

OS MICROPROCESSADORES 8086/88 - HARDWARE

1 - INTRODUÇÃO

Um **microprocessador** é uma pastilha que incorpora em um único componente, a maior parte do hardware necessário para a implementação de uma CPU completa de computador. Entre outros, o microprocessador contém registradores, pelo menos uma unidade lógica e aritmética, hardware para cálculo de endereços e um controlador capaz de executar instruções de uma linguagem de máquina.

Existem diversos tipos diferentes de microprocessadores, adequados a diferentes aplicações. Existem por exemplo, microprocessadores otimizados para realizar cálculos aritméticos em alta velocidade chamados **DSP** ("Digital Signal Processors"). Os DSPs encontram aplicações em sistemas de tempo real que implementam funções de processamento digital de sinais como modems para comunicação de dados, telefones celulares digitais, CD players, equipamentos para home theater, processadores de som e vídeo etc. Um DSP geralmente possui instruções específicas para uso em filtragem digital e processamento digital como cálculo de FFT ("Fast Fourier Transform") entre outras. Uma outra classe de microprocessadores é composta pelos chamados **microcontroladores**, que são pastilhas de microprocessadores que já contém embutidos uma série de periféricos, tais como memórias RAM e ROM, portas seriais, portas paralelas, timers, controladores de interrupção, controladores de RAM dinâmica, controladores de DMA etc (Obs.: nem todos os microcontroladores possuem todos os periféricos citados embutidos, podendo utilizar alguns deles externamente se desejado). Os microcontroladores são ideais para aplicações que requerem alta compactação física, baixo consumo de energia e baixo custo, como fornos de microondas, impressoras, robôs de sensoriamento remoto etc.

Em geral, pode ser muito difícil comparar dois microprocessadores e eleger um "melhor" porque seu desempenho é bastante dependente da aplicação. No entanto, é comum dividir-se os microprocessadores em classes em função do comprimento de seus registradores internos. Assim um microprocessador é dito de 16 bits, quando seus registradores são de 16 bits, 32 bits quando seus registradores são de 32 bits e assim por diante. Em geral, um microprocessador é capaz de executar programas complexos mais rapidamente a medida que o comprimento de seus registradores cresce e por isso existe a tendência de assumir que um processador de 32 bits é "melhor" ou mais "potente" que um de 16 bits, apesar de isto não ser sempre verdade para qualquer aplicação. Apenas o estudo detalhado de um determinado microprocessador permite determinar se ele é ou não o mais adequado a uma dada aplicação.

2 - ESTUDO DE UM PROCESSADOR

O estudo de uma determinada pastilha de microprocessador pode ser feito de forma sistemática, abordando-se os seguintes tópicos:

a) **Arquitetura do microprocessador**: Consiste de uma descrição de todo o hardware, com nível de detalhe suficiente para a exploração de todas as potencialidades do chip. Ao estudar-se a arquitetura de um microprocessador, determina-se por exemplo quais são os registradores que este contém e a função de cada um, o mecanismo associado ao cálculo de endereços, dispositivos acessórios disponíveis tais como circuito para paginação de memória, lógica para segmentação, etc. Neste estudo, procura-se ignorar todas as partes do circuito que são "invisíveis ao programador", ou seja, detalhes de circuitos que permitem o controle dos barramentos de comunicação internos, registradores temporários etc. que

possibilitam a operação correta do microprocessador mas que não podem ser diretamente controlados por nenhuma instrução. Em outras palavras, estuda-se tudo aquilo que permite compreender o que é possível fazer com os recursos do microprocessador, deixando-se de lado o estudo do "como é feito".

b) **Pinagem:** Uma descrição de todos os pinos do chip, bem como de suas funções lógicas. É necessário conhecer bem a pinagem de um microprocessador para poder usá-lo em um sistema como um computador.

c) **Temporização e níveis elétricos:** Nesta fase, estuda-se as formas de onda dos sinais em cada pino e as tensões e correntes, de modo a poder conectar o chip a periféricos com sucesso.

d) **Conjunto de instruções:** É composto de todas as instruções de máquina que o microprocessador é capaz de executar. Seu conhecimento é fundamental para que um programa executável pelo microprocessador possa ser escrito.

Cada uma das etapas será ilustrada mediante o estudo de um microprocessador exemplo o 8086 (8088).

2.1 - Arquitetura do 8086/88

O 8088 é um microprocessador de 16 bits lançado pela INTEL e utilizado pela IBM no seu primeiro computador pessoal, o IBM - PC -XT. A figura 2.1 ilustra a arquitetura interna simplificada do 8088. O 8086 é um microprocessador cuja arquitetura interna é quase igual à do 8088 (a diferença está no tamanho da fila de bytes que armazena instruções). A principal diferença entre os dois está no tamanho do barramento de dados externo, que é de 8 bits no 8088 e de 16 bits no 8086.

