

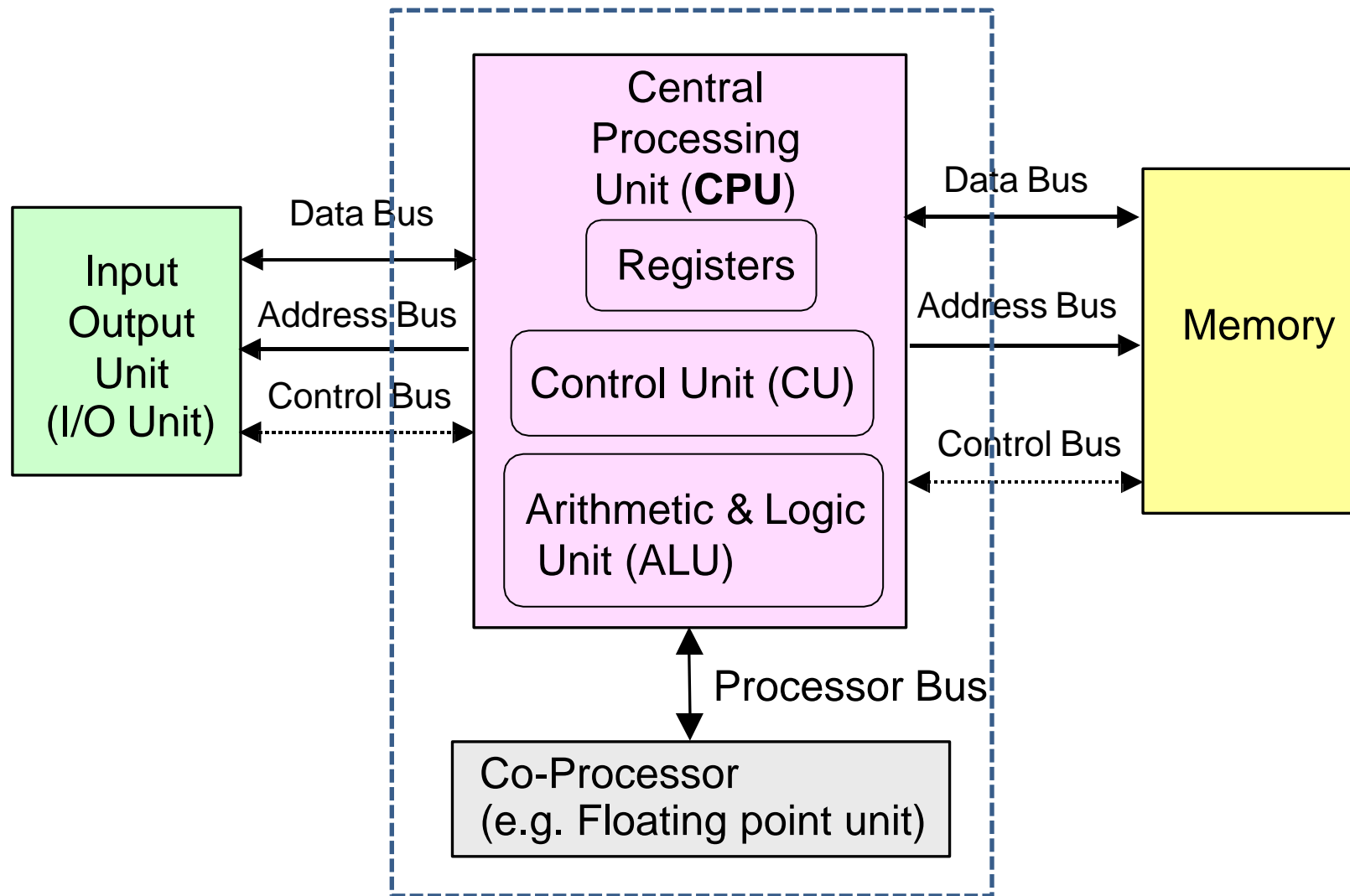
ELE8575

Arquitetura do 8088/8086

**Slides baseados no material do curso
do**

Prof. Chen Lian Kuan

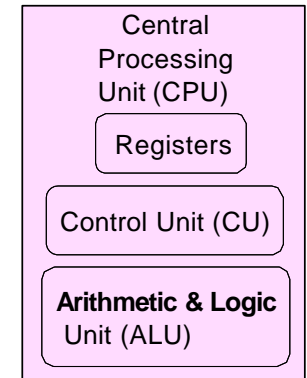
Sistema Microprocessado



Sistema Microprocessado

Unidade de controle (*Control Unity*) gera todos os sinais de controle dentro da CPU. Ela inicializa os registros ao ligar, gera o sinal para buscar as instruções para a ULA (Unidade Lógica e Aritmética).

A unidade de controle pode ser implementada (i) completamente por hardware (controlador com fio rígido, por exemplo, usando um contador de estado e uma matriz lógica programável) ou (ii) por uma mistura de instruções de software (microcódigo armazenado na CPU) e hardware (controle microprogramado). Tanto a família Intel 8086 como a família Motorola 68000 utilizam controladores microprogramados.



Registradores (*Registers*) - elemento de memória estática, e rápida, baseado em Flip-Flops, que normalmente armazena dados e endereços associados à instrução que está sendo executada.

ALU realiza operações aritméticas e lógicas.

Ciclo de Instrução

Duas etapas principais no ciclo:

- a) Buscar (*Fetch*) a próxima instrução na memória principal
- b) Decodificar (*Decode*) e Executar (*Execute*) a instrução

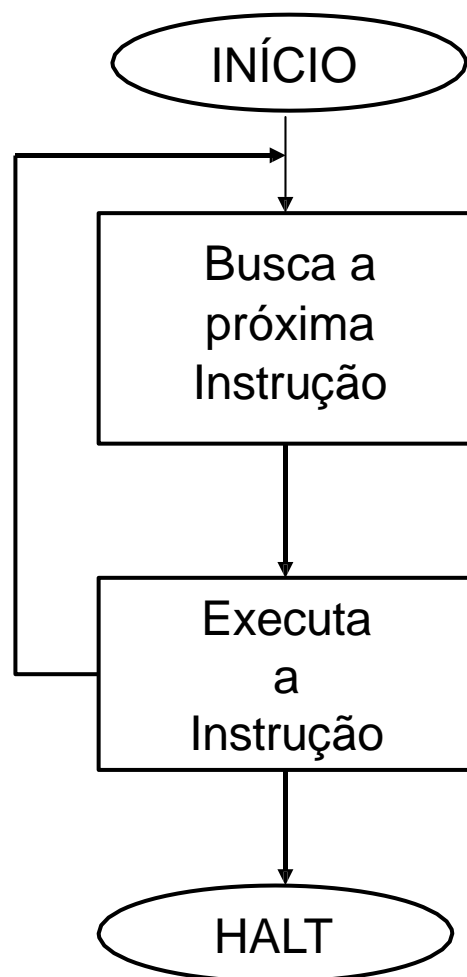
O ciclo *Fetch* consiste em usar o ponteiro de instruções (IP) para

- a) definir o barramento de endereço com o endereço da próxima instrução e incrementar o ponteiro de instruções;
- b) esperar (poucas centenas de nanossegundos) para que os dados sejam transferidos da memória para o barramento de dados e;
- c) ler os dados do barramento de dados.

O Ciclo de Execução consiste em

- a) Decodificar a instrução e gerar a seqüência correta de sinais internos e externos
- b) Executar a instrução e reiniciar o ciclo Fetch

