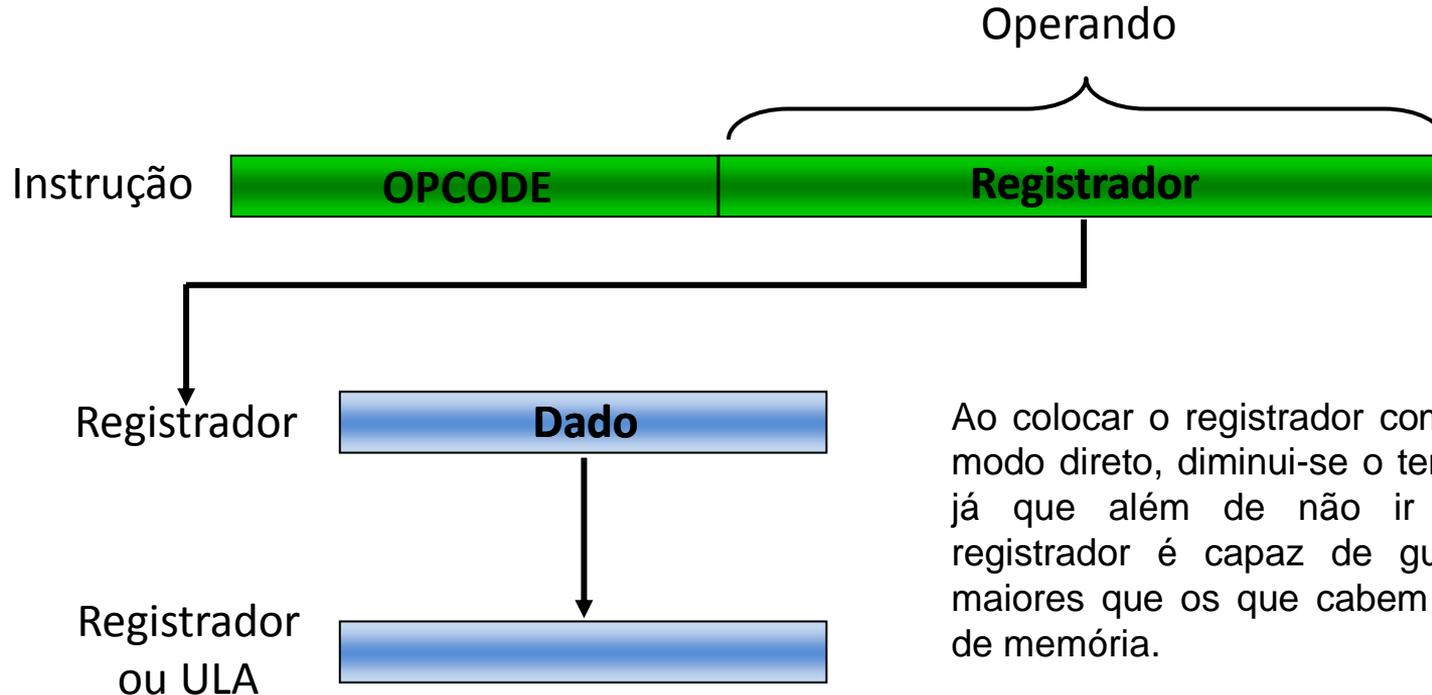
Modos de Endereçamentos

Endereçamento por Registrador

- Esse método tem características semelhantes aos modos direto e indireto, exceto que a célula de memória referenciada na instrução é substituída por um dos registradores do processador. Com isso, o endereço mencionado na instrução passa a ser o de um dos registradores, e não mais de uma célula da MP.
- A primeira vantagem logo observada, consiste no menor número de bits necessários para endereçar os registradores, visto que estes existem em muito menor quantidade que as células de memória. Isso reduz o tamanho geral das instruções.
- Um computador que tenha, por exemplo, um processador com 16 registradores, requer apenas 4 bits para endereçá-los; no caso de endereçamentos de células da MP, como o espaço de endereçamento da memória é grande, há necessidade de 20 ou mais bits para indicar o endereço de cada uma das células.
- Outra vantagem está no próprio emprego do dado, que passa a ser armazenado em um meio (registrador) cujo acesso é muito mais rápido que o acesso à memória.