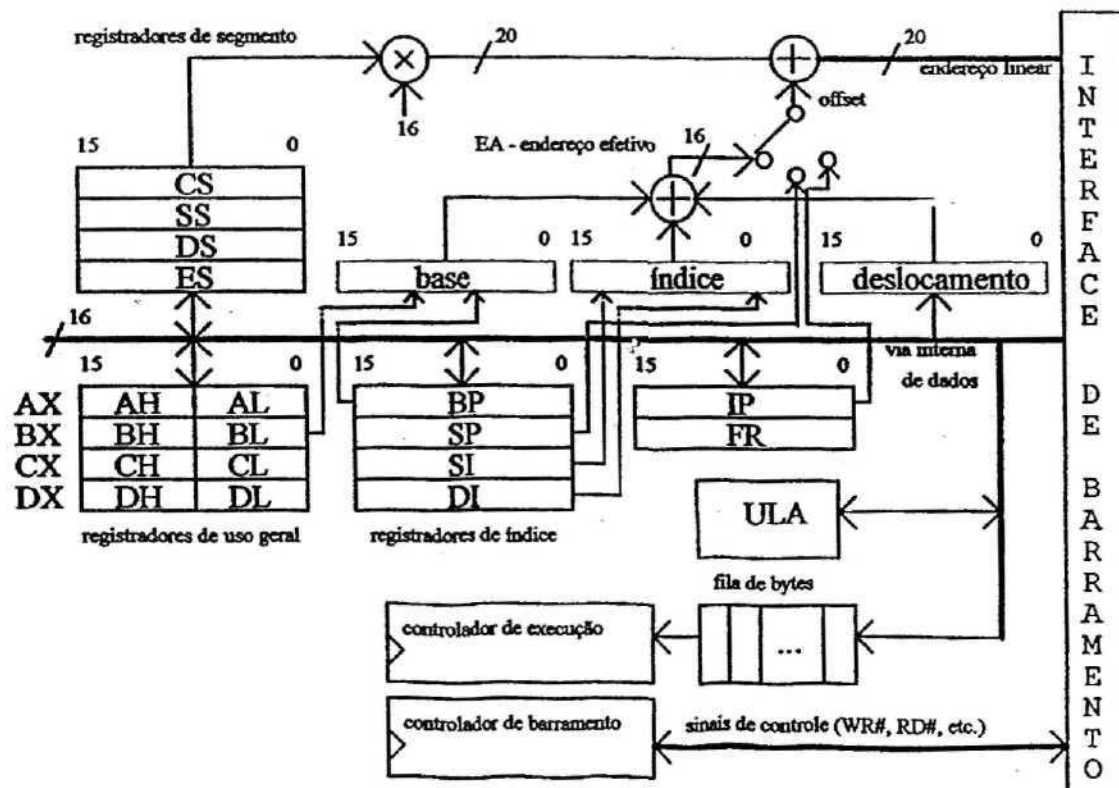


Figura 2.1 - Arquitetura interna simplificada do 8086/88.

Os microprocessadores 8086/88 possuem 14 registradores de 16 bits, dos seguintes tipos:

a) **4 registradores de uso geral: AX, BX, CX, DX.** Estes registradores podem ser usados na maior parte das operações lógicas e aritméticas que o 8086 (8088) é capaz de realizar. Cada um destes registradores está subdividido em dois registradores de 8 bits, um correspondendo à parte alta (byte mais significativo) e o outro à parte baixa (byte menos significativo). Desse modo o AX se subdivide em AH e AL, o BX em BH e BL, o CX em CH e CL e o DX em DH e DL. Cada um destes registradores pode ser acessado e usado como um registrador de 16 bits ou 2 registradores de 8 bits. Por exemplo, supondo que o registrador AX é carregado com o valor de 16 bits F75A, automaticamente AH é carregado com F7 enquanto AL é carregado com 5A. Se em seguida o registrador AL é modificado, por exemplo com o valor 77, AH permanece inalterado e o conteúdo de AX passará a ser F777. O seguinte exemplo ilustra bem a relação entre AX, AH e AL.

Carrega-se inicialmente AX com 00FF. Se em seguida o registrador AX é incrementado, o conteúdo de AX passa a ser 0100. Contudo, se AX é inicialmente carregado com 00FF porém ao invés de incrementar o AX incrementa-se o AL, o resultado em AX será 0000. Isto ocorre porque uma operação com AL não gera um transporte do bit mais significativo de AL para o menos significativo do AH.

O mesmo tipo de relacionamento existe entre os demais registradores de uso geral.

Apesar de semelhantes, os registradores de uso geral não são idênticos, porque existem determinadas instruções de software que só podem ser executadas usando determinados registradores dentre eles. Por exemplo, somente o BX pode ser usado para compor um endereço (endereçamento indireto), somente o CX pode ser usado como contador em instruções de loop (repetição). O registrador AX, que recebe o nome especial de **acumulador**, é o único que pode ser usado em todos os tipos de operações aritméticas e lógicas.

b) **4 registradores apontadores e de índice: BP, SP, SI e DI.** Estes registradores são usados para compor endereços de operandos na memória mas também podem ser utilizados em operações lógicas e aritméticas com valores de 16 bits.

Os registradores SP e BP são usados para acessar dados no segmento de pilha. O SP (Stack Pointer) armazena o *offset* do endereço do topo da pilha, controlando a alocação dinâmica de variáveis e o empilhamento de endereços de retorno de subrotinas. Todas as referências ao SP, por definição, usam o registrador de segmento SS. O BP é tipicamente usado para acessar parâmetros que foram passados via pilha para uma determinada subrotina.

Os registradores SI e DI facilitam a movimentação de dados seqüenciados entre posições fonte (indicado por SI) e posições destino (indicado por DI), sendo muito aplicados nas operações com strings.

c) **1 ponteiro de instrução: IP.** É um registrador de 16 bits, sendo equivalente ao PC (program counter) de outros microprocessadores. O IP é automaticamente incrementado em função do número de bytes da instrução executada, de modo a apontar sempre para a instrução seguinte.

d) **1 registrador de estado (flags): PSW (Program Status Word).** É um registrador de 16 bits, os quais são tratados como bits individuais, cada qual indicando alguma propriedade. Sete destes bits não são utilizados e geralmente recebem valor lógico zero. A principal aplicação dos *flags* resultantes de operações matemáticas é o controle de saltos condicionais.

Sua organização é a seguinte:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				O	D	I	T	S	Z		A		P		C

d.1) **Flag de Carry (C)**: recebe valor lógico 1 quando ocorre o "vai um" do bit mais significativo nas operações de soma (de 8 ou 16 bits) ou na subtração que não gera "empréstimo" (*borrow*). Também é modificado por algumas instruções de rotação e deslocamento..

d.2) **Flag de Paridade (P)**: indica a paridade (par) dos 8 bits menos significativos do resultado de uma operação realizada. $P=1$ se ocorre um número par de "1's" nos 8 bits menos significativos e $P=0$ se ocorre um número ímpar de "1's" nos 8 bits menos significativos.

d.3) **Flag de Carry Auxiliar (A)**: recebe valor lógico 1 quando ocorre o "vai um" do bit 3 para o bit 4 em uma operação de soma ou o não empréstimo do bit 4 para o bit 3 em uma subtração..

d.4) **Flag de Zero (Z)**: recebe valor lógico 1 quando o resultado da última operação aritmética ou lógica for igual a zero.

d.5) **Flag de Sinal (S)**: é igual ao bit de mais alta ordem do resultado de uma operação aritmética em complemento a 2 (se não ocorrer erro de transbordamento - *overflow*). Se $S=0$ implica que o resultado é positivo. Se igual a 1 indica que é negativo.

d.6) **Flag de Trap (Armadilha) (T)**: recebe valor lógico 1 quando após a execução da próxima instrução ocorrer uma exceção (interrupção gerada pelo próprio processador). É utilizado para a depuração de programas, colocando o 8086/8088 no modo passo a passo.

