

Arquitetura de Computadores 2 - Evolução

Prof. Dr. David Alain do Nascimento

Principais pontos

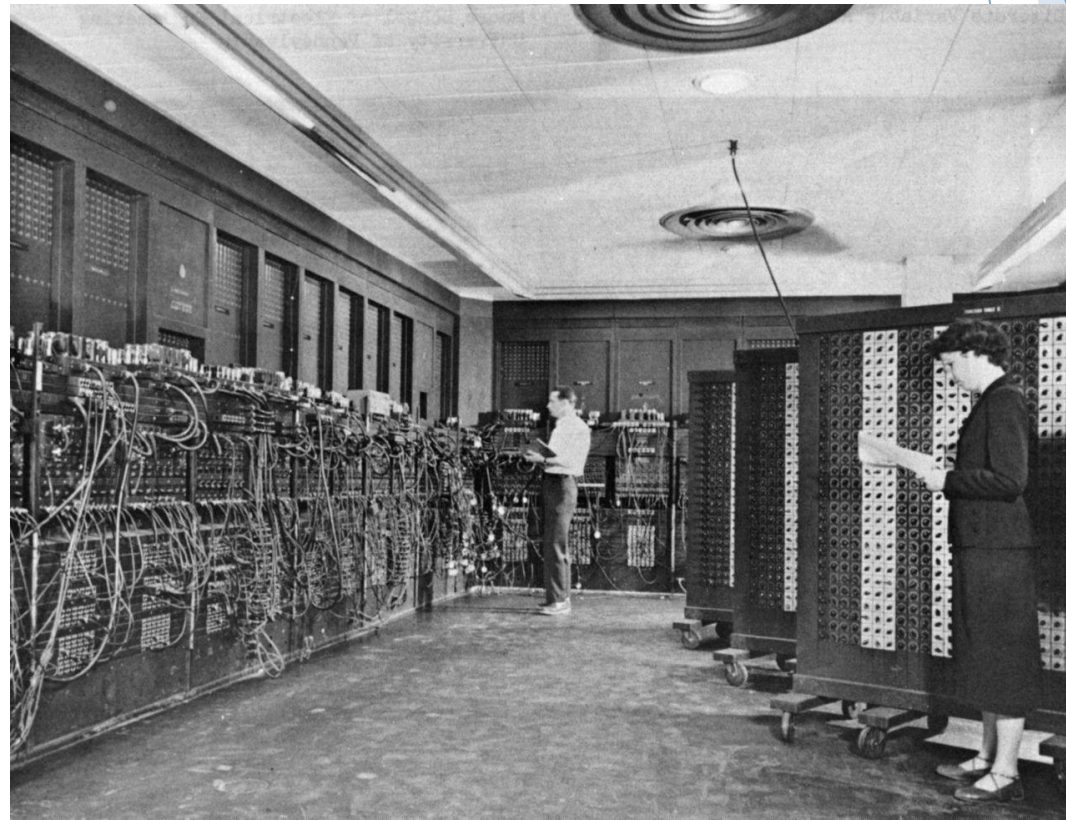
- ▶ A evolução dos computadores tem sido caracterizada pelo aumento na velocidade do processador, diminuição no tamanho do componente, aumento no tamanho da memória e aumento na capacidade e velocidade da E/S.
- ▶ Um fator responsável pelo grande aumento na velocidade do processador é o **encolhimento no tamanho dos componentes do microprocessador**; isso reduz a distância entre os componentes e, portanto, aumenta a velocidade.
- ▶ Contudo, os verdadeiros ganhos na velocidade nos anos recentes têm vindo da **organização do processador**, incluindo o uso intenso das técnicas de **pipeline** e **execução paralela** e do uso de técnicas de **execução especulativas** (tentativa de execução de instruções futuras que poderiam ser necessárias). Todas essas técnicas são projetadas para manter o processador ocupado pelo máximo de tempo possível.

Histórico dos computadores

- ▶ Primeira geração: válvulas
- ▶ Segunda geração: transistores
- ▶ Terceira geração: circuitos integrados
- ▶ Gerações posteriores

Primeira geração: válvulas

- ▶ O ENIAC (*Electronic Numerical Integrator And Computer*), projetado e construído na Universidade da Pensilvânia, foi o primeiro computador digital eletrônico de uso geral do mundo.



Primeira geração: válvulas

- ▶ O projeto foi uma resposta às necessidades dos EUA durante a Segunda Guerra Mundial. O *Ballistics Research Laboratory* (BRL) do Exército, uma agência responsável por desenvolver tabelas de faixa e trajetória para novas armas, estava tendo dificuldade para fornecer essas tabelas com precisão e dentro de um espaço de tempo razoável.
- ▶ Sem essas tabelas de disparo, as novas armas e artilharia eram inúteis aos artilheiros. O BRL empregou mais de 200 pessoas que, usando calculadoras de mesa, solucionavam as equações de balística necessárias. A preparação das tabelas para uma única arma exigiria muitas horas, até mesmo dias, de uma pessoa.

Primeira geração: válvulas

- ▶ John Mauchly, professor de engenharia elétrica na Universidade da Pensilvânia, e John Eckert, um de seus alunos formados, propôs construir um computador de uso geral usando válvulas para a aplicação do BRL.
- ▶ Em 1943, o Exército aceitou essa proposta, e foi iniciado o trabalho no ENIAC. A máquina resultante era enorme, pesava 30 toneladas, ocupando 1 500 pés quadrados (cerca de 140 metros quadrados) de superfície e contendo mais de 18 000 válvulas.
- ▶ Quando estava em operação, ela consumia 140 kilowatts de potência. Ela também era substancialmente mais rápida que qualquer computador eletromecânico, capaz de realizar 5 000 adições por segundo.