Modos de Endereçamentos

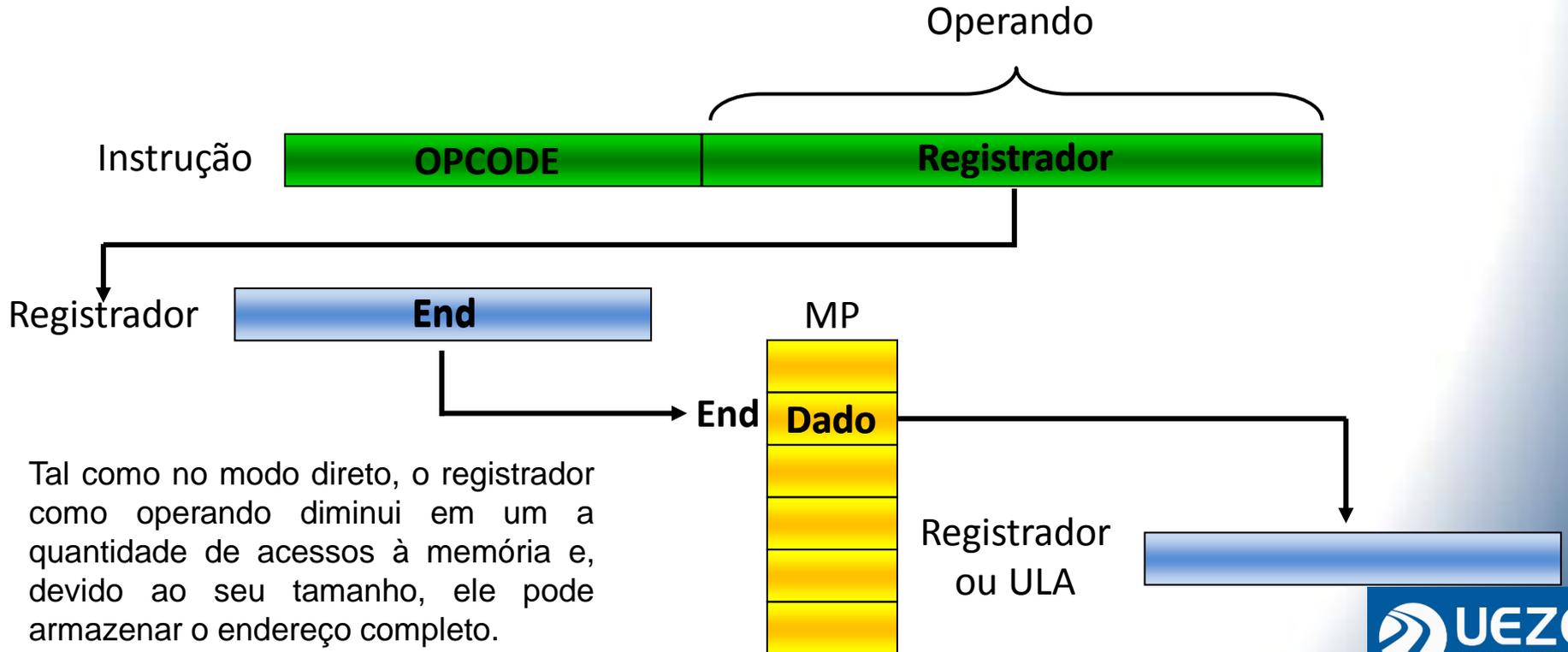
Modo Direto por Registrador



Ao colocar o registrador como operando no modo direto, diminui-se o tempo de acesso, já que além de não ir a memória o registrador é capaz de guardar números maiores que os que cabem em uma célula de memória.