d.7) **Flag de Interrupção (I)**: recebe valor lógico 1 quando habilita a ocorrência de interrupções.

d.8) **Flag de Direção (D)**: usado para indicar a direção em que as operações com *strings* são realizadas. Se $D=1$, os registradores SI e DI serão decrementados, ou seja, a *string* será acessada a partir do endereço mais alto em direção ao mais baixo. Se $D=0$, os registradores SI e DI serão incrementados, ou seja, a *string* será acessada a partir do endereço mais baixo em direção ao mais alto.

d.9) **Flag de Overflow (O)**: recebe valor lógico 1 quando houver um erro de transbordamento. Ele indica um *overflow* de magnitude em aritmética binária com sinal.

e) 4 registradores de segmentos: CS, SS, DS e ES. Um endereço de memória no 8086 (8088) é sempre composto de duas partes: Um valor de **segmento** e um **offset** dentro do segmento. O segmento é obtido multiplicando-se por 16 o conteúdo de um dos registradores de segmento, sendo o resultado representado com precisão de 20 bits. Por exemplo, se o registrador DS contém o valor 1000, o valor de segmento definido por este registrador será 10000. O offset é um número de 16 bits obtido de três modos diferentes, dependendo do tipo de operação efetuada pelo microprocessador:

e.1) Busca (fetch) de instrução: Neste caso, a instrução é buscada na memória no segmento dado por CS e no offset dado por IP (instruction pointer - apontador de instrução). O valor de IP (de 16 bits) é estendido para precisão de 20 bits colocando-se 4 bits zero à esquerda do mesmo e é então somado ao segmento, formando deste modo um **endereço linear** de 20 bits. Por exemplo, se num dado momento, CS contém F000 e IP contém 0123, a próxima instrução será buscada a partir do endereço F0123 ($F0000 + 00123$).

e.2) Busca de operando: Neste caso o segmento é dado por qualquer um dos 4 registradores de segmento, enquanto o offset é calculado pela soma módulo 16 de até 3 parcelas de 16 bits: uma **base**, um **índice** e um **deslocamento**. Para o valor de base pode ser usado o conteúdo do registrador BX ou o conteúdo do registrador BP. Para o índice, podem ser usados SI ou DI. O deslocamento é um número de 8 bits (estendido para 16) ou 16 bits que é codificado diretamente na instrução a ser executada. O resultado da soma módulo 16 é também chamado de endereço efetivo (EA - effective address).

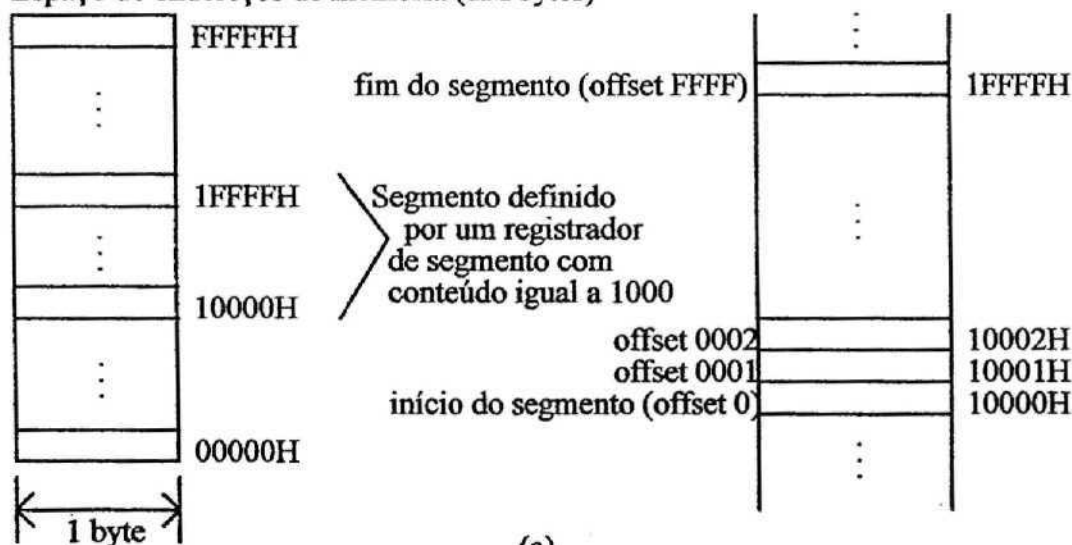
e.3) Operações com a pilha: Neste caso o segmento é dado por SS e o offset é dado por SP (stack pointer - apontador de pilha).

Observa-se que devido a forma de cálculo dos endereços, a memória do 8086 (8088) é segmentada. Os 20 bits de endereço linear permitem endereçar $2^{20} = 1048576$ bytes, ou seja 1M bytes. Contudo, o espaço de endereçamento está subdividido em segmentos de tamanho fixo igual a 64K bytes ($2^{16} = 65536$). São necessários pelo menos 16 segmentos diferentes para endereçar todas as posições de memória. Num dado instante, o 8086 (8088) é capaz de acessar dados em até 4 segmentos distintos, um para cada registrador de segmento. Contudo nada impede que mais de um destes segmentos estejam mapeados na mesma faixa de endereços lineares, pois os registradores de segmentos podem ser carregados com qualquer valor, independentemente. O tamanho de 64K fixo dos segmentos é determinado pela precisão de 16 bits usada para representar o offset. A figura 2.2a ilustra a segmentação do espaço de endereçamento de memória do 8086 (8088).

Além do espaço de endereçamento de memória, o 8086 (8088) possui um espaço de endereços adicional para I/O (Input/Output - entrada e saída) de 64K bytes. Este espaço é independente do espaço de memória e o 8086 (8088) possui um mecanismo de sinalização externo que indica a qual dos espaços o processador está se referindo em um dado instante.