A MÁQUINA DE VON NEUMANN

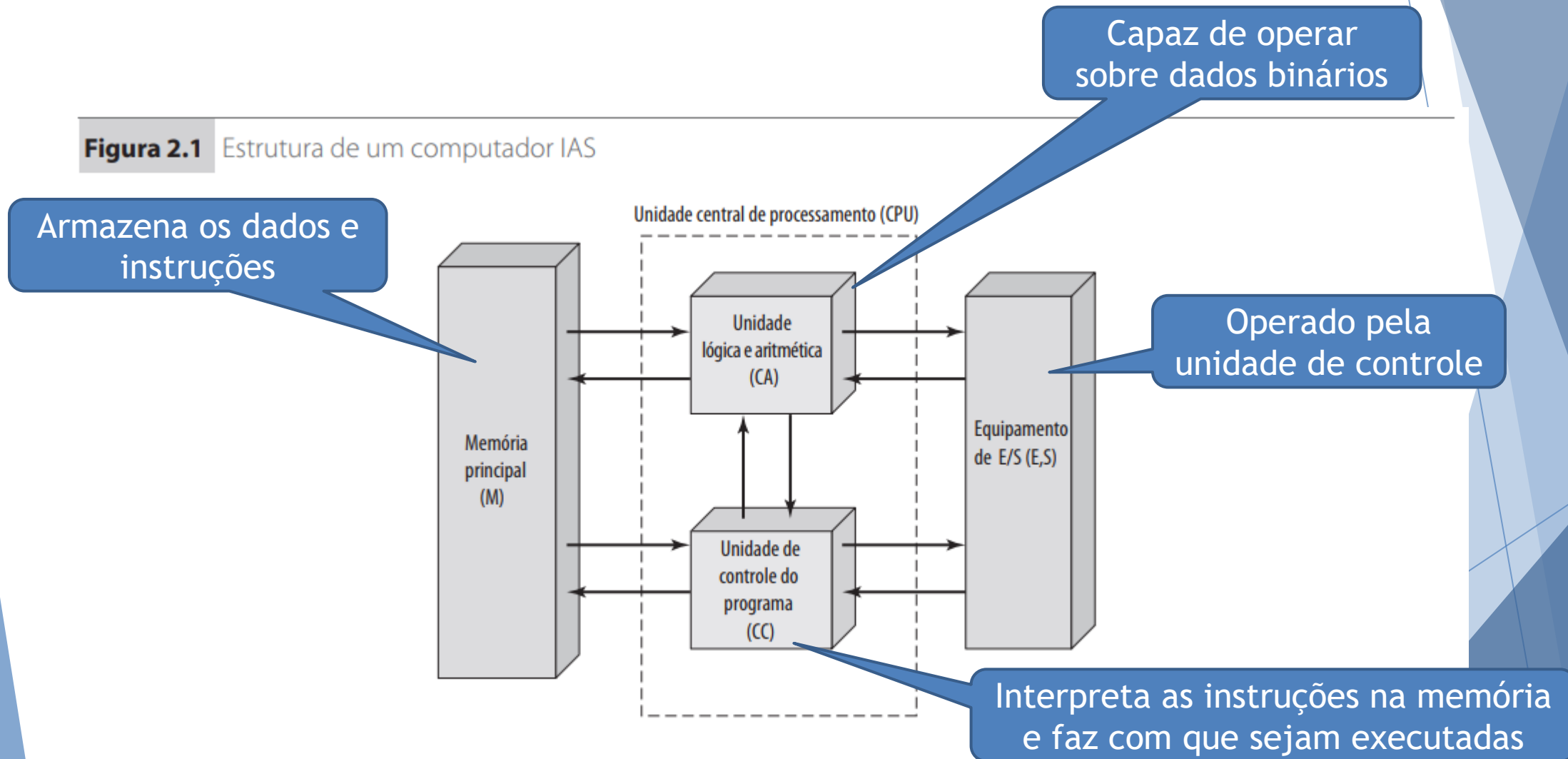
- ▶ A tarefa de entrar e alterar programas para o ENIAC era extremamente enfadonha. O processo de programação poderia ser facilitado se o programa pudesse ser representado em uma forma adequada para armazenamento na memória junto com os dados.
- ▶ **Um computador poderia obter suas instruções lendo-as da memória, e um programa poderia ser criado ou alterado definindo-se os valores de uma parte da memória.** Essa ideia, conhecida como conceito de **programa armazenado**, normalmente é atribuída aos projetistas do ENIAC, principalmente o matemático John von Neumann, que foi consultor no projeto ENIAC. Alan Turing desenvolveu a ideia praticamente ao mesmo tempo.

A MÁQUINA DE VON NEUMANN

- ▶ A primeira publicação da ideia foi em uma proposta de 1945 de von Neumann para um novo computador, o EDVAC (*Electronic Discrete Variable Computer*). Em 1946, von Neumann e seus colegas começaram o projeto de um novo computador de programa armazenado, conhecido como computador IAS, no *Princeton Institute for Advanced Studies*.
- ▶ O computador IAS, embora não concluído antes de 1952, é o protótipo de todos os computadores de uso geral.