Modos de Endereçamentos

Modo Indireto por Registrador



Tal como no modo direto, o registrador como operando diminui em um a quantidade de acessos à memória e, devido ao seu tamanho, ele pode armazenar o endereço completo.

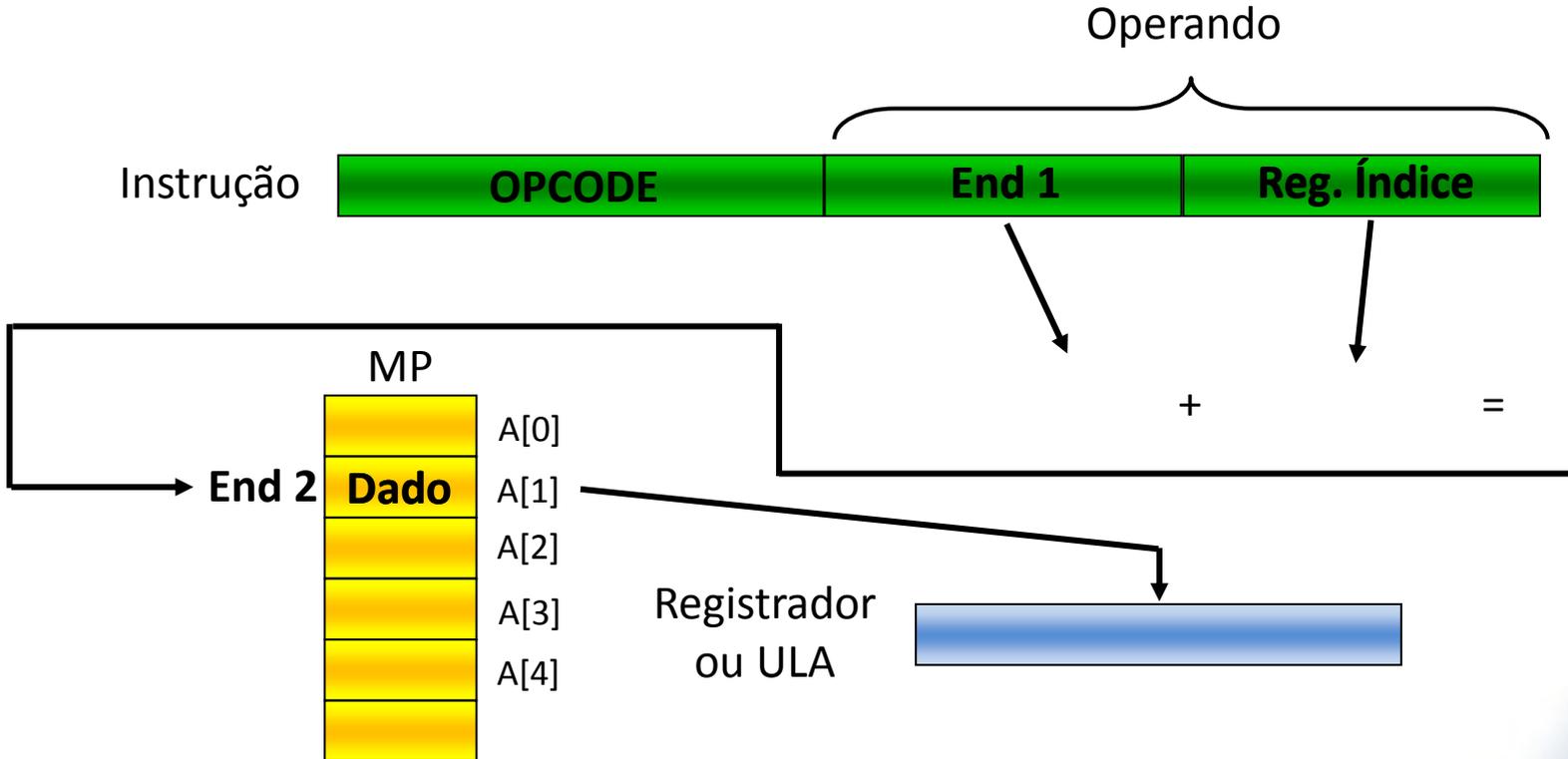
Modos de Endereçamentos

Modo Indexado

- Frequentemente, durante a execução dos programas, existe a necessidade de se manipular endereços de acesso a elementos de certos tipos especiais de dados. Esses endereços servem, na realidade, de ponteiros para os referidos elementos.
- Por exemplo, o acesso aos elementos de um vetor deve considerar que tais elementos são armazenados sequencialmente na memória e que sua localização pode ser referenciada por um ponteiro (endereço) que é alterado para indicar o elemento desejado (índice do elemento identifica univocamente cada um).
- A descrição dessas instruções caracterizam o modo de endereçamento denominado indexado. Esta denominação advém do fato de que a obtenção do endereço de um dado (elemento de um array) relaciona-se com o seu índice.
- Nesse tipo de instrução, o endereço do dado é a soma do valor do campo operando (fixo para todos os elementos de um dado array) e de um valor armazenado em um dos registradores (normalmente denominado registrador índice). O valor armazenado nesse registrador varia para o acesso a cada elemento.
- Na verdade, esse modo de endereçamento é uma evolução das técnicas desenvolvidos desde os primórdios da computação para manipulação dessas estruturas de dados especiais.

Modos de Endereçamentos

Modo Indexado



Modos de Endereçamentos

Modo Base mais Deslocamento

- Este modo de endereçamento tem característica semelhante ao modo indexado, visto que o endereço de acesso a uma célula de memória se obtém através da soma de dois valores, um inserido no campo apropriado da instrução (normalmente denominado campo deslocamento) e o outro valor inserido é em um determinado registrador, denominado registrador – base ou registrador de segmento.
- A diferença entre eles está na aplicação do propósito do método e, por conseguinte, na forma de implementá-lo. Nesse caso, o valor a se manter fixo é o do registrador – base/segmento, variando o conteúdo do campo deslocamento em cada instrução, diferentemente do modo indexado, onde o conteúdo do registrador é que se altera.
- Os processadores da família Intel x86 possuem alguns registradores projetados especificamente com a finalidade de servir como registrador de segmento, como os registradores de 16 bits dos primeiros processadores Pentium seguidos posteriormente pelos registradores de segmentos de 32 bits.
- Este método de endereçamento acarreta uma redução do tamanho das instruções (e, com isso, economiza memória), bem como facilita o processo de realocação dinâmica de programas.