Ciclo de Instrução



Ciclo *Fetch*

Ciclo *Decodifica e Executa*

Ciclos de Busca e Execução

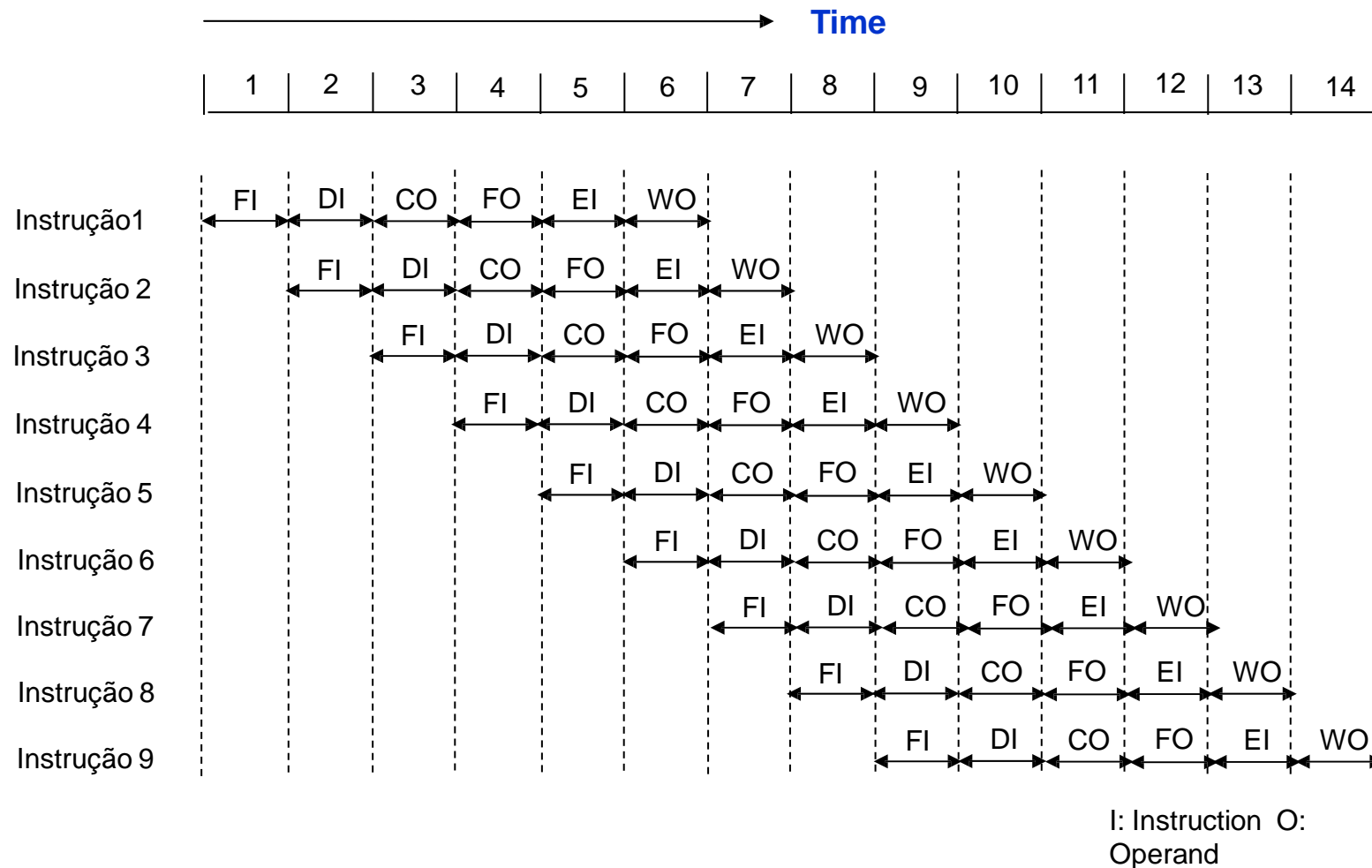
Instruction Fetch and Execution pipeline

Fetch	Execute	Fetch	Execute	Fetch	Execute	Fetch	
-------	---------	-------	---------	-------	---------	-------	--

- No 8088 e 8086 os ciclos de Busca e Execução (*Fetch & Execution*) são implementados por duas unidades de processo dentro da CPU:
 - i) Bus Interface Unit (BIU) vai buscar instruções da memória, passa a instrução para a fila de bytes do fluxo de instruções e começa a buscar a próxima instrução imediatamente;
 - ii) Unidade de Execução (EU) retira as instruções da fila de instruções e as executa.

Como as 2 unidades são praticamente processadores separados, então tanto a BIU quanto a EU podem trabalhar simultaneamente sem esperar pela conclusão da outra tarefa (processamento *pipeline*).