A MÁQUINA DE VON NEUMANN

Figura 2.1 Estrutura de um computador IAS

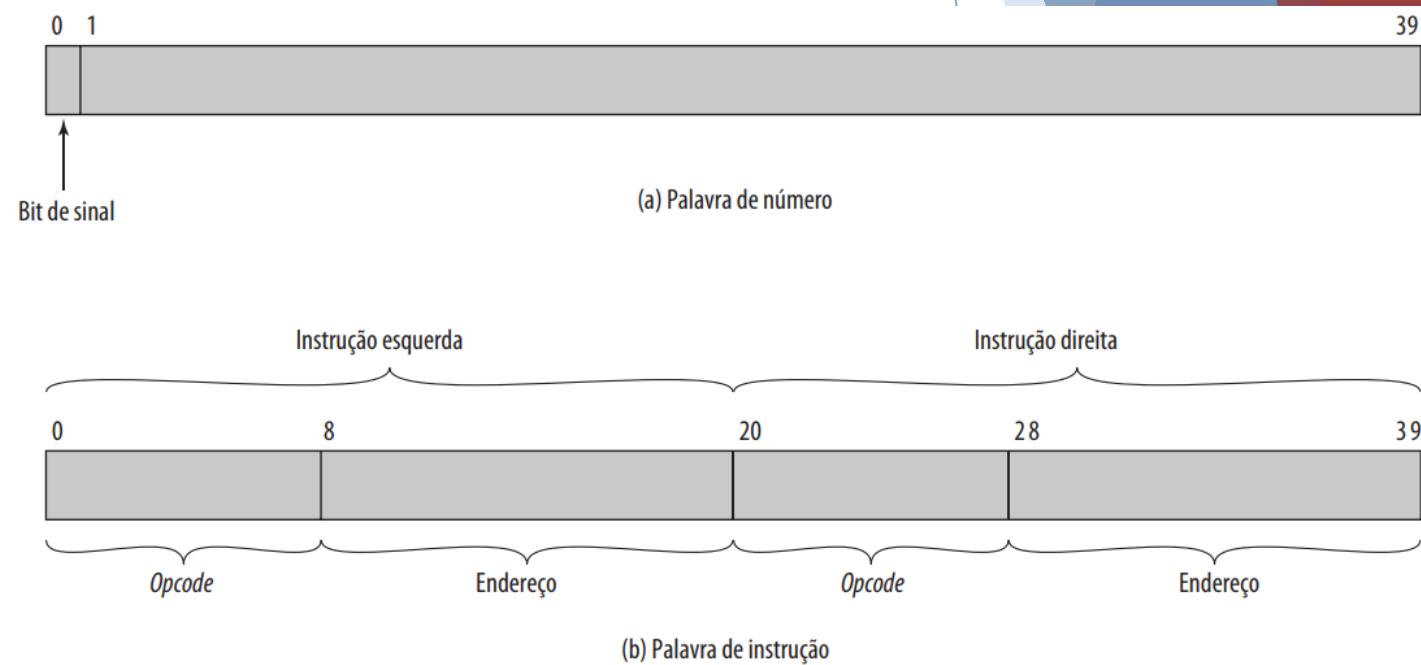


A MÁQUINA DE VON NEUMANN

- ▶ Com raras exceções, todos os computadores de hoje têm essa mesma estrutura e função geral, e são conhecidos como máquinas de von Neumann.
- ▶ Utilizaremos o computador IAS para explicar como funciona um computador.

Computador IAS

- ▶ A memória do IAS consiste em 1 000 locais de armazenamento, chamados *palavras* (*words*), de 40 dígitos binários (bits) cada.
- ▶ Tanto dados quanto instruções são armazenados lá. Números são representados em formato binário e cada instrução é um código binário.
- ▶ Cada número é representado por um bit de sinal e um valor de 39 bits. Uma palavra também pode conter duas instruções de 20 bits, com cada instrução consistindo em um código de operação de 8 bits (*opcode*), especificando a operação a ser realizada, e um endereço de 12 bits, designando uma das palavras na memória (numeradas de 0 a 999).



Computador IAS

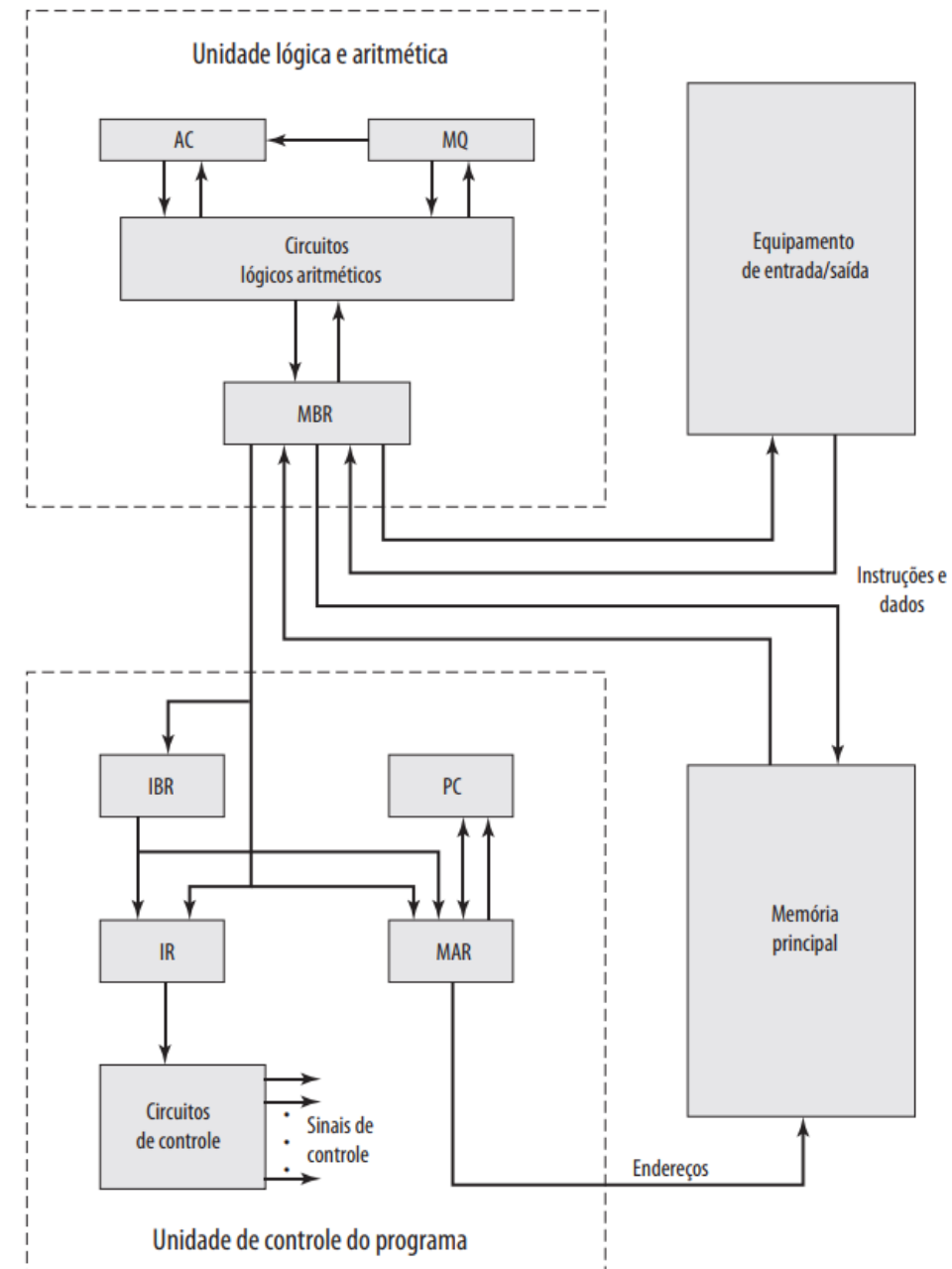
- ▶ A unidade de controle opera o IAS buscando instruções da memória e executando-as uma de cada vez.
- ▶ Sua unidade de controle e ALU contêm locais de armazenamento, chamados registradores, definidos da seguinte forma:
 - ▶ **Registrador de buffer de memória (MBR, do inglês *Memory Buffer Register*):** contém uma palavra a ser armazenada na memória ou enviada à unidade de E/S, ou é usada para receber uma palavra da memória ou de uma unidade de E/S.
 - ▶ **Registrador de endereço de memória (MAR, do inglês *Memory Address Register*):** especifica o endereço na memória da palavra a ser escrita ou lida no MBR.
 - ▶ **Registrador de instrução (IR, do inglês *Instruction Register*):** contém o *opcode* de 8 bits da instrução que está sendo executada.

