

# MC404

---

## ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

2006

Prof. Paulo Cesar Centoducatte

[ducatte@ic.unicamp.br](mailto:ducatte@ic.unicamp.br)

[www.ic.unicamp.br/~ducatte](http://www.ic.unicamp.br/~ducatte)

# MC404

---

## ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

**“Processador INTEL 80X86”**

# Processador Intel 8086

## Sumário

---

- **Unidades do 8086**
- **Organização**
- **Registradores**
  - **Propósito Geral**
  - **Propósito Específico**
- **Gerenciamento de Memória**
- **Organização de Memória - DOS**
- **Interrupção**

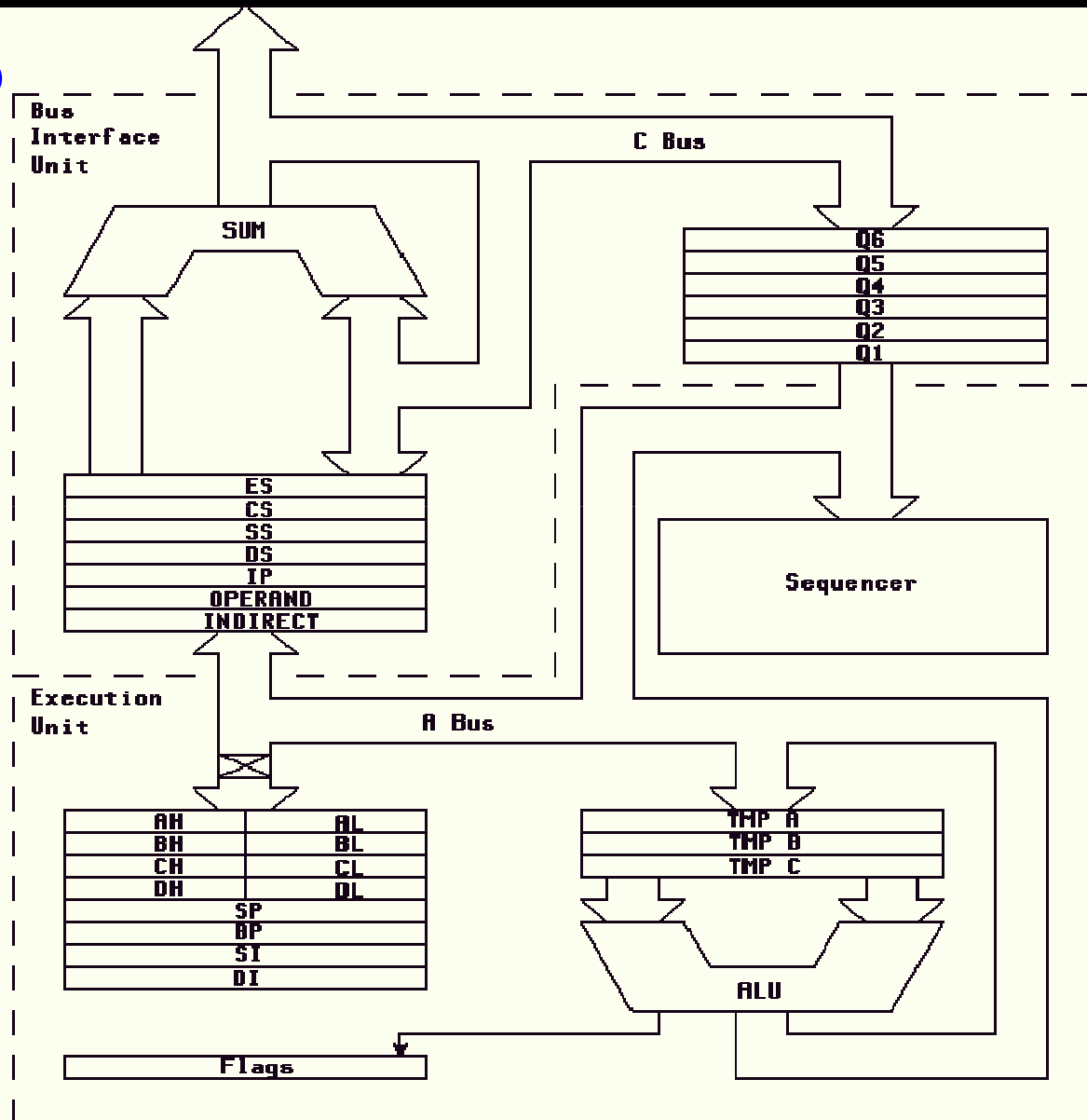
# Processador INTEL 80X86

---

- O 8086 divide-se internamente em duas unidades:
  - **Execution Unit (EU) - Unidade de Execução:**
    - ULA - realiza operações aritméticas de +, -, \*, / e operações lógicas AND, OR, NOT, XOR;
    - Registradores para armazenamento temporário durante as operações, que são endereçados por nome.
  - **BUS Interface Unit (BIU) - Unidade de Interface de Barramento:**
    - Faz a comunicação de dados entre a **EU** e o meio externo (memória, E/S);
    - Controla a transmissão de sinais de endereços, dados e controle;
    - Controla a sequência de busca e execução de instruções;
    - Mecanismo de **pre-fetch**: busca até 6 instruções futuras deixando-as na fila de instruções (**instruction queue**) → aumento de velocidade.

# Processador INTEL 80X86

## Organização



# Processador INTEL 80X86

---

- **Registradores:**

- de propósito geral ou de dados;
- de endereços (segmentos, apontadores e índices);
- sinalizadores de estado e controle (FLAGS);

- **Registradores de propósito geral (de dados):**

- AX, BX, CX e DX
  - são todos registradores de 16 bits
  - utilizados nas operações aritméticas e lógicas
  - podem ser usados como registradores de 16 ou 8 bits:

- AH e AL
- BH e BL
- CH e CL
- DH e DL

} 8 registradores  
de 8 bits cada

"H" → byte alto ou superior

"L" → byte baixo ou inferior

# Processador INTEL 80X86

---

- Registradores de propósito geral (de dados):
  - **AX (acumulador)** → utilizado como acumulador em operações aritméticas e lógicas; em instruções de E/S, ajuste decimal, conversão etc