Diagrama de tempo hipotético de um Pipeline de Instrução

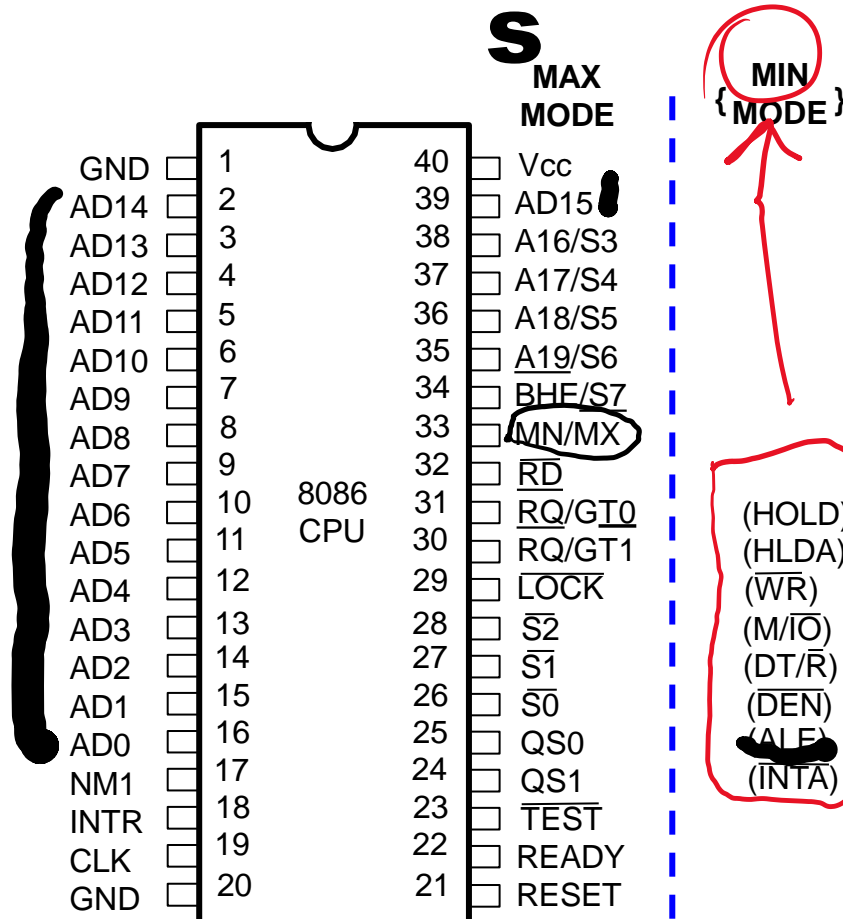


Fetch Instruction – Decode Instruction – Check Operand – Fetch Operand (se necessário) – Execute Instruction – Write Output
 (“Operando” é definido como parte da Instrução)