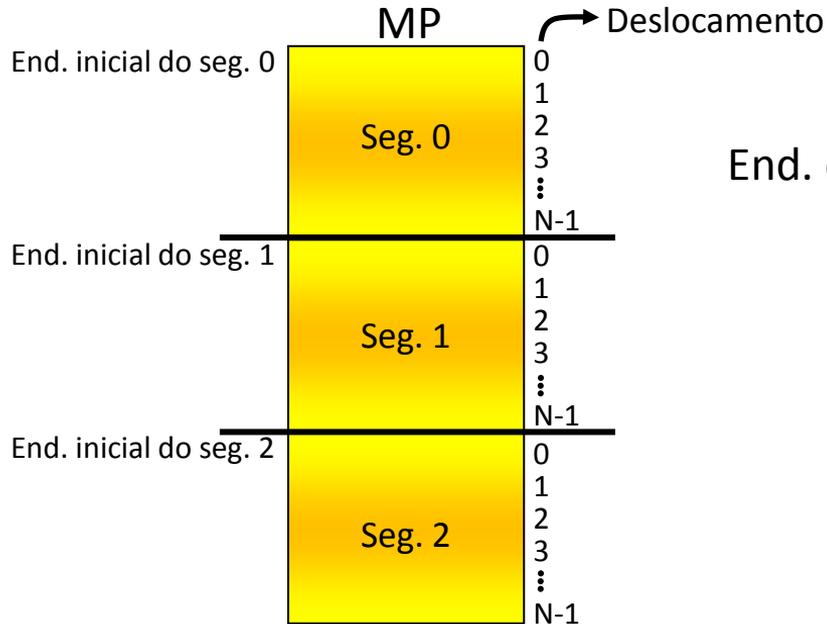
Modos de Endereçamentos

Modo Base mais Deslocamento

- A sua escolha decorre de dois fatores:
 - a) Durante a execução de uma grande quantidade de programas as referências as células de memória, onde se localizam os operandos, normalmente são sequenciais, ocorrendo poucos acessos a outras instruções fora de ordem (exceto os desvios);
 - b) A maioria dos programas ocupam um pequeno espaço da MP disponível.
- Dessa forma, em vez de ser necessário, em cada instrução, que o campo operando tenha um tamanho correspondente à capacidade total de endereçamento da MP, basta que o endereço desejado seja obtido pela soma de um valor existente em um dos registradores da CPU com o valor contido na instrução.
- Por isso o método é chamado de base + deslocamento, consistindo, então, na utilização de dois campos na instrução (que substituem o campo operando): um, com o endereço de um registrador (chamado de base ou segmento), e outro, com valor denominado deslocamento (porque contem um valor relativo – que se desloca em relação à primeira instrução).