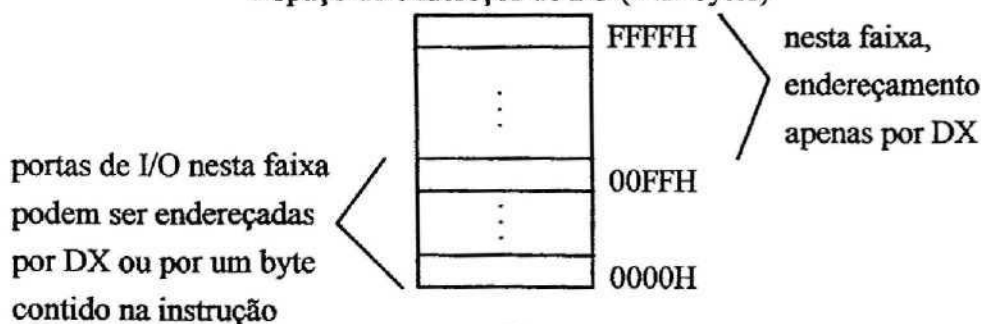
O espaço de I/O não é segmentado e os endereços são dados pelo conteúdo do registrador DX ou por um byte contido na instrução sendo executada. A figura 2.2b ilustra o espaço de endereçamento de I/O.

Espaço de endereços de memória (1M bytes)



(a)

Espaço de endereços de I/O (64K bytes)



(b)

Figura 2.2 - Espaços de endereçamento de memória e I/O do 8086.

Observa-se que um endereço pode ser referido usando-se um endereço segmentado, por exemplo 1000:0020 (segmento = 10000, offset = 00020) ou o seu equivalente linear, neste caso 10020).

O microprocessador 8086/88 está dividido em duas unidades independentes que operam de forma assíncrona:

- **BIU (Bus Interface Unit)** – Unidade de Interface com o Barramento - responsável pelo interfaceamento com o barramento externo e geração de endereços para acessar a memória e I/O. Possui uma fila de bytes para código (6 bytes no 8086 e 4 bytes no 8088).
- **EU (Execution Unit)** – Unidade de Execução - responsável pela decodificação e execução das instruções.

Esta divisão permite ao 8086/88 utilizar o barramento externo para fazer a busca prévia de instruções (prefetch) enquanto a unidade de execução está ocupada com operações internas do microprocessador. As instruções são lidas da memória pela BIU como uma sequência de bytes que são armazenados em uma fila na entrada da EU. Com isto obtém-se uma melhoria do uso do barramento, porque em geral não há necessidade da busca de instrução na memória antes de executá-la, pois normalmente ela estará na fila da BIU.

A BIU é independente da EU e tenta manter cheia (com códigos) sua fila de bytes. Se duas ou mais posições estiverem vazias, a BIU executa ciclos de busca para completá-las, desde que a EU não faça qualquer pedido de serviço para a BIU. Se a EU faz um pedido de serviço para a BIU, enquanto ela estiver realizando um ciclo de busca, esse pedido só será atendido após a BIU terminar o ciclo atual. Instruções de desvio inutilizam essa fila.

OBS: A fila de bytes da BIU não deve ser confundida com a memória *cache*. Os códigos que são colocados nessa fila são as instruções que serão executadas pela EU. Os dados nunca são colocados nesta fila.

2.2 - Pinagem.

Os sinais dos barramentos internos de dados, endereços e controle se comunicam com o exterior do componente através da interface de barramento da figura 2.1. Esta interface está ligada aos pinos do chip. Por causa do número limitado de pinos disponíveis no encapsulamento escolhido para o 8086 (8088), os sinais do barramento de endereços e os sinais do barramento de dados estão **multiplexados** e compartilham os mesmos pinos do chip. A figura 2.3 ilustra a pinagem de um 8088 (figura 2.3a) e de um 8086 (figura 2.3b).

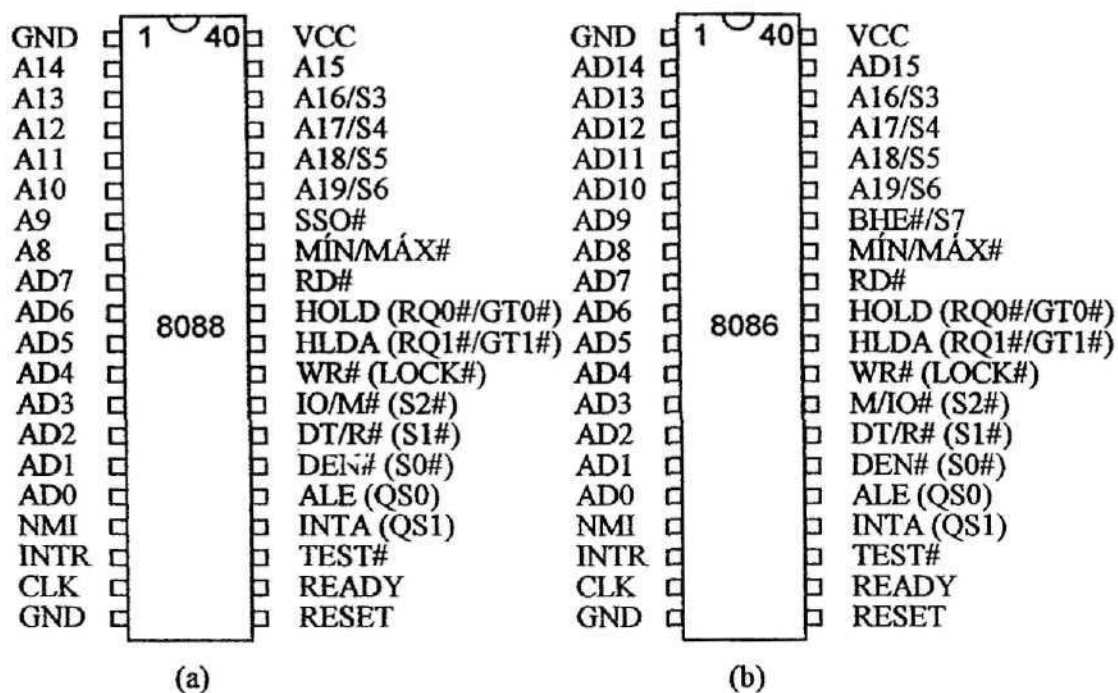


Figura 2.3 - Pinagem do 8088 e do 8086.

2.2.1 - Descrição dos pinos do 8086 (8088).