Computador IAS

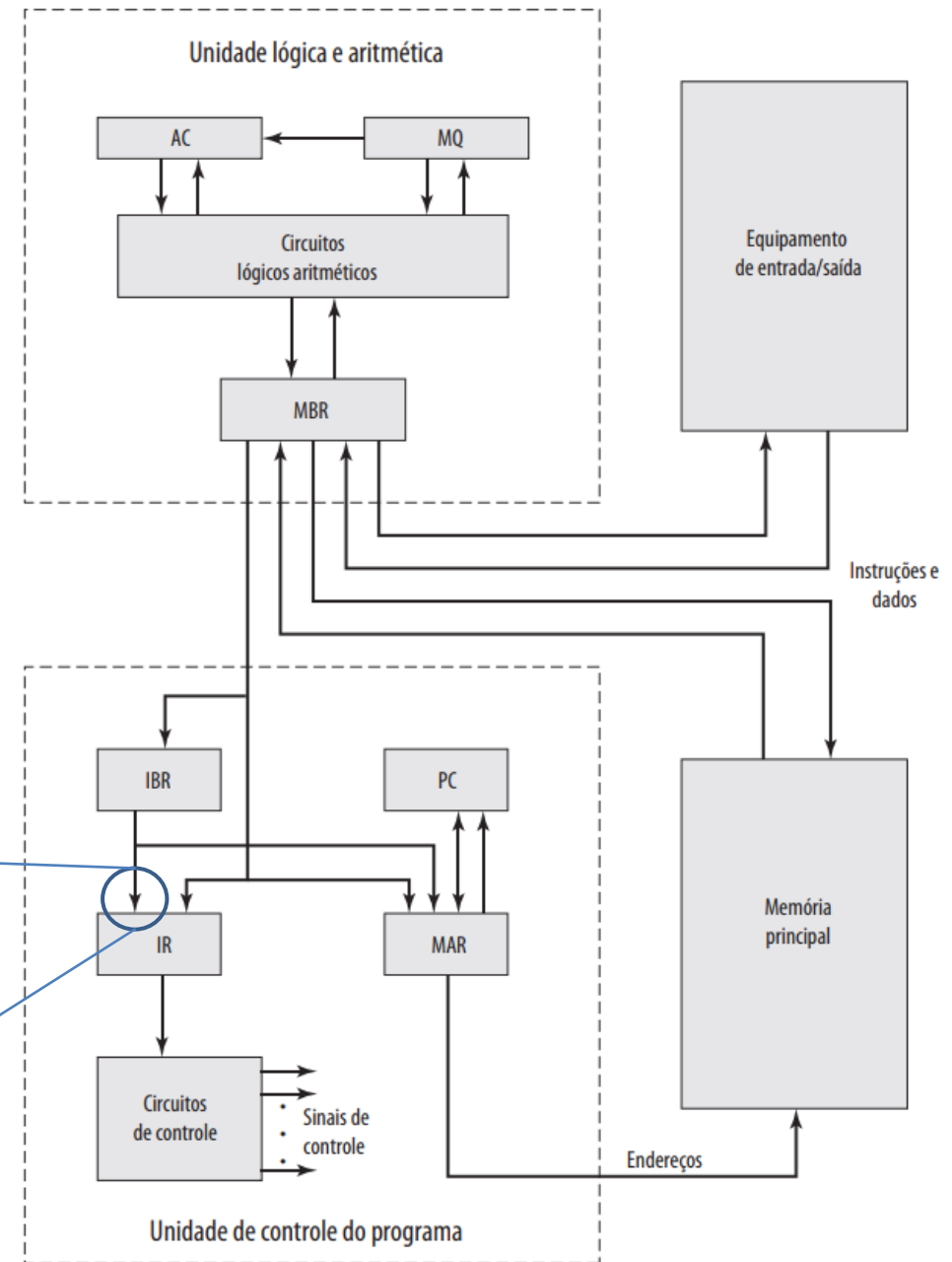
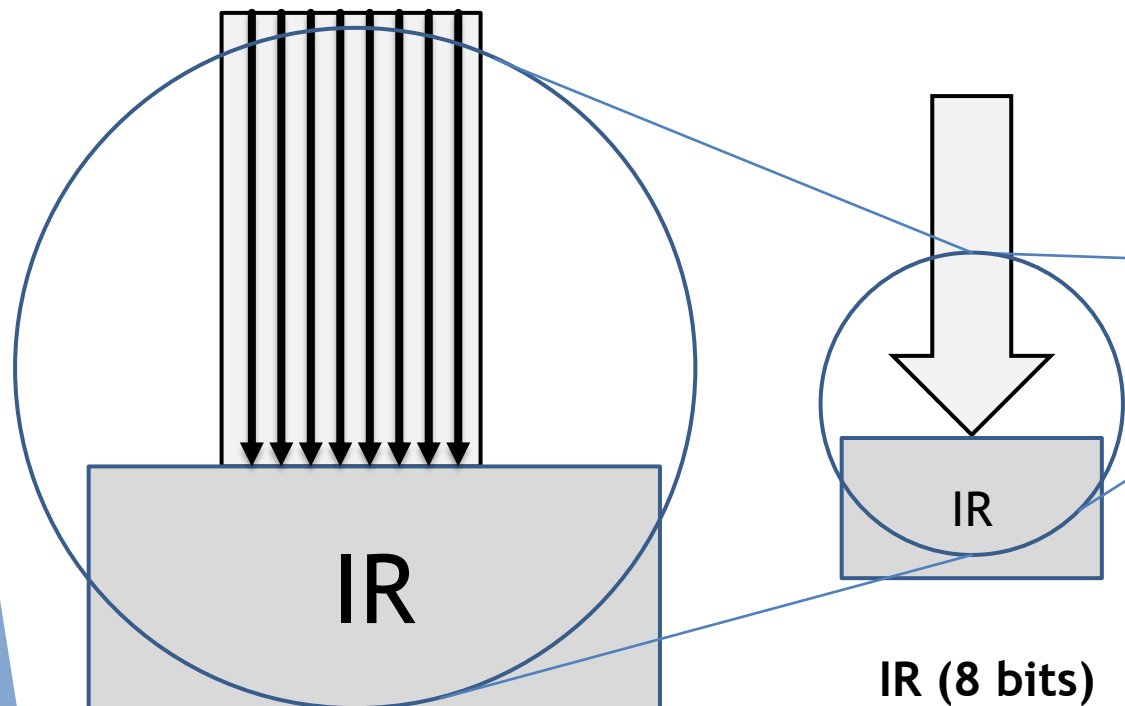
- ▶ **Registrador de buffer de instrução (IBR, do inglês *Instruction Buffer Register*):** empregado para manter temporariamente a próxima instrução a ser executada.
- ▶ **Contador de programa (PC, do inglês *Program Counter*):** contém o endereço do próximo par de instruções a ser apanhado da memória.
- ▶ **Acumulador (AC) e quociente multiplicador (MQ, do inglês *Multiplier Quotient*):** empregado para manter temporariamente operandos e resultados de operações da ALU. Por exemplo, o resultado de multiplicar dois números de 40 bits é um número de 80 bits; os 40 bits mais significativos são armazenados no AC e o menos significativos no MQ.