Modos de Endereçamentos

Modo Base mais Deslocamento

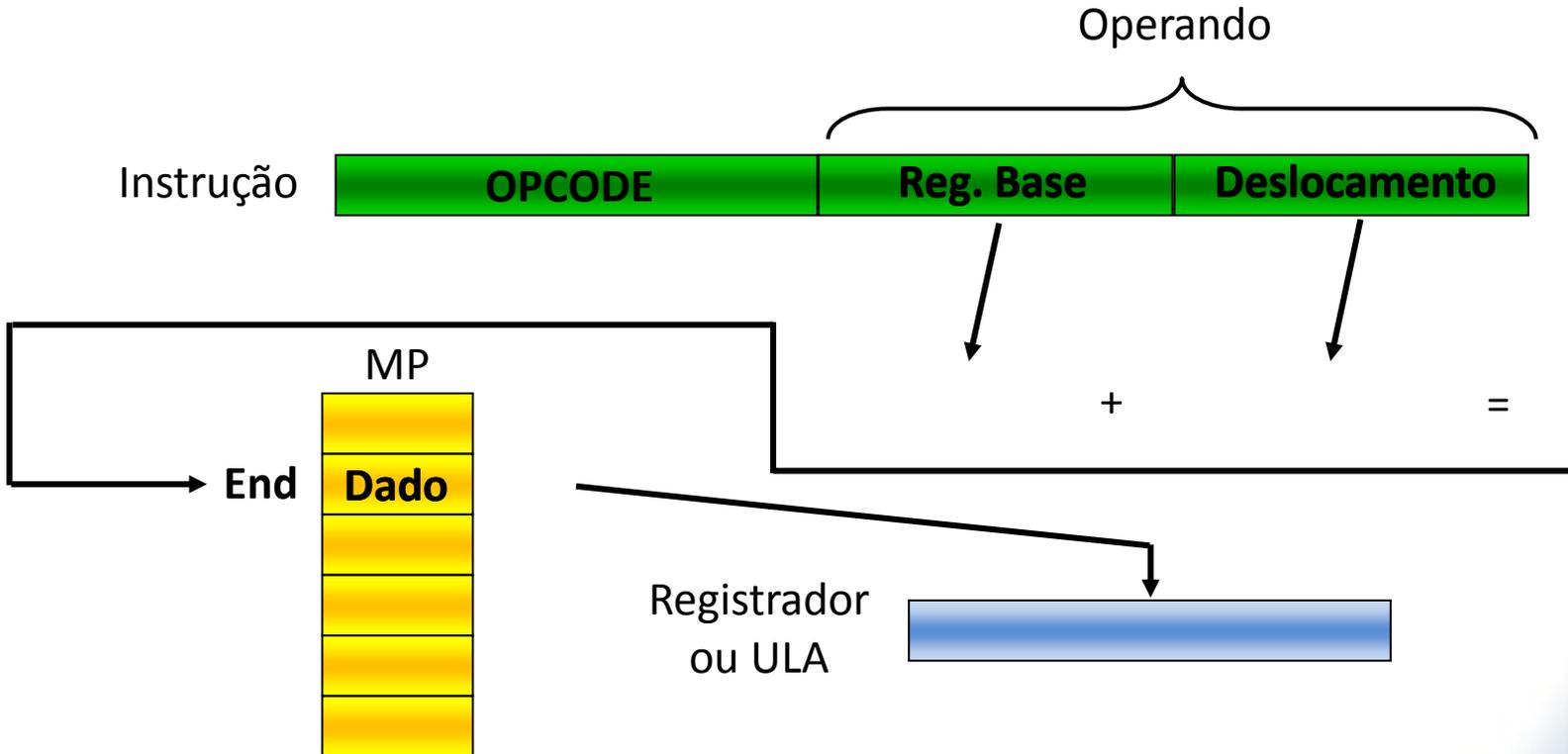


$$\text{End. do Dado} = \text{End. Inicial do Seg.} + \text{Deslocamento}$$



Modos de Endereçamentos

Modo Base mais Deslocamento



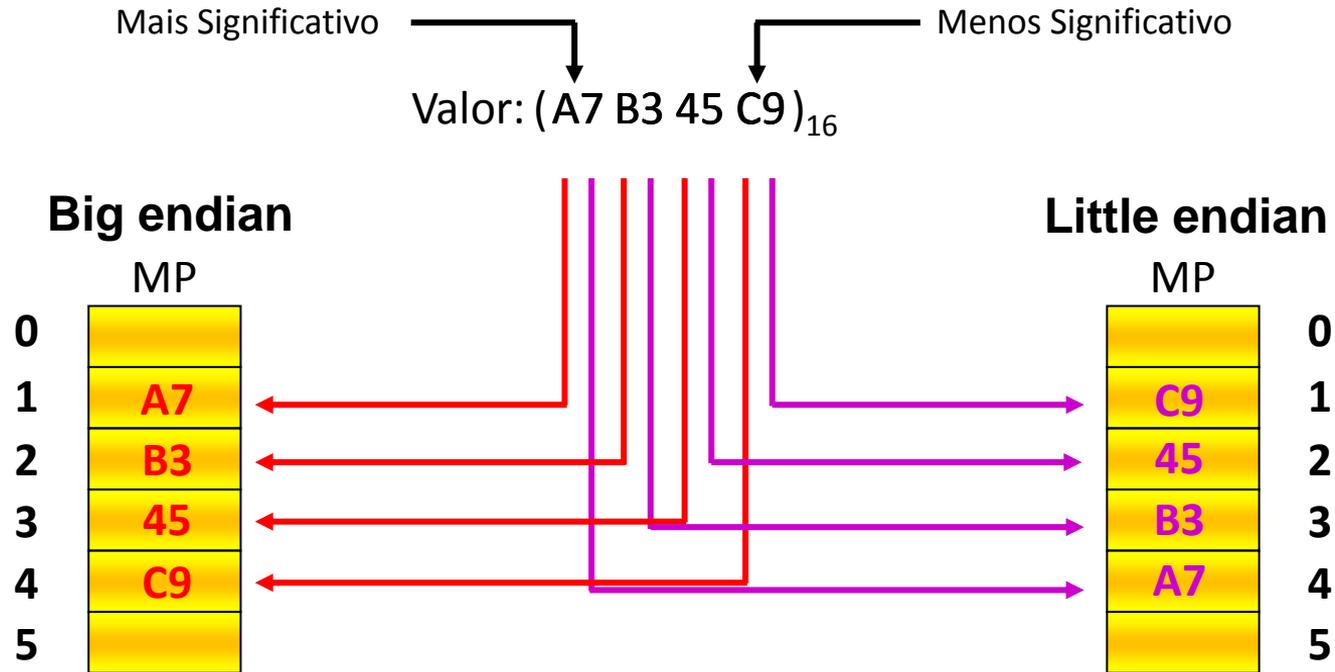
Arquitetura da CPU

Endians

- Quando o valor a ser guardado ou lido da memória ocupar mais de uma célula, o valor deve ser dividido em pedaços e então guardados nas células de forma sequencial.
- A questão envolvida aqui diz respeito ao sentido deste armazenamento. Este sentido é denominado endian.
- A informação pode ser armazenada a partir do byte mais significativo para o menos significativo, isto é, o **byte mais significativo** fica na célula de **endereço mais baixo** enquanto o **byte menos significativo** fica na célula de **endereço mais alto**. Este método é denominado **big endian**.
- Se for armazenado no sentido contrário, a partir do byte menos significativo para o mais significativo, isto é, o **byte menos significativo** fica na célula de **endereço mais baixo** enquanto o **byte mais significativo** fica na célula de **endereço mais alto**, este método será denominado **little endian**.
- A seguir temos um exemplo para cada formato.

Modos de Endereçamentos

Endians



Arquitetura da CPU

CISC X RISC

- CISC → Conjunto Complexo de Instruções.
