

# Conjunto de Instruções

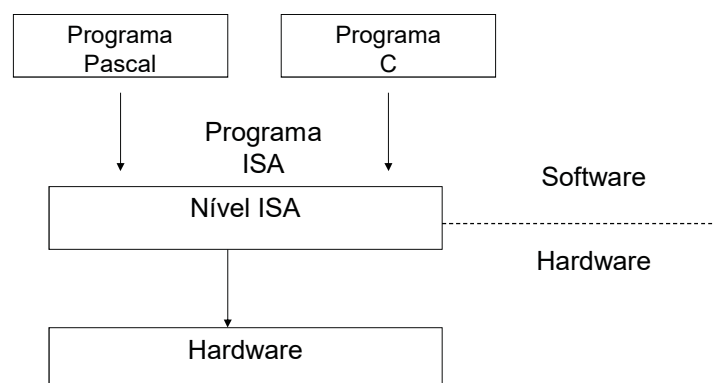
*Prof. Edson Pedro Ferlin*

- Objetivos
  - Estudar o Conjunto de Instruções dos processadores
- Conteúdos
  - *Instruction Set Architecture (ISA)*
  - Instruções
  - Endereçamento
  - Conjunto de Instruções
  - Fluxo de Controle

## Definições

- Nível ISA (*Instruction Set Architecture*).
- Está posicionado entre o nível da microarquitetura e o nível do sistema operacional.
- É a interface entre o software e o hardware.
- Nesse nível está definida a interface entre os compiladores e o hardware.

## Nível ISA (*Instruction Set Architecture*)



## Compatibilidade

- Manter a compatibilidade com as máquinas anteriores.
- O desafio é construir máquinas melhores mantendo a compatibilidade com os modelos anteriores.
- Projetos equivalentes, ISAs diferentes podem ser responsáveis por diferenças de até 25% na performance.
- As instruções do nível ISA são aquelas para as quais o compilador deve gerar código.
- Dois modos de execução: modo Kernel (supervisor) e modo usuário.

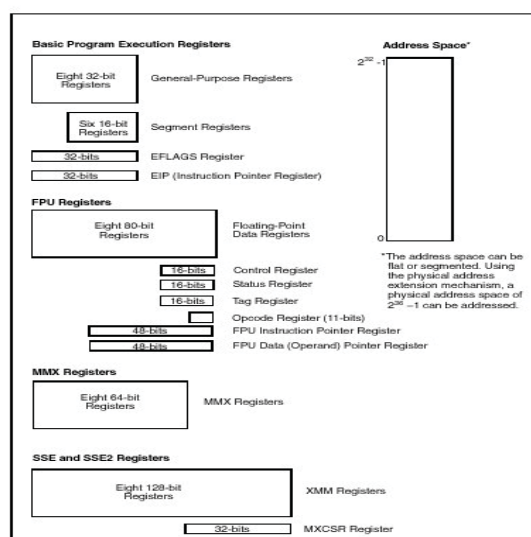
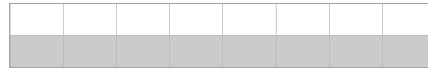


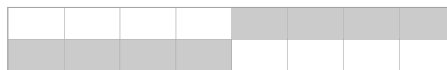
Figure 3-1. IA-32 Basic Execution Environment

## Modelos de Memória

- Conjunto de células que têm endereços consecutivos.
- Palavras alinhadas.



- Palavras arbitrárias necessita de um hardware extra.



- Único espaço linear de 0 a n.
- Espaço de endereçamento separado para dados e instruções (arquitetura harvard)

## Registradores IA-32

- The 32-bit general-purpose registers (EAX, EBX, ECX, EDX, ESI, EDI, ESP, or EBP).
- The 16-bit general-purpose registers (AX, BX, CX, DX, SI, DI, SP, or BP).
- The 8-bit general-purpose registers (AH, BH, CH, DH, AL, BL, CL, or DL).
- The segment registers (CS, DS, SS, ES, FS, and GS).
- The EFLAGS register.
- The x87 FPU registers (ST0 through ST7, status word, control word, tag word, data operand pointer, and instruction pointer).
- The MMX registers (MM0 through MM7).
- The XMM registers (XMM0 through XMM7) and the MXCSR register.
- The control registers (CR0, CR2, CR3, CR4) and system table pointer registers (GDTR, LDTR, IDTR, task register).
- The debug registers (DR0, DR1, DR2, DR3, DR6, DR7).
- The MSR registers.

## Tipos de Dados

- Todos os computadores precisam de dados.
- É necessário representá-lo dentro do computador.
- Isto acontece devido à existência de instruções apropriadas.
- Tipos: Numérico (Inteiro e Ponto Flutuante), Não-Numéricos (Caracteres, booleano) e Ponteiro.