Computador IAS

- ▶ **MBR (40 bits):** Contém uma palavra a ser armazenada na memória ou enviada à unidade de E/S, ou é usado para receber uma palavra da memória ou de uma unidade de E/S.
- ▶ **MAR (12 bits):** Especifica o endereço na memória da palavra a ser escrita no ou lida do MBR.
- ▶ **IR (8 bits):** Contém o *opcode* de 8 bits da instrução que está sendo executada.
- ▶ **IBR (20 bits):** Usado para manter temporariamente a próxima instrução a ser executada.
- ▶ **PC (12 bits):** Contém o endereço do próximo par de instruções a ser apanhado da memória.
- ▶ **AC (40 bits) e MQ (40 bits):** Utilizados para manter temporariamente operandos e resultados de operações da ALU.



Computador IAS



Computador IAS

- ▶ O IAS opera realizando repetidamente um *ciclo de instrução*.
- ▶ Cada ciclo de instrução consiste em dois subciclos.
- ▶ Durante o *ciclo de busca (fetch cycle)*, o *opcode* da próxima instrução é carregado no **IR** e a parte de endereço é carregada no **MAR**. Essa instrução pode ser retirada do **IBR** ou pode ser obtida da memória carregando-se uma palavra no **MBR**, e depois para o **IBR**, **IR** e **MAR**.
 - ▶ **IR (8 bits)**: *Opcode* da instrução que está para ser executada.
 - ▶ **MAR (12 bits)**: Endereço de memória para leitura ou escrita de/para fora da CPU (ou seja, memória RAM ou E/S).
 - ▶ **MBR (40 bits)**: Palavra (conteúdo) para leitura ou escrita de/para fora da CPU (ou seja, de/para memória ou E/S).
 - ▶ **IBR (20 bits)**: Próxima instrução a ser executada.

Computador IAS

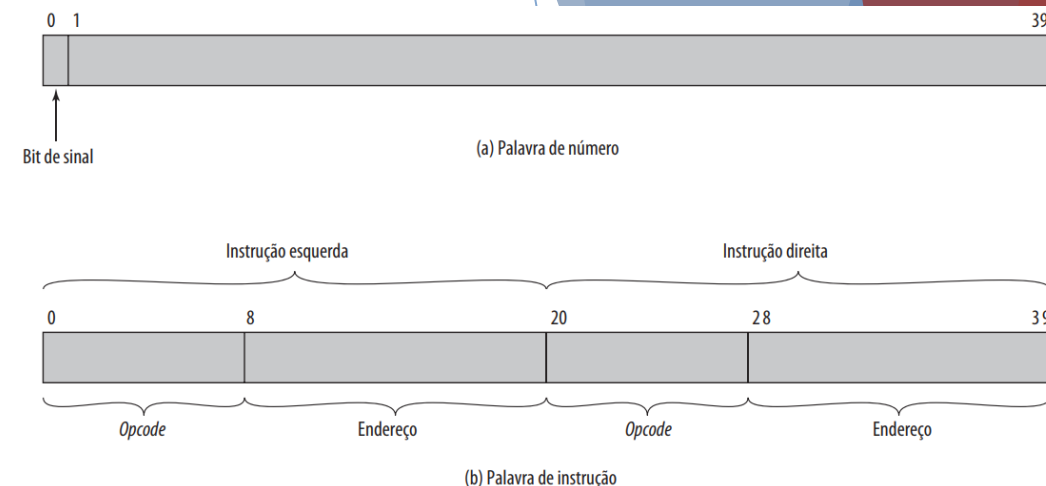
- Modelo de memória do IAS

Leitura de valor da memória:

```
reg <- M(X)
```

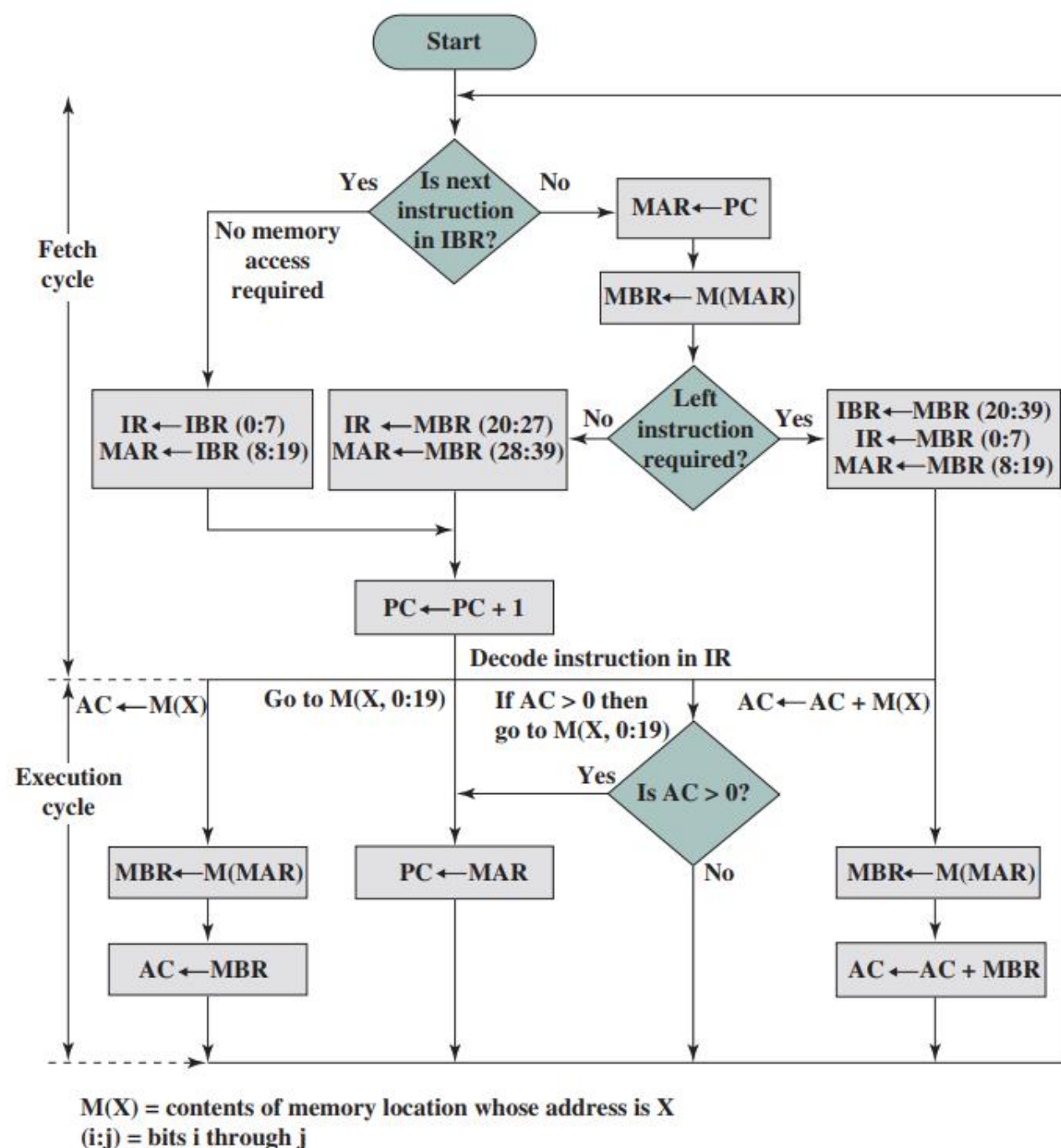
reg recebe o valor lido da memória RAM
que está armazenado no endereço **X**

Exemplo:

[illegible][illegible]

Computador IAS

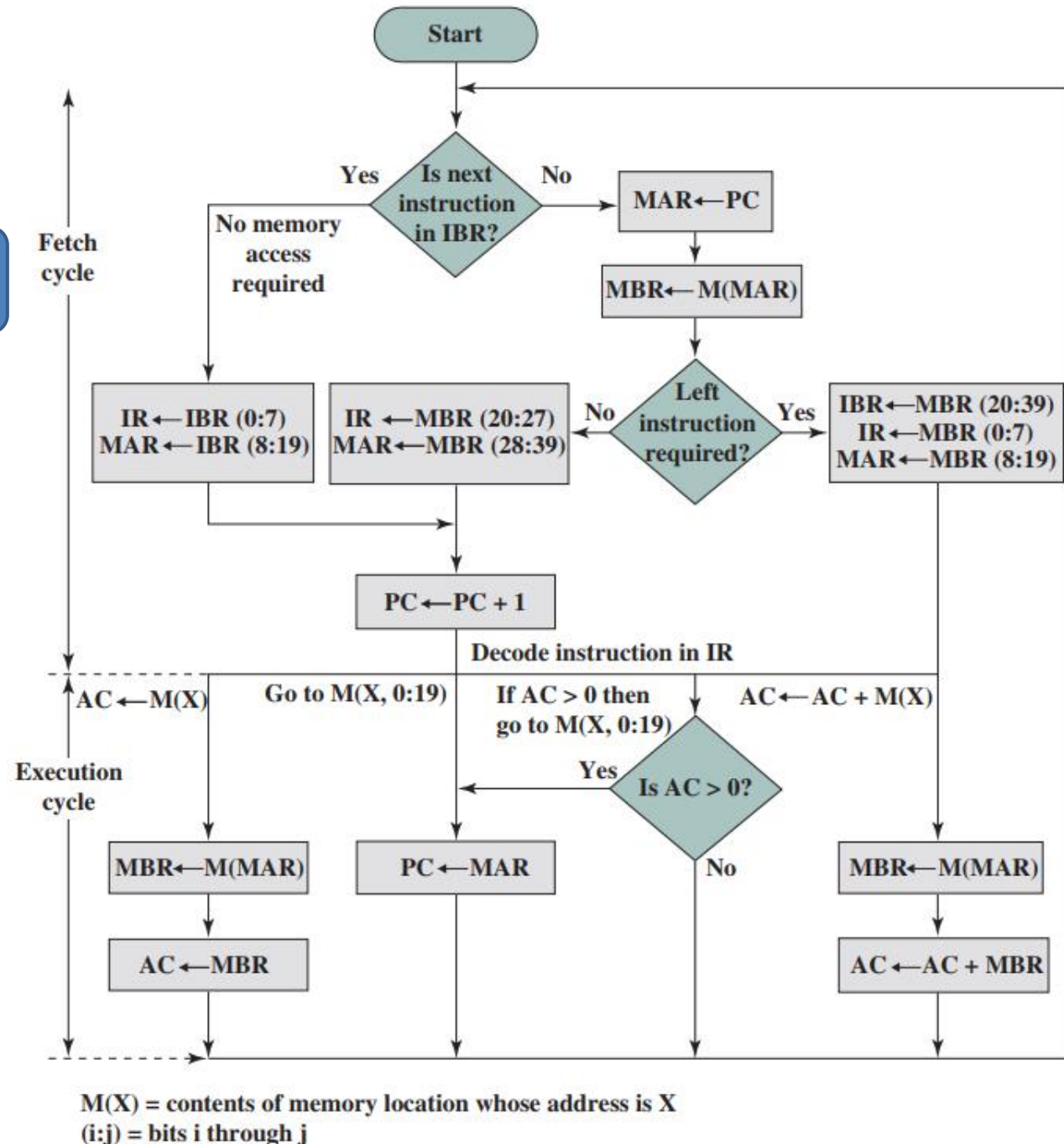
- **MBR (40 bits):** Contém uma palavra a ser armazenada na memória ou enviada à unidade de E/S, ou é usado para receber uma palavra da memória ou de uma unidade de E/S.
- **MAR (12 bits):** Especifica o endereço na memória da palavra a ser escrita ou lida no MBR.
- **IR (8 bits):** Contém o *opcode* de 8 bits da instrução que está sendo executada.
- **IBR (20 bits):** Usado para manter temporariamente a próxima instrução a ser executada.
- **PC (12 bits):** Contém o endereço do próximo par de instruções a ser apanhado da memória.
- **AC (40 bits) e MQ (40 bits):** Utilizados para manter temporariamente operandos e resultados de operações da ALU.



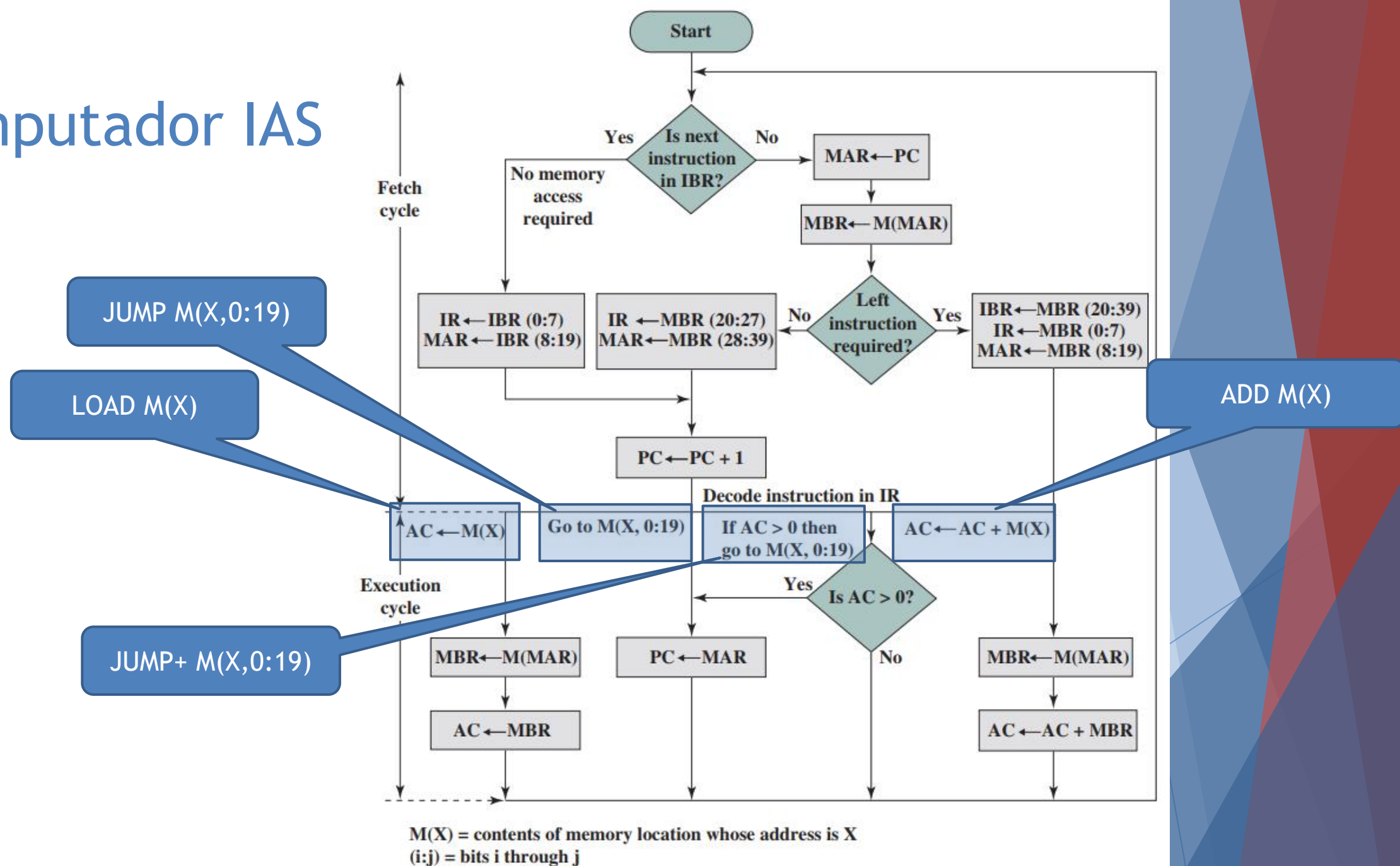
Computador IAS

Indireção: Representação de acesso ao conteúdo da memória utilizando-se endereço armazenado em registrador: $M(X)$

- ▶ Por que a indireção? Essas operações são controladas por circuitos eletrônicos e resultam no uso de caminhos de dados. Para simplificar a eletrônica, somente um registrador é usado para especificar o endereço na memória para uma leitura ou escrita e somente um registrador é usado para a origem ou o destino.
- ▶ Quando o *opcode* está no **IR**, o *ciclo de execução* é realizado. O circuito de controle interpreta o *opcode* e executa a instrução enviando os sinais de controle apropriados para que os dados sejam movidos ou uma operação seja realizada pela ALU.



Computador IAS



Computador IAS

- ▶ O computador IAS tinha um total de 21 instruções que podem ser agrupadas da seguinte forma:
 - ▶ **Transferência de dados:** movem dados entre memória e registradores da ALU ou entre dois registradores da ALU.
 - ▶ **Desvio incondicional:** normalmente, a unidade de controle executa instruções em sequência a partir da memória. Essa sequência pode ser alterada por uma instrução de desvio, que facilita operações repetitivas.
 - ▶ **Desvio condicional:** o desvio pode se tornar dependente de uma condição, permitindo assim pontos de decisão.
 - ▶ **Aritméticas:** operações realizadas pela ALU.
 - ▶ **Modificação de endereço:** permite que os endereços sejam calculados na ALU e depois inseridos em instruções armazenadas na memória. Isso permite a um programa uma flexibilidade de endereçamento considerável.

Computador IAS

► Instruções do IAS

Tipo de instrução	Opcode	Representação simbólica	Descrição
Transferência de dados	00001010	LOAD MQ	Transfere o conteúdo de MQ para AC
	00001001	LOAD MQ,M(X)	Transfere o conteúdo do local de memória X para MQ
	00100001	STOR M(X)	Transfere o conteúdo de AC para o local de memória X
	00000001	LOAD M(X)	Transfere M(X) para o AC
	00000010	LOAD – M(X)	Transfere – M(X) para o AC
	00000011	LOAD M(X)	Transfere o valor absoluto de M(X) para o AC
	00000100	LOAD – M(X)	Transfere - M(X) para o acumulador

Computador IAS

► Instruções do IAS

Tipo de instrução	Opcode	Representação simbólica	Descrição
Desvio incondicional	00001101	JUMP M(X,0:19)	Apanha a próxima instrução da metade esquerda de M(X)
	00001110	JUMP M(X,20:39)	Apanha a próxima instrução da metade direita de M(X)
Desvio condicional	00001111	JUMP+ M(X,0:19)	Se o número no AC for não negativo, apanha a próxima instrução da metade esquerda de M(X)
	00010000	JUMP+ M(X,20:39)	Se o número no AC for não negativo, apanha a próxima instrução da metade direita de M(X)

Computador IAS

► Instruções do IAS

Tipo de instrução	Opcode	Representação simbólica	Descrição
Aritmética	00000101	ADD M(X)	Soma M(X) a AC; coloca o resultado em AC
	00000111	ADD M(X)	Soma M(X) a AC; coloca o resultado em AC
	00000110	SUB M(X)	Subtrai M(X) de AC; coloca o resultado em AC
	00001000	SUB M(X)	Subtrai M(X) de AC; coloca o resto em AC
	00001011	MUL M(X)	Multiplica M(X) por MQ; coloca os bits mais significativos do resultado em AC; coloca bits menos significativos em MQ
	00001100	DIV M(X)	Divide AC por M(X); coloca o quociente em MQ e o resto em AC
	00010100	LSH	Multiplica o AC por 2; ou seja, desloca à esquerda uma posição de bit
	00010101	RSH	Divide o AC por 2; ou seja, desloca uma posição à direita

Computador IAS

► Instruções do IAS

Tipo de instrução	Opcode	Representação simbólica	Descrição
Modificação de endereço	00010010	STOR M(X,8:19)	Substitui campo de endereço da esquerda em M(X) por 12 bits mais à direita de AC
	00010011	STOR M(X,28:39)	Substitui campo de endereço da direita em M(X) por 12 bits mais à direita de AC