 - Motivadores
 - Memória muito cara e conseqüentemente de baixa capacidade;
 - Lentidão no acesso à memória.
 - Características
 - Instruções que realizam tarefas complexas;
 - Uso de micro-instruções;
 - Possibilidade de trabalhar com muitos parâmetros;
 - Endereçamento de memória variado em instruções aritméticas e lógicas;
 - Instruções de tamanho variável.
 - Desvantagens
 - Circuitos muito complexos;
 - Grande geração de dependência de dados;
 - Descasamento nos tempos de acesso entre memória e registradores;
 - Maior dificuldade em criar arquiteturas mais eficientes.

Arquitetura da CPU

CISC X RISC

- RISC → Conjunto Reduzido de Instruções.
 - Motivadores
 - Uso da memória cache para diminuir o tempo médio no acesso à memória;
 - Barateamento da memória principal permitindo uma alta capacidade de armazenamento;
 - Interesse em simplificar a arquitetura da CPU para aumentar o desempenho com estruturas mais eficientes.
 - Características
 - As instruções realizam tarefas simples e básicas;
 - Uso de decodificação por hardware através de circuitos combinacionais, não são utilizadas micro instruções;
 - Endereçamento de memória variado, porém apenas para as instruções de leitura e escrita;
 - As operações aritméticas e lógicas só acessam dados em registradores;
 - Pequena variedade de Instruções.