O 8086 (8088) possui dois modos de operação: O modo mínimo e o modo máximo. No modo mínimo, o microprocessador gera diretamente os sinais ALE, WR#, IO/M#, DT/R#, DEN#, INTA#, HLDA a serem descritos adiante e possui entrada HOLD. No modo máximo, o microprocessador não gera estes sinais, que são substituídos por outros. O modo máximo é usado em sistemas que contam com coprocessadores, por exemplo o coprocessador aritmético 8087, porque neste modo são gerados sinais que permitem a sincronização do coprocessador (QS0 e QS1). No entanto, os sinais ALE, WR#, IO/M#, DT/R#, DEN# e INTA# são fundamentais para o sistema e devem ser gerados externamente. Para que isto seja possível o microprocessador fornece estes sinais de uma forma codificada nos pinos S0#, S1# e S2#. O pino MÍN/MÁX# seleciona o modo de operação. Este pino é amostrado quando o microprocessador é resetado e coloca o 8086 (8088) no modo máximo se estiver conectado ao nível lógico 0. Na figura 2.3, os sinais entre parênteses são os gerados no modo máximo.

2.2.1.1 - Pinos do barramento multiplexado de dados e endereços.

AD0...7 - Transportam ora os bits 0 a 7 do barramento de endereços ora os bits 0 a 7 do barramento de dados. O sinal ALE (ALE = 1) indica o momento em que estes pinos estão transportando endereços. São pinos bidirecionais (entrada e saída).

AD8...15 - Transportam ora os bits 8 a 15 do barramento de endereços ora os bits 8 a 15 do barramento de dados. O sinal ALE (ALE = 1) indica o momento em que estes pinos estão transportando endereços. São pinos bidirecionais (entrada e saída). São exclusivos do 8086.

A8...15 - Transportam os bits 8 a 15 do barramento de endereços. O sinal ALE (ALE = 1) indica o momento em que estes pinos estão transportando endereços válidos. São pinos de saída. São exclusivos do 8088.

A16/S3, A17/S4, A18/S5, A19/S6 - Transportam ora os bits 16 a 19 do barramento de endereços ora os sinais de status S3 a S6. O sinal ALE (ALE = 1) indica o momento em que estes pinos estão transportando endereços. São pinos de saída.

BHE#/S7 - Transporta o sinal de habilitação da metade mais significativa do barramento de dados. Em operações de transferência de dados, indica que os 8 bits mais significativos do barramento de dados deverão conter dados válidos. O sinal ALE (ALE = 1) indica o momento em que este pino está transportando o sinal BHE#. É exclusivo do 8086. No 8088 este pino transporta apenas a informação de status SSO (idêntica à S7).

2.2.1.2 - Pinos de sinais de controle do barramento.

ALE (QS0) - "Address Latch Enable" ou "queue status 0" - No modo mínimo é uma saída que indica (ALE = 1) quando os pinos de barramento multiplexado de endereços/dados e endereços/status estão sendo usados para transportar endereços. No modo máximo é uma saída que informa, juntamente com QS1 o estado da fila de bytes na entrada da unidade de execução EU. Esta informação é necessária para permitir a sincronização do coprocessador.

RD# - "Read" - Indica (RD# = 0) quando o microprocessador está executando uma operação de leitura. É uma saída.

WR# (LOCK) - "Write" ou "Locked" - No modo mínimo indica (WR# = 0) quando o microprocessador está executando uma operação de escrita. No modo máximo indica (LOCK = 0) que outros processadores não devem tentar tomar o barramento pois o 8086 (8088) não atenderá a uma solicitação HOLD = 1. É uma saída.

M/IO# (S2#) - "Memory/InputOutput" ou linha de status 2 - Em um 8086 no modo mínimo indica se as linhas de endereços se referem ao espaço de endereçamento de memória (M/IO# = 1) ou ao espaço de endereços de I/O (M/IO# = 0). Quando M/IO# = 0, as linhas de endereços A16 a A19 são inválidas (o espaço de I/O é de 64K). No modo máximo o sinal S2#, juntamente com os sinais S1# e S0#, indicam qual a operação que está sendo efetuada, de modo a permitir a geração externa dos sinais ALE, WR#, IO/M#, DT/R#, DEN# e INTA#. É uma saída. **Atenção: No 8088 esta linha é invertida (IO/M#).**

DT/R# (S1#) - "Data Transmit/Receive" ou linha de status S1 - No modo mínimo indica se a operação de transferência de dados é no sentido microprocessador-para-periférico (DT/R# = 1) ou do periférico para o microprocessador (DT/R# = 0). É usado para controlar a direção dos *transceivers* (ver obs abaixo) do barramento de dados. É uma saída.

DEN# (S0#) - "Data Enable" ou linha de status S0 - No modo mínimo indica quando os pinos do barramento multiplexado estão sendo usados para transportar dados ou status. É usado para habilitar cada *transceiver* (ver obs abaixo), de forma a enviar dados ao barramento do sistema, ou então receber dados que chegam pelo barramento. É uma saída.

READY - Este pino permite avisar ao microprocessador que ele pode concluir uma operação de leitura ou escrita já iniciada (READY = 1). Usado por periféricos lentos para pedir à CPU que espere (READY = 0). É uma entrada.

OBS: Os barramentos são conjuntos de linhas condutoras por onde trafegam sinais de um determinado tipo. Normalmente, eles interconectam diversos dispositivos e, por isso, oferecem uma carga considerável. As CPUs, devido à baixa capacidade de corrente, necessitam de *drivers* (amplificadores lógicos de corrente) para entregar seus sinais ao barramento. Para retirar os sinais do barramento, usa-se um dispositivo similar, denominado

receiver. Para as linhas bidirecionais, torna-se necessário um *driver* e um *receiver*; a esse conjunto dá-se o nome de *transceiver*.

2.2.1.3 - Pinos de sinais de controle de execução.

RESET - Reinicializa o microprocessador ($RESET = 1$). Após o RESET, o CS é inicializado com FFFF e o IP com 0000. Isso faz com que a execução seja retomada a partir do endereço linear FFFF0H. O endereço de reset é portanto 16 bytes antes do topo do espaço de endereços de memória. É uma entrada.

INTR - Interrupção mascarável. Quando ativado ($INTR = 1$) solicita ao microprocessador que desvie a execução para uma rotina de tratamento de interrupções. O microprocessador termina a execução da instrução corrente e, caso as interrupções estejam habilitadas, atende o pedido. É uma entrada.

INTA# (QS1) - No modo mínimo indica a resposta (dois pulsos consecutivos de $INTA\# = 0$) do microprocessador a um pedido de interrupção INTR. Permite que um dispositivo controlador de interrupções externo forneça um número de 8 bits (vetor) para o microprocessador indicando qual interrupção, dentre 256 possíveis, foi solicitada. No modo máximo indica, juntamente com o QS0 o estado da fila de bytes na entrada da EU. É uma saída.