## Formato de Instruções

OPCODE
--------

Ex: NOP

OPCODE	Endereço
--------	----------

Ex: INC A

OPCODE	Endereço 1	Endereço 2
--------	------------	------------

Ex: MOV A,B

OPCODE	Endereço 1	Endereço 2	Endereço 3
--------	------------	------------	------------

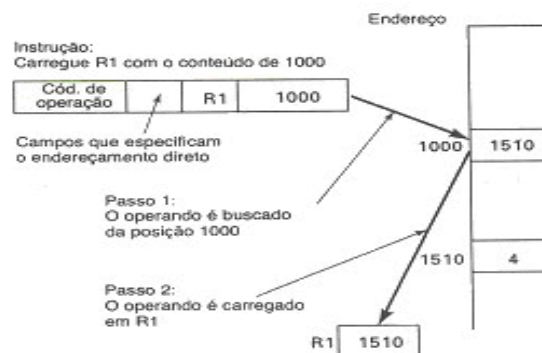
Ex: ADD A,B,C

## Formato de Instruções

- Uma instrução é composta obrigatoriamente por um código de operação e, em geral, por algumas outras informações a respeito da fonte e do destino de seus operandos
- Código da operação para informar ao hardware o que deve ser feito
- A especificação de onde estão os operandos (ou seja, os seus endereços) é conhecida como endereçamento.

## Endereçamento

**Direto** – o endereço está contido no campo do operando

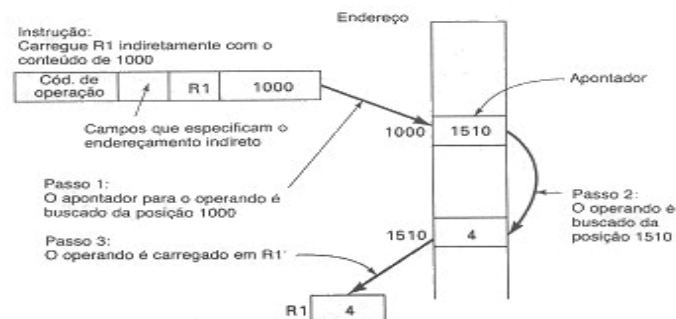


## Modos de Endereçamento - Direto

- INERENTE ou IMPLÍCITO
  - o próprio OP-CODE identifica o endereço. Ex: RET ou RETI
- REGISTRADOR
  - o operando especifica um registrador. Ex: ADD A,Rn
- ABSOLUTO
  - o operando referencia a memória: Ex: INC 20h
- IMEDIATO
  - o operando contém o próprio dado. Ex: MOV A,#20h

## Endereçamento

**Indireto** – o endereço está contido em uma posição especificada em um operando





## Modos de Endereçamento - Indireto

- RELATIVO

- o endereço é calculado pela adição do conteúdo do PC com um deslocamento contido no campo operando: EX: `MOVC A,@A+PC`

- PILHA

- utiliza como referência a estrutura lógica pilha. EX: `PUSH`, `POP`

- INDEXADO

- o endereço é calculado pela adição do conteúdo de um registrador com o endereço-base contido no campo operando. Ex: `MOV A,@A+[20h]`

- BASEADO

- o endereço é calculado pela adição do conteúdo de um registrador-base com o deslocamento contido no campo operando. Ex: `MOV A,@A+DPTR`



## Tipos de Instruções

- MOVIMENTO de DADOS: Operação fundamental.

- Ex: `MOV A,B`

- OPERAÇÕES DIÁDICAS: Produzem um resultado com base de dois operandos.

- Ex: `ADD A,B`

- OPERAÇÕES MONÁDICAS: São aquelas que têm apenas um operando e produzem um resultado.

- Ex: `INC A`

- COMPARAÇÕES e DESVIOS: Testam os dados e alteram a sequência de execução de suas instruções com base no resultado desses testes.

- Ex: `JC Teste`



## Tipos de Instruções

- **CHAMADA de PROCEDIMENTO:** Grupo de instruções que podem ser chamadas a qualquer momento dentro do programa. Cuidado com o Recursivo.

Ex: CALL Teste

- **CONTROLE de LAÇO:** Repetir a execução de um grupo de instruções um número fixo de vezes.

Ex: DJNZ A, Teste

- **ENTRADA/SAÍDA:** Três modalidades: E/S programada com espera ocupada; E/S dirigida por interrupção; E/S com acesso direto à memória (DMA)