 - Desvantagens
 - Maior complexidade na geração de programas;
 - Programas maiores.

Multiprocessadores

- A evolução tecnológica dos processadores iria diminuir drasticamente.
- O caminho para o aumento de desempenho é de unir mais de um processador para realizar a mesma tarefa em menos tempo através da divisão das tarefas entre os processadores.
- Os sistemas paralelos foram classificados em 4 categorias em função das instruções e dos dados. É a Taxonomia de Flynn.
 - **SISD** (*Single Instruction stream, Single Data stream*) – Uniprocessador.
 - **SIMD** (*Single Instruction stream, Multiple Data stream*) – A mesma instrução é executada por múltiplos processadores, porém cada um deles possui diferentes dados.
 - **MISD** (*Multiple Instruction stream, Single Data stream*) – Sem uso comercial.
 - **MIMD** (*Multiple Instruction stream, Multiple Data stream*) – Cada processador busca sua própria instrução e opera seu próprio dado.

Multiprocessadores

- Desafio do Processamento Paralelo.
- Para o aumento de desempenho com multiprocessadores nos deparamos com dois grandes desafios.
- O primeiro diz respeito ao limitado paralelismo disponível nos programas
- O segundo se refere ao alto custo nas comunicações.
- A limitação no paralelismo disponível dificulta em adquirirmos um bom ganho (*speedup*) nas máquinas paralelas.