NMI - Entrada de interrupção não mascarável. Gera uma interrupção de tipo (vetor) fixo. Por não necessitar ler um vetor, seu atendimento é mais rápido. Não pode ser desabilitada.

2.2.1.4 - Pinos de arbitragem do barramento.

HOLD (RQ0/GT0) - "HoldRequest" ou "Request0/Grant0" - No modo mínimo é uma entrada que permite a um periférico pedir o controle do barramento. Quando o microprocessador amostra esta entrada em 1, ele termina a execução da instrução corrente e pára até que $HOLD = 0$. Enquanto estiver parado, todas as suas linhas de saídas de sinais de barramento e controle do barramento são colocadas em alta impedância. O microprocessador sinaliza que parou através da sua saída $HLDA = 1$. No modo máximo é bidirecional e funciona para ceder e tomar de volta o barramento, de modo análogo a HOLD e HLDA, exceto que os sinais são pulsos e não níveis.

HLDA (RQ1/GT1) - No modo mínimo é uma saída que fornece a resposta do microprocessador a um pedido de HOLD. No modo máximo é análogo ao pino RQ0/GT0.

2.2.1.5 - Outros.

VCC e GND - Pinos de alimentação. ($VCC = +5V$, $GND = 0V$).

CLK - Entrada de relógio. Neste pino deve ser conectado um sinal periódico com ciclo de trabalho de 1/3, ou seja 1/3 do período em nível lógico 1 e 2/3 do período em nível lógico 0. O período deve estar compreendido em uma faixa estabelecida pelo fabricante do microprocessador, podendo variar de 212ns (4.7 MHz) até 100ns (10 MHz) dependendo da versão do chip.

MÍN/MÁX# - Permite selecionar o modo mínimo ($MÍN/MÁX\# = 1$) ou o modo máximo.

TEST# - Quando executa a instrução WAIT (não confundir com *wait state*) o microprocessador amostra o pino TEST# e só prossegue a execução se $TEST\# = 1$. Senão o 8086 (8088) espera. Isto é usado para sincronizar coprocessadores. Esta entrada é ignorada quando o microprocessador executa qualquer outra instrução.

2.3 - Temporização e níveis lógicos.

Os sinais nos pinos do 8086 (8088) são compatíveis com os níveis elétricos da família TTL, embora sua construção interna não seja baseada no transistor bipolar. De fato, internamente os microprocessadores 8086 e 8088 podem usar tecnologia NMOS, CMOS, HCMOS entre outras porém todos mantêm compatibilidade com níveis TTL em seus pinos (as diferentes tecnologias levam à especificações de consumo e velocidade diferentes).

A tensão de alimentação nominal aplicada entre os pinos VCC e GND é de +5V.

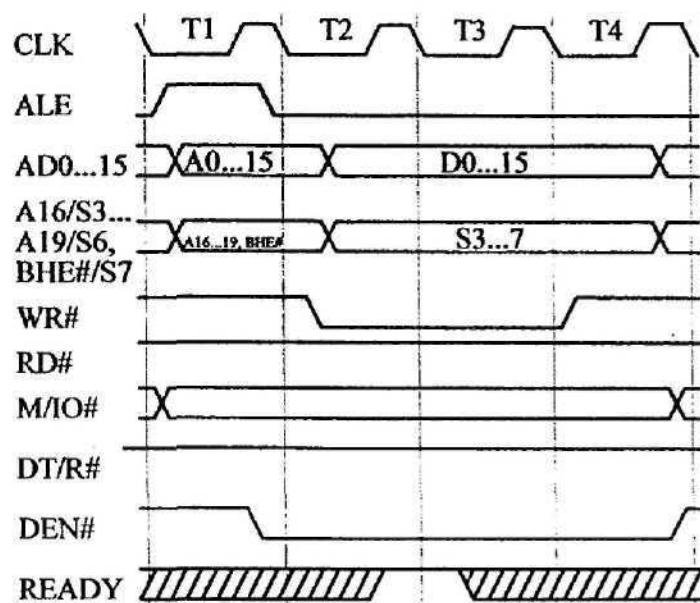
A temporização dos sinais nos pinos do 8086 (8088) é dependente do tipo de operação que ele está executando. Por isso é costume estudar os diversos tipos de operação para determinar as formas de onda.

2.3.1 - Ciclos de barramento.

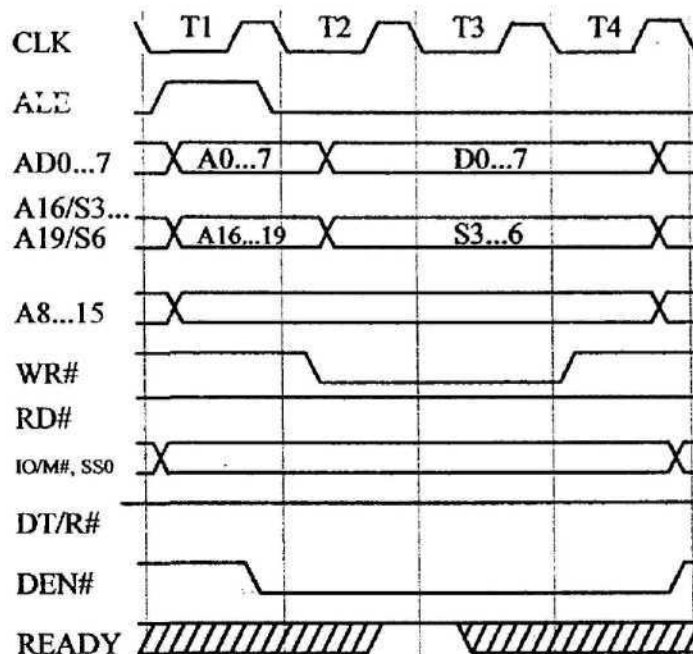
Sempre que o 8086 (8088) precisa de um dado externo ao chip, numa busca de operando ou instrução por exemplo, ele executa um ciclo de barramento. Pode-se dizer então que um ciclo de barramento é a sequência com que são gerados os diversos sinais necessários para a CPU acessar a memória e os periféricos. Um ciclo de barramento contém pelo menos 4 estados distintos: T1, T2, T3 e T4 e pode ser de **escrita**, de **leitura** ou de **atendimento à interrupção**. Quando a CPU não está executando ciclos de barramento, ela executa períodos ociosos (*idle*), denominados T1. Durante estes períodos, o conteúdo do barramento de dados geralmente fica em alta impedância.