  - **BX (base)** → usado como registrador BASE para referenciar posições de memória; BX armazena o endereço BASE de uma tabela ou vetor de dados, a partir do qual outras posições são obtidas adicionando-se um valor de deslocamento (*offset*).

# Processador INTEL 80X86

---

- Registradores de propósito geral (de dados):
  - **CX (contador)** → utilizado em operações iterativas e repetitivas para contar bits, bytes ou palavras, podendo ser incrementado ou decrementado; **CL** funciona como um contador de 8 bits.
  - **DX (dados)** → utilizado em operações de multiplicação para armazenar parte de um produto de 32 bits, ou em operações de divisão, para armazenar o resto; **utilizado em operações de E/S para especificar o endereço de uma porta de E/S.**



# Processador INTEL 80X86

---

- Registradores de Segmentos:

- CS, DS, SS e ES

- são todos registradores de 16 bits
- o endereçamento no 8086 (20 bits) é diferenciado para:
  - código de programa (instruções) → CS
  - Dados → DS e ES
  - Pilhas → SS

# Processador INTEL 80X86

---

- **Registradores de Segmentos:**

- segmento: é um bloco de memória de 64 KBytes, endereçável.
- durante a execução de um programa no 8086, há 4 segmentos ativos:
  - segmento de código endereçado por **CS**
  - segmento de dados endereçado por **DS**
  - segmento de pilha endereçado por **SS** (stack segment)
  - segmento extra (de dados) endereçado por **ES**

# Processador INTEL 80X86

---

- Registrador Apontador de Instrução:
  - **IP (*instruction pointer*)**: utilizado em conjunto com CS para localizar a posição, dentro do segmento de código corrente, da próxima instrução a ser executada;
  - IP é automaticamente incrementado em função do número de bytes da instrução executada.
  - **Observação: o IP é o PC (program counter) do INTEL 80x86**

# Processador INTEL 80X86

---

- **Registradores Apontador de Pilha e de Índice:**
  - Armazenam valores de deslocamento de endereços (*offset*), a fim de acessar regiões da memória muito utilizadas:
    - pilha,
    - blocos de dados,
    - *arrays e strings*.

# Processador INTEL 80X86

- **Registradores Apontador de Pilha e de índice:**
  - **SP** (*stack pointer - apontador de pilha*) é utilizado em conjunto com SS, para acessar a área de pilha na memória; aponta para o topo da pilha - atualizado automaticamente.
  - **BP** (*base pointer - apontador base*) é o ponteiro que, em conjunto com SS, permite acesso a dados dentro do segmento de pilha.
  - **SI** (*source index - índice fonte*) usado como registrador índice em alguns modos de endereçamento indireto, em conjunto com DS.
  - **DI** (*destination index - índice destino*) similar ao SI, atuando em conjunto com ES.
- Obs: SI e DI facilitam a movimentação de dados armazenados em seqüência entre posições **fonte** (indicado por SI) e posições **destino** (indicado por DI).

# Processador INTEL 80X86

---

- Registrador de Sinalizadores (FLAGS):
  - indica o estado do microprocessador após a execução de cada instrução;
  - conjunto de bits individuais, cada qual indicando alguma propriedade;
  - subdividem-se em: FLAGS da **estado** (*status*) e FLAGS de **controle**.
  - organização:
    - 1 registrador de 16 bits;
    - 6 FLAGS de estado;
    - 3 FLAGS de controle;
    - 7 bits não utilizados (sem função);

# Processador INTEL 80X86

- **Registrador de Sinalizadores (FLAGS):**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		OF	DF	IF	TF	SF	ZF		AF		PF		CF		

- **Flags de Estados:**

- **CF - Flag de Carry**

- CF = 1 → após instruções de soma que geram "vai um"; após instruções de subtração que geram "emprestimo" ("empresta um");
    - CF = 0 → caso contrário.

- **PF - Flag de paridade**

- PF = 1 → caso o byte inferior do resultado de alguma operação aritmética ou lógica apresentar um número par de "1's";
    - PF = 0 → caso contrário (número ímpar).

# Processador INTEL 80X86

---

- **Flags de Estados (continuação):**
  - **AF - Flag de Carry Auxiliar:** utilizado em instruções com números BCD
    - **AF = 1** → caso exista o "vai um" do bit 3 para o bit 4 de uma adição; caso exista "empréstimo" do bit 4 para o bit 3 numa subtração;
    - **AF = 0** → caso contrário.
  - **ZF - Flag de Zero**
    - **ZF = 1** → caso o resultado da última operação aritmética ou lógica seja igual a zero;
    - **ZF = 0** → caso contrário.



# Processador INTEL 80X86

---

- **Flags de Estados (continuação):**
  - **SF - Flag de Sinal:** utilizado para indicar se o número resultado é positivo ou negativo em termos da aritmética em Complemento de 2 (se não ocorrer erro de transbordamento - *overflow*).
    - SF = 1 → número negativo;
    - SF = 0 → número positivo.
  - **OF - Flag de Overflow** (erro de transbordamento).
    - OF = 1 → qualquer operação que produza *overflow*;
    - OF = 0 → caso contrário.

# Processador INTEL 80X86

---

- **Flags de Controle:**

- **TF - Flag de Trap** (armadilha)

- TF = 1 → após a execução da próxima instrução, ocorrerá uma interrupção; a própria interrupção faz TF = 0;
    - TF = 0 → caso contrário

- **IF - Flag de Interrupção**

- IF = 1 → habilita a ocorrência de interrupções;
    - IF = 0 → inibe interrupções tipo INT externas.

- **DF - Flag de Direção:** usado para indicar a direção em que as operações com *strings* são realizadas.

- DF = 1 → decremento do endereço de memória (**DOWN**);
    - DF = 0 → incremento do endereço de memória (**UP**).

# Processador INTEL 80X86

- Os registradores do 8086 (resumo):

Registradores de dados			
AH	AL	→	AX
BH	BL	→	BX
CH	CL	→	CX
DH	DL	→	DX
Registradores de segmentos			
	CS		
	DS		
	SS		
	ES		
Registradores índices e apontadores			
	SI		
	DI		
	SP		
	BP		
	IP		
Registrador de sinalizadores			
	FLAGS		

# Processador INTEL 80X86

---

- **Barramentos**

- Barramento de endereços e de dados comuns:

- 20 bits**

- **Endereços:** 20 bits  $\rightarrow 2^{20} = 1.048.576$  combinações  
 $\rightarrow 1$  MByte (1 MB)

- **Dados:** **16 bits** (os menos significativos do barramento comum)

- Barramento de controle: **16 bits**

- **OBS.:** 8088  $\rightarrow$  versão do 8086 que se conecta externamente a um barramento de dados de 8 bits, processando internamente 16 bits.

# Processador INTEL 80X86

---

- Gerenciamento de Memória

- O 8086 possui 20 bits para acessar posições de memória física
- $2^{20} = 1.048.576$  bytes (1 Mbyte) posições endereçáveis
- Exemplos de endereços:

0000 0000 0000 0000 0000<sub>2</sub> -> 00000<sub>16</sub>  
0000 0000 0000 0000 0001<sub>2</sub> -> 00001<sub>16</sub>  
0000 0000 0000 0000 0010<sub>2</sub> -> 00002<sub>16</sub>  
0000 0000 0000 0000 0011<sub>2</sub> -> 00003<sub>16</sub>  
.....  
1111 1111 1111 1111 1111<sub>2</sub> -> FFFFF<sub>16</sub>

# Processador INTEL 80X86

---

- **Gerenciamento de Memória**
  - O 8086 opera internamente com 16 bits (todos os registradores internos são de 16 bits)
    - Problema: Como gerar endereços com 20 bits?
    - Solução: Utilizar a idéia de segmentação de memória!  
**Como?**

# Processador INTEL 80X86

---

- Gerenciamento de Memória

- Segmento de memória: bloco de 64 Kbytes de posições de memória consecutivas
- $2^{16} = 65.536$  bytes (64 Kbytes)
- Segmento de memória é identificado por um número de segmento (registrador de segmento)
- Uma posição de memória é especificada pelo número de segmento e por um deslocamento (*offset*) em relação ao início do segmento

# Processador INTEL 80X86

## • Gerenciamento de Memória

- Formato de endereço lógico → segmento:offset

- Exemplo de endereçamento:

Dado o endereço lógico:	8350:0420h	
reconhece-se: segmento no.	8350 <sub>16</sub>	} endereço lógico
deslocamento	0420 <sub>16</sub>	

o endereço físico vale:

8350<sub>16</sub> -> desloca-se 1 casa hexa (4 casas binárias)  
+ 0420<sub>16</sub> -> soma-se o deslocamento  
83920<sub>16</sub> -> endereço físico resultante (20 bits)



# Processador INTEL 80X86

---

- Gerenciamento de Memória

- O identificador de segmento (base) aponta para uma região da memória;
- O **offset** aponta para um local dentro deste segmento;
- O **offset** é aquele que aparece nos programas como o endereço dos dados, rótulos e endereços de instruções;

# Processador INTEL 80X86

---

- **Gerenciamento de Memória**

- Segmentação é um esquema muito útil para gerar códigos relocáveis;
- Endereços lógicos diferentes podem representar o mesmo endereço físico;

**Exemplo:**

base →	028C <sub>16</sub>
offset →	0003 <sub>16</sub>
endereço físico →	028C3 <sub>16</sub>

base →	0287 <sub>16</sub>
offset →	0053 <sub>16</sub>
endereço físico →	028C3 <sub>16</sub>

# Processador INTEL 80X86

- **Organização de Memória**

- A ocupação de memória feita pelo DOS (1024 Kbytes = 1 Mbyte) considera: 640 Kbytes para programas (10 segmentos disjuntos de 64 Kbytes) e 384 Kbytes reservados para memória de vídeo, BIOS, etc.

BIOS	F0000h
Reservado	E0000h
Reservado	D0000h
Reservado	C0000h
Vídeo	B0000h
Vídeo	A0000h
Área de programas de aplicação	
Sistema Operacional (DOS)	
BIOS e dados da BIOS	00400h
Vetores de interrupção	00000h

# Processador INTEL 80X86

---

- **Interrupção**

- Ocorrência eventual, durante a execução de um processamento pelo computador, que deve ser prontamente atendida, causando a suspensão do processamento em curso para o atendimento da "*chamada*".

- **Tipos de Interrupções do 8086:**

- Não mascarada : Causadas pela ocorrência de eventos "catastróficos": falta de energia, erro de memória, erro de paridade em comunicações, etc. (este tipo de interrupção não pode ser inibida).
- Mascarada: Causadas pela ação de dispositivos externos (periféricos): podem ser habilitadas ou inibidas.

# Processador INTEL 80X86

---

- **Tipos de Interrupções do 8086:**
  - Causadas pelo próprio programa em curso: erro de divisão, erro de transbordamento (*overflow*).
  - TRAP - útil para depuração de um programa
  - BREAKPOINT - colocado em pontos estratégicos do programa para permitir processamento especial
  - Interrupção RESET: permite a inicialização do microprocessador, via *hardware*.