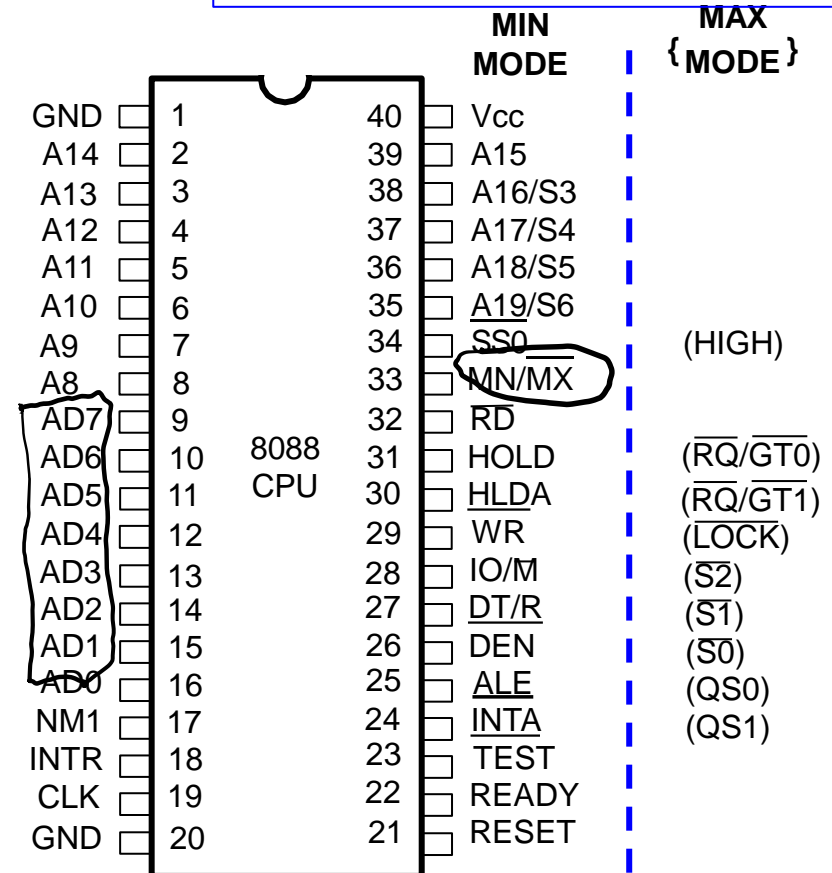
Introduction to Intel 8086/8088

Microprocessor

Alguns pinos têm funções diferentes para dois modos de operação diferentes.



8086 pin diagram



8088 pin diagram

Microprocessadores 8086/8088



8088 e 8086 são quase idênticos, exceto que o 8088 tem apenas 8 linhas de dados externos, enquanto o 8086 tem 16 linhas de dados externos.

- i) Ambos têm Barramento de dados de 16 bits de largura internamente ao microprocessador;
- ii) 20 pinos de endereço, incluindo 16 pinos de endereço/dados (AD0-AD15) + 4 pinos de endereço/status (A16/S3-A19/S7) para 8086, permitindo uma faixa máxima de endereços de memória de 1MByte;
- iii) Ambos realizam multiplexação de endereço/dados (8088 somente multiplexação de 8 pinos, AD0- AD7);
- iV) Ambos possuem 2 modos de operação (os modos máximo e mínimo) e o mesmo conjunto de instruções.

Arquitetura Interna



Ambos, 8088 e 8086, empregam processamento paralelo.

- i) Contêm duas unidades de processamento: Unidade de execução (UE) e unidade de interface Bus (BIU); operar ao mesmo tempo.
- ii) A BIU envia endereços, pega instruções da memória, lê dados de portas e memória e grava dados em portas e memória, ou seja, a BIU lida com todas as transferências de dados e endereços nos barramentos para a unidade de execução.
- iii) A UE diz à BIU onde buscar instruções ou dados, decodifica instruções e executa as instruções.