2.3.1.1 - Ciclos de escrita.

São executados sempre que o 8086 (8088) precisa escrever um dado em uma posição de memória ou em uma porta de I/O. A figura 2.4a ilustra um ciclo de escrita em memória ou I/O do 8086 **operando no modo mínimo**. O ciclo de escrita do 8088 no modo mínimo será idêntico, a menos dos sinais AD0...15 que passam a ser AD0...7 e A8...15, do sinal M/IO# que é invertido e muda para IO/M# e do sinal BHE# que não existe. Um ciclo de escrita do 8088 está ilustrado na figura 2.4b. Para simplificar as ilustrações, apenas a ordem relativa dos sinais foi preservada, não sendo mostrados os valores exatos dos retardos entre os diversos sinais. Esta informação é importante para um projeto e é obtida do manual do fabricante do microprocessador.



(a)



(b)

Figura 2.4 - Ciclos de escrita do 8086 e do 8088 no modo mínimo.

Os ciclos de escrita, como todo ciclo de barramento, são compostos de pelo menos 4 estados, cada um com a duração de um ciclo do relógio fornecido ao pino CLK do microprocessador. Cada ciclo inicia-se em uma borda de descida do relógio CLK.

O ciclo de escrita inicia-se em T1, quando o microprocessador coloca, no seu barramento de endereços/ dados e status multiplexados, a informação de endereço. O microprocessador indica que o barramento contém informação de endereços ativando o sinal ALE. Em seguida, durante parte de T2, T3 e parte de T4, o microprocessador usa o seu

barramento de endereços/ dados e status multiplexados para transportar as informações de dados e status. Durante T2, os estados (status) estão disponíveis nas quatro linhas de endereço mais significativas e assim continuarão até o término do atual ciclo de barramento. As linhas de estado apresentam as seguintes informações codificadas:

A17 / S4	A16 / S3	SIGNIFICADO
0	0	Segmento extra
0	1	Segmento de Pilha
1	0	Segmento de código
1	1	Segmento de dados
S5 = IF	Estado da habilitação da interrupção	
S6 = 0	CPU 8086 tem controle do barramento	

Durante um ciclo de escrita, ocorre uma transferência de um dado armazenado em um registrador interno do microprocessador para um dispositivo externo ao microprocessador. Por isso, durante parte de T2, T3 e parte de T4 os pinos AD0...15 (ou AD0...7 no caso do 8088) refletem o conteúdo de algum registrador interno do microprocessador. O 8086 (8088) possui registradores de 8 bits (AL, AH, BL, BH, CL, CH, DL e DH) e 16 bits. Se a operação de escrita envolve um registrador de 8 bits, apenas metade dos bits do barramento multiplexado do 8086 estarão válidos durante a transferência de dados (que ocorre durante parte de T2, todo T3 e parte de T4). Por outro lado, se o microprocessador é o 8088 e a operação de escrita envolve um registrador de 16 bits (AX, BX, CX, DX, BP, SP, SI, DI, CS, DS, SS, ES, FR, IP), o conteúdo do mesmo não pode ser passado pelos 8 bits do barramento de dados do 8088 de uma só vez. Por isso, serão necessários dois ciclos de escrita, um para o byte menos significativo e outro para o byte mais significativo do registrador. Por causa disso, uma CPU baseada no microprocessador 8086 será mais rápida do que uma CPU baseada em 8088.

O sinal DEN# é ativado (DEN# = 0) durante parte de T1, T2, T3 e parte de T4 para indicar que o barramento multiplexado conterá dados. Observa-se na figura 2.4 que este sinal não é simplesmente o sinal ALE invertido, sendo ativado antes mesmo que ALE volte a zero. Isto é feito para melhorar a temporização em conexões típicas de componentes auxiliares usados para separar o barramento multiplexado em dois barramentos, um de endereços e outro de dados.

O sinal DT/R# é ativado (DT/R# = 1) antes mesmo do início de T1 e permanece válido até o fim do ciclo de escrita, indicando que a transferência de dados se dará no sentido do microprocessador para um periférico.

O sinal WR# é ativado (WR# = 0) durante parte de T2, T3 e parte de T4 para indicar a operação de escrita. Nota-se que o conteúdo do barramento de dados permanece inalterado por um certo tempo após a desativação do sinal de escrita. Isto é feito para atender à especificação de tempo de retenção ("hold time") de memórias e periféricos, facilitando a interligação destes dispositivos com o microprocessador. Para a maior parte das memórias e dispositivos periféricos, a escrita será feita na borda de subida do sinal WR#.

Observa-se, pela figura 2.4, que o sinal WR# possui duração entre 1 e 3 ciclos de relógio. De fato, consultando o manual do fabricante INTEL para o 8086-1 (8086 com relógio de 10 MHz), verifica-se que a duração do sinal WR# é igual a duas vezes o período do relógio CLK menos 35 ns (o que resulta em 165 ns à frequência nominal de relógio). Se for necessário conectar ao microprocessador um periférico ou um dispositivo de memória que necessite de uma duração mais longa do sinal de escrita WR#, é possível aumentar a duração mínima básica mediante a inserção de estados de espera ou "wait states".

O sinal READY é usado para controlar a inserção de "wait states". No início do estado T3 de um ciclo de barramento, a entrada READY é amostrada. Se READY = 1, o estado seguinte à T3 será o T4, como na figura 2.4. Se contudo READY é amostrada em

zero, o estado seguinte à T3 será um TW. No início de cada TW, a entrada READY é novamente amostrada e caso seja zero um outro estado TW seguirá o estado TW corrente. Somente quando a entrada READY for amostrada em 1 no início de um TW (ou de um T3 como em 2.4) o estado seguinte será um T4, finalizando o ciclo. A figura 2.5 ilustra um ciclo de escrita para o 8086 com um "wait state". O ciclo para o 8088 será semelhante, com diferenças análogas às exibidas pelos ciclos de escrita do 8086 e do 8088 na figura 2.4 sendo por isso omitido.

A entrada READY é uma entrada síncrona e por isso deve-se respeitar tempos de setup e hold em relação às bordas de descida de CLK especificadas pelo fabricante. Caso READY deva ser controlada por sinais não sincronizados, é necessário primeiro sincronizar o sinal de controle (usando FLIP-FLOPS) antes de conectá-lo ao pino READY.