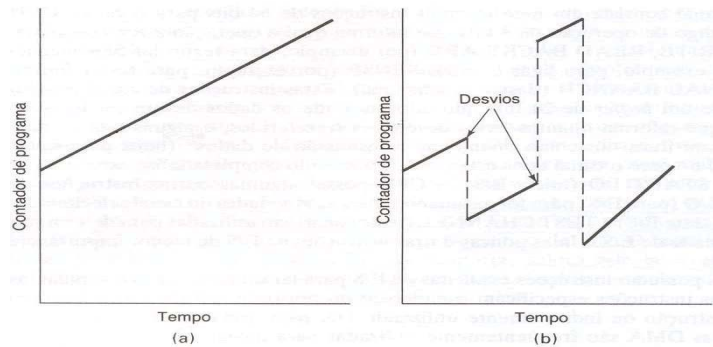
Ex: IN A, OUT A

## Fluxo de Controle

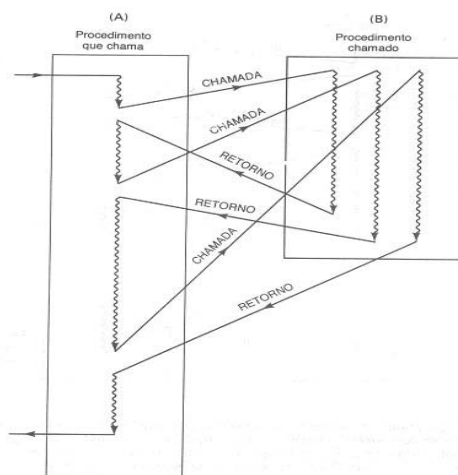
A sequência na qual as instruções são dinamicamente executadas, ou seja, a ordem na qual instruções são executadas no decorrer da execução de um programa.

## Execução Sequencial e Desvios

A maioria das instruções de uma máquina não altera o fluxo de controle.



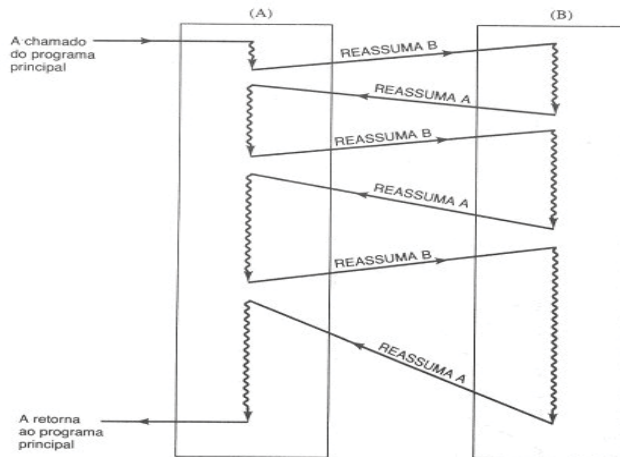
## Procedimentos



É a técnica mais importante disponível para a estruturação de programas.

Procedimento Recursivo.

## Co-Rotinas



São rotinas trabalhando de forma intercalada.

## Traps

São chamadas de procedimento automática iniciada sempre que ocorrer alguma condição específica causada pela execução de um programa.

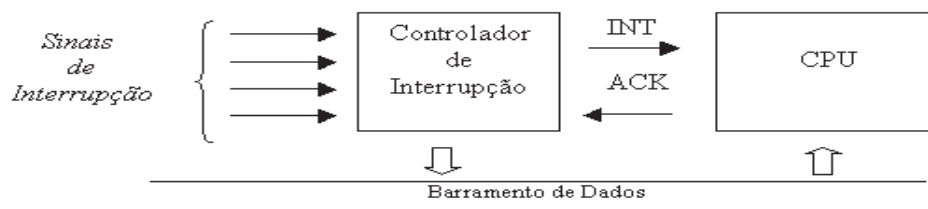
Exemplo: Overflow

## Interrupções

São modificações no fluxo de controle de um programa causadas por um evento externo ao processamento do programa, usualmente eventos relacionados a operações de E/S.

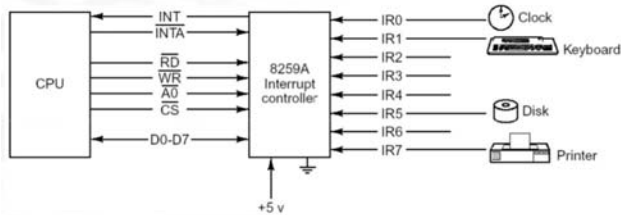
Dois componentes: Hardware e Software.

## Interrupções



Quando ocorre um sinal em uma linha de interrupção o controlador de interrupção (INT) do processador e após obter a permissão da CPU (ACK-Acknowledge) coloca o endereço de interrupção no barramento de dados, para ser lido pelo processador de posse deste endereço, o processador, via um processamento interno, executará a rotina de interrupção.

## Interrupção – IRQ (*Interrupt ReQuests*)



25

Conjunto de Instruções

Prof. Edson Pedro Ferlin

## Sequência de Atendimento das Interrupções



- Salva PC
- Salva PSW (*Program Status Word*)
- Desvia para a rotina
  - Rotina
  - Reti (*Return of Interrupt*)
- Restaura PC
- Restaura PSW

26

Conjunto de Instruções

Prof. Edson Pedro Ferlin

## Atividade

- Resolver os exercícios.