Diagrama de blocos interno do 8086

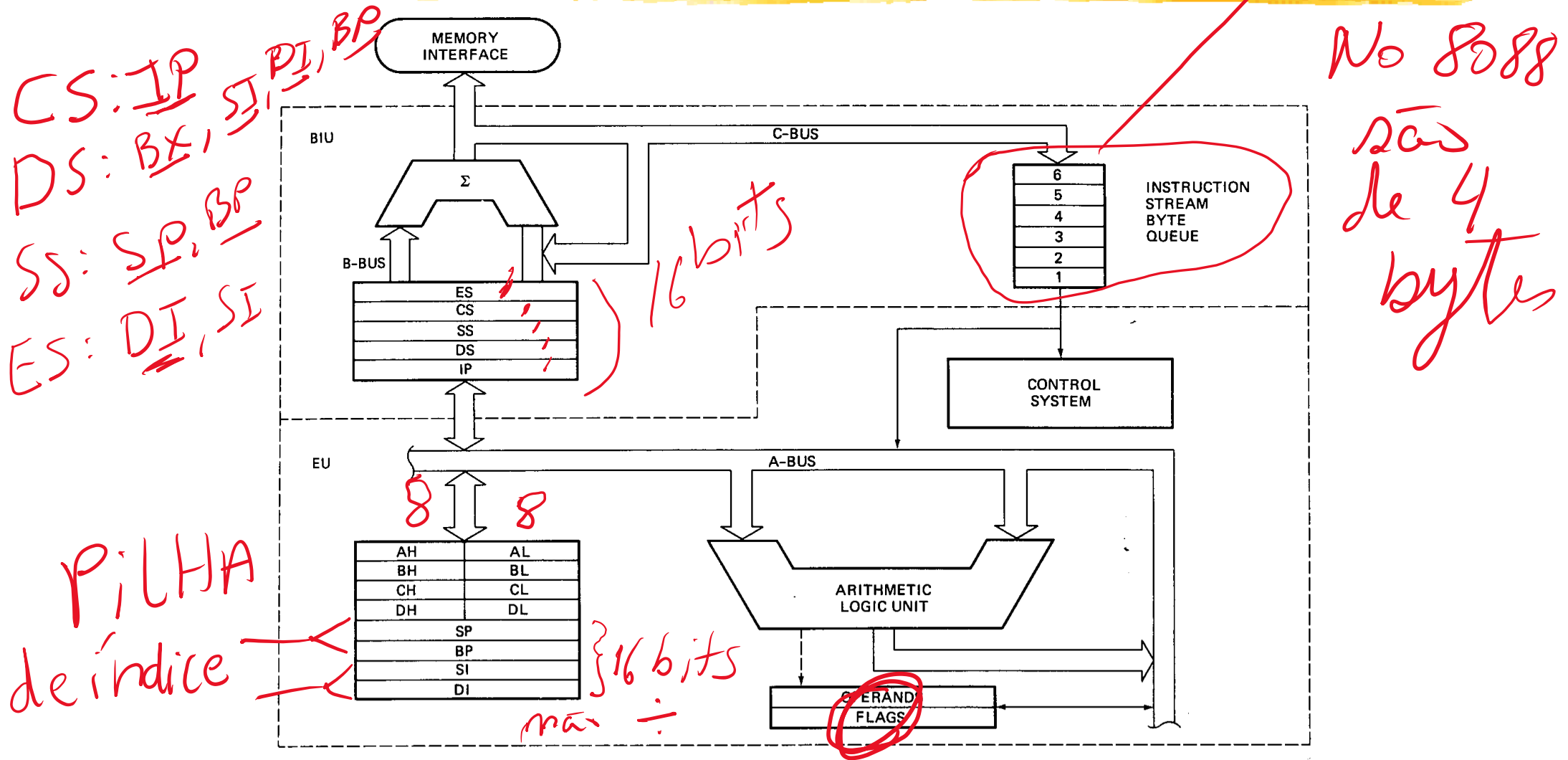