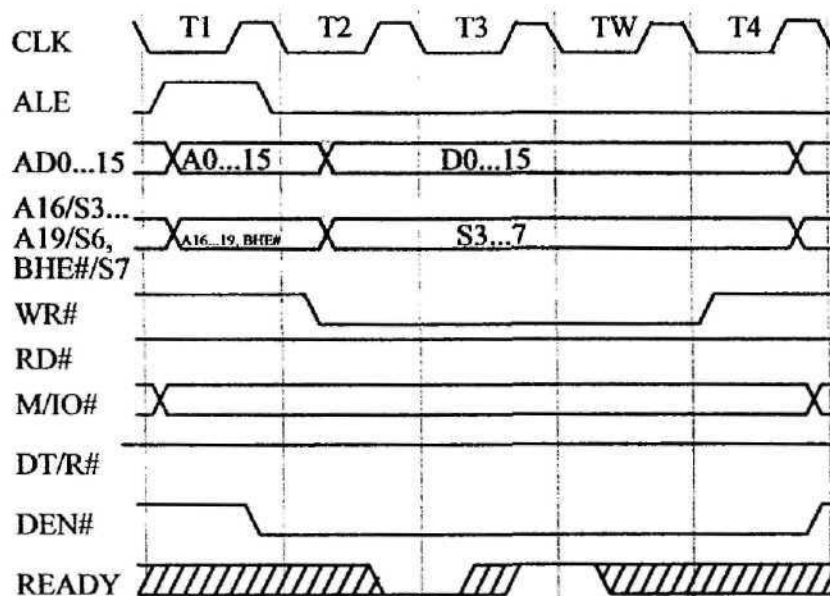


Figura 2.5 - Ciclo de escrita do 8086 no modo mínimo com um "wait state".

No modo máximo, os sinais ALE, DEN#, DT/R# e WR# da figura 2.4 não são gerados pelo microprocessador. Ao invés disso, o 8086 (8088) fornece os sinais S0#, S1# e S2# que, juntamente com CLK, são usados para gerá-los externamente (o chip 8288 da INTEL pode ser usado para este fim). A figura 2.6 ilustra um ciclo de escrita do 8086 no modo máximo.

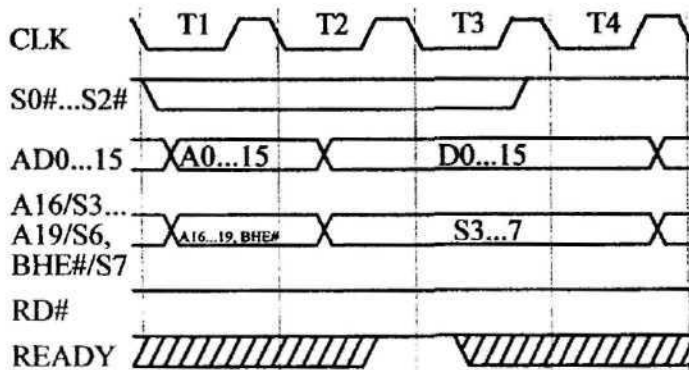


Figura 2.6 - Ciclo de escrita para o 8086 no modo máximo.

A interpretação dos sinais S0#, S1# e S2# é dada na tabela 2.1.

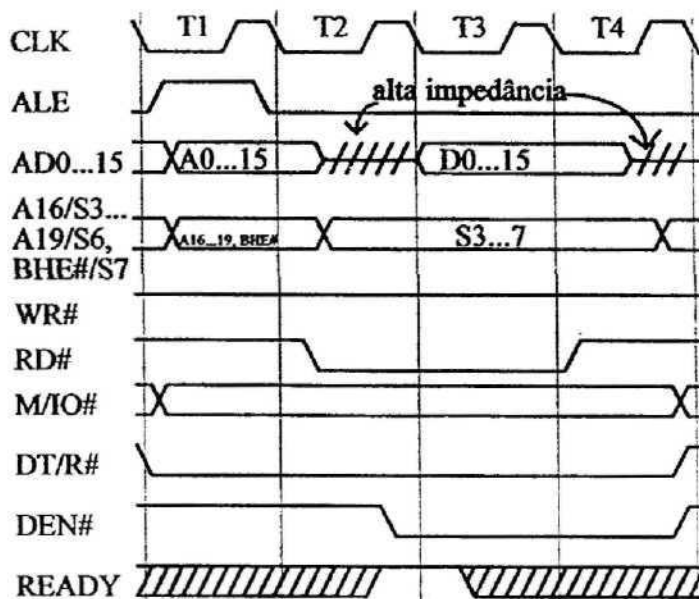
S2#	S1#	S0#	Tipo de ciclo de barramento
0	0	0	Atendimento de interrupção
0	0	1	Leitura em I/O
0	1	0	Escrita em I/O
0	1	1	Halt
1	0	0	Busca de instrução (<i>fetch</i>)
1	0	1	Leitura de memória
1	1	0	Escrita em memória
1	1	1	Inativo (sem ciclo de barramento)

Tabela 2.1 - Sinais S0#, S1# e S2#.

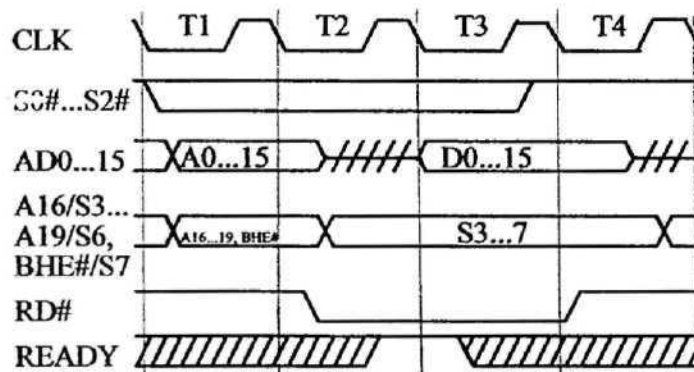
Quando o 8086 inicia um ciclo de barramento operando no modo máximo, as linhas de status S0#, S1# e S2# mudam de 111 (Inativo) para um valor correspondente ao tipo de ciclo iniciado. Neste momento um circuito sequencial síncrono externo ao 8086 (por exemplo, o 8288) inicia a geração dos sinais ALE, DT/R#, DEN#, WR#, M/IO# e INTA#, de acordo com o código colocado nas linhas de status. A temporização destes sