X86

Marcos Monteiro Junior

Introdução

O microprocessador 8086 da Intel é um microprocessador de 16 bits, de forma que sua unidade lógica e aritmética, os seus registradores internos, e a maior parte das suas instruções foram projetados para trabalhar com palavras de 16 bits.

Além disso o 8086 tem um barramento de dados de largura 16 bits, ou seja, pode ler e escrever na memória ou nos pontos de E/S utilizando 16 bits de uma só vez.

O barramento de endereços é de 20 bits, de forma que o 8086 pode endereçar 1 MB(220) posições de memória.

Introdução

Cada uma destas posições de memória é ocupada por um Byte.

A arquitetura do 8086 pode ser organizada em duas unidades distintas: a BIU (Bus Interface Unit) e a EU (ExecutionUnit).

A BIU

- envia endereços para o barramento de endereços
- lê instruções da memória
- lê e escreve dados nas portas e na memória.

Assim, a BIU é a unidade responsável por todas as transferências de dados e endereços através dos barramentos.

Por sua vez, a EU diz à BIU onde encontrar instruções ou dados, decodifica e executa as instruções.

Introdução

O 8086 foi lançado em junho de 1978 e tinha um clock de 4,77 MHz.
A IBM resolveu adotar o 8088 para o seu computador pessoal por dois motivos:

Manter os custos do PC reduzidos

Manter a compatibilidade com chips periféricos: o 8088 aceitava um barramento interno de 16 bits (como o 8086), mas seu barramento externo era de 8 bits.

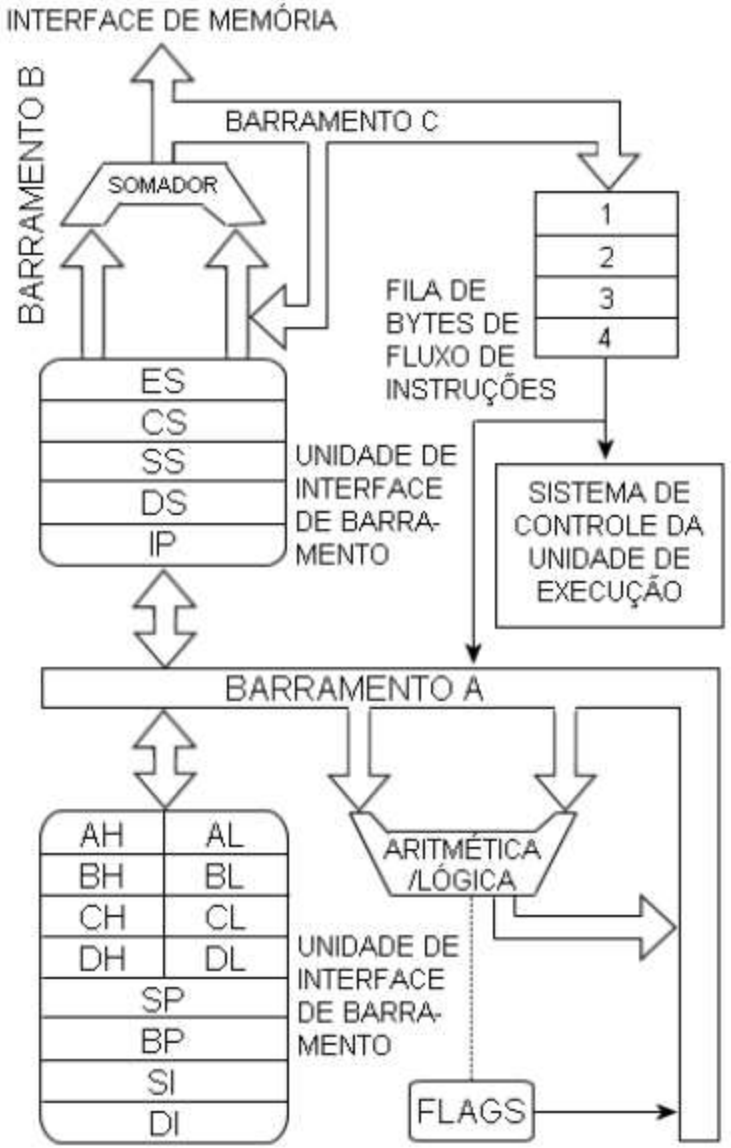
Introdução

O processador 8088 foi implementado com 29000 transistores e compactado num pacote de 40 pinos.

O modelo de programação básico é muito similar aos processadores mais modernos (de 32 e 64 bits) e as características atuais (como registradores, tipos de dados e modo de endereçamento) são extensões do conjunto de recursos do 8086 original.

Introdução

A UCP tem 4 registradores internos, cada um de 16 bits, chamados de AX, BX, CX e DX.
Esses registradores são de uso geral e também podem ser usados como registradores de 8 bits. Para isso, tais registradores devem ser referenciados como, por exemplo, AH e AL, que são, respectivamente, o Byte high e o low do registrador AX.
Esta nomenclatura também se aplica para os registradores BX, CX e DX.


INTERFACE DE MEMÓRIA
BARRAMENTO B
BARRAMENTO C
SOMADOR
1
2
3
4
FILA DE BYTES DE FLUXO DE INSTRUÇÕES
ES
CS
SS
DS
IP
UNIDADE DE INTERFACE DE BARRA-MENTO
SISTEMA DE CONTROLE DA UNIDADE DE EXECUÇÃO
BARRAMENTO A
AH
AL
BH
BL
CH
CL
DH
DL
SP
BP
SI
DI
ARITMÉTICA /LÓGICA
UNIDADE DE INTERFACE DE BARRA-MENTO
FLAGS

Registradores

AX	Registrador Acumulador
BX	Registrador Base
CX	Registrador Contador
DX	Registrador de Dados
DS	Registrador de Segmento de Dados
ES	Registrador de Segmento Extra
SS	Registrador de Segmento de Pilha
CS	Registrador de Segmento de Código
BP	Registrador Apontador da Base
SI	Registrador de Índice Fonte
DI	Registrador de Índice Destino
SP	Registrador Apontador de Pilha
IP	Registrador Apontador da Próxima Instrução
F	Registrador de Flag

Registradores

- AX (acumulador) -> utilizado como acumulador em operações aritméticas e lógicas; em instruções de E/S, ajuste decimal, conversão, etc.
- BX (base) -> usado como registrador de 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).
- 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 E/S

Registradores de segmentos

- CS, DS, SS e ES
- São todos registradores de 16 bits .
- O endereçamento no 8086 é diferenciado para:
  - código de programa (instruções)
  - dados
  - pilhas

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: DS
  - segmento de pilha: SS (stack segment)
  - segmento extra: ES

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.

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.

- Podem ser utilizados em operações aritméticas e lógicas, possibilitando que os valores de deslocamento sejam resultados de computações anteriores.

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.
- BP (base pointer - apontador de base) é o ponteiro que, em conjunto com SS, permite acesso de 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.

Registrador de sinalizadores (FLAGS)

- Indica o estado do microprocessador durante 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)

Flags de estados

- CF - Flag de Carry
  - CF = 1 -> após instruções de soma que geram "vai um" após instruções de subtração que não 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 impar). ·
- 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 não exista "empréstimo" do bit 4 para o bit 3 numa subtração;
  - AF = 0 -> caso contrário.

Flags de estados

- ZF - Flag de Zero
  - ZF = 1 -> caso o resultado da última operação aritmética ou lógica seja igual a zero;
  - ZF = 0 -> caso contrário. ·
- 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.

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).