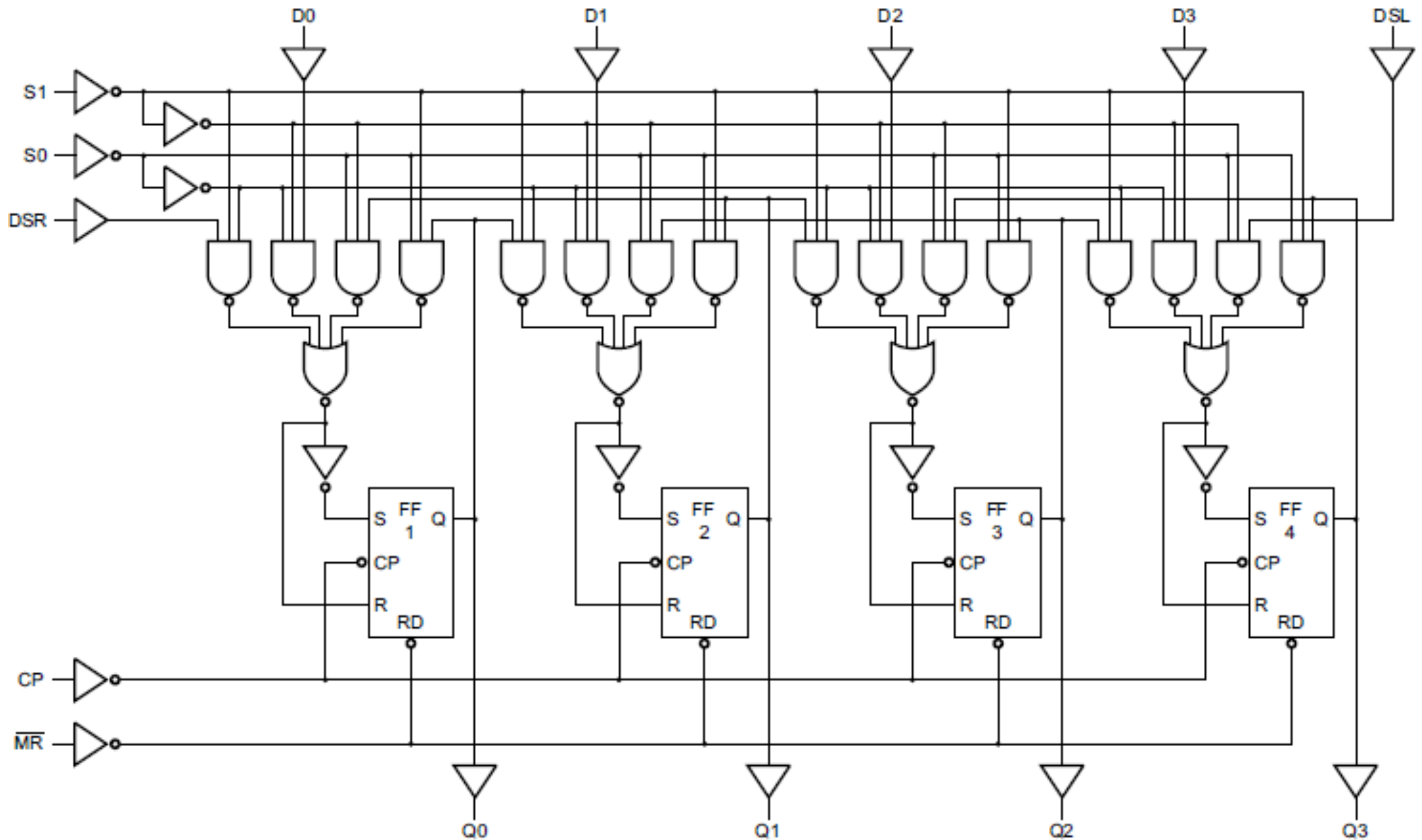


FIGURE 2-7 8086 internal block diagram. (Intel Corp.)

Exemplo de um registrador síncrono universal de 4 bits



Bus Interface Unit (BIU)

- ✓ Realiza operações de barramento como busca de instruções, leitura/escrita de operando de dados para memória, entrada/saída de dados para periféricos de E/S.
- ✓ Executa outras funções, tais como fila de instruções e aquisição de dados.
- ✓ Barramento de dados bi-direcional de 8 bits (16 bits) para 8088 (8086).
- ✓ O barramento de endereço de 20 bits pode endereçar qualquer um dos 2^{20} (1.048.576) bytes de memória .
- ✓ Contém registro de segmento, ponteiro de instruções, somador de geração de endereços, lógica de controle do barramento e uma fila de instruções.
- ✓ Use a fila de instruções para implementar uma arquitetura pipelinada (prefetch até 4 (6) bytes de código de instrução para 8088 (8086) e depois armazene e acesse os códigos na ordem FIFO).

Execution Unit (**EU**)

Responsável pela decodificação e execução das instruções.

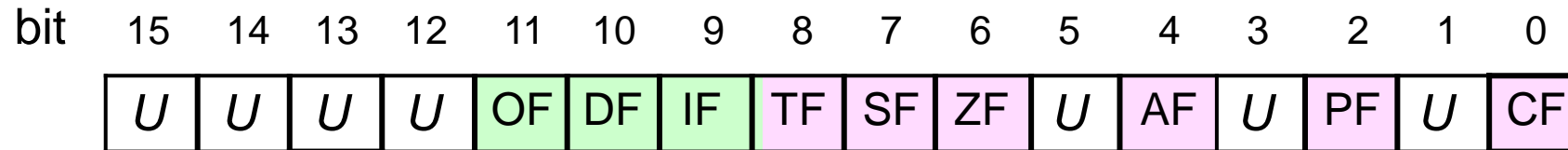
- ✓ Contém: unidade lógica aritmética (ALU), sinalizadores (**flags**) de status e controle, registros de propósito geral e registro de operação temporária.
- ✓ A EU acessa a instrução a partir do final da fila de saída da instrução e os dados do registro de propósito geral;
- ✓ Lê uma instrução de cada vez, decodifica-as, gera endereço de operando se necessário, passa-as para a BIU e solicita a execução do ciclo de leitura/gravação na memória ou E/S, e executa a operação especificada pela instrução no operando;
- ✓ Durante a execução, a EU pode testar os bits de **flag** de status e controle e atualizar estas bandeiras com base nos resultados da execução.

Registrador de Flags

O registro de *flags* indica a condição do microprocessador, assim como controla suas operações.

- ✓ Um registrador de *flags* é um flip-flop que indica alguma condição produzida pela execução de uma instrução ou controla certas operações da EU.
- ✓ O registrador de flags de 16 bits na EU contém nove bits de *flags* ativos. (Cada flag ocupa um bit no registrador de *flags*).
- ✓ São dois tipos de flags:
- ✓ *flags* condicionais: Seis bits de *flags* são condicionais. Eles são definidos ou reiniciados pela EU com base nos resultados de alguma operação aritmética ou lógica, processada pela ULA (ou ALU).
- ✓ *flags* de controle : os três bits de *flags* restantes no registrador são usados para controlar certas operações do processador. Eles são chamadas de *flags* de controle.

Registrador de Flags



U=Undefined

Flags Condicionais

Carry Flag (CF)- set by carry out of MSB.
Parity Flag (PF)- set if result has even parity.
Auxiliary carry Flag (AF)- for BCD
Zero Flag (ZF)- set if results = 0
Sign Flag (SF) = MSB of result
Overflow Flag (OF)- overflow flag

Flags de Controle

IF- interrupt enable flag
DF- string direction flag
TF- single step trap flag

MSB: Most Significant Bit

Flags Condicionais

carry flag (CF)- indica um transporte após a adição ou um empréstimo após a subtração.

parity flag (PF)- vale “0” para paridade ímpar e “1” para paridade par.

auxiliary carry flag (AF)- importante para adição e subtração BCD; mantém um transporte (empréstimo) após adição (subtração) entre bit-3 e bit-4. Usado somente para instruções DAA e DAS para ajustar o valor de AL após uma adição (subtração) de BCD

zero flag (ZF)- indica, em “1”, se o resultado da operação realizada na ULA foi “0”;

sign flag (SF)- indica o sinal aritmético da operação realizada na ULA;

overflow flag (OF)- uma condição que ocorre quando números com sinais são adicionados ou subtraídos. Uma situação de *overflow* indica que o resultado excedeu a capacidade de representação do registrador ou da variável declarada em memória.

Flags de Controle

The control flags are deliberately set or reset with specific instructions YOU put in your program. The three control flags are:

trap flag (TF) – usado para fazer com que o processador execute uma instrução por vez (muito usado para debugging);

interrupt flag (IF) – usado para liberar ou bloquear a ocorrência de interrupções do tipo int;

direction flag (DF) – usado em instruções de manipulação de strings.

Não existe uma instrução específica para definir o TF.

Registradores de Propósito Geral

A EU possui oito registradores de 8-bit denominados *general-purpose registers* (*registradores de propósito geral*): AH, AL, BH, BL, CH, CL, DH, and DL. Esses registradores podem ser utilizados individualmente para armazenamento temporário de dados de 8 bits.

- ✓ Os pares de registros AH-AL, BH-BL, CH-CL e DH-DL podem ser usados em conjunto para formar os registros AX, BX, CX e DX e podem ser usados para armazenar palavras de dados de 16 bits.
- ✓ O registrador AL também é chamado de acumulador. Ele tem algumas características que os outros registros de propósito geral não possuem.
- ✓ O uso dos registradores internos é vantajoso porque eles podem ser acessados mais rapidamente do que a memória externa. Nenhuma referência de memória ou ciclo de memória é necessário para obter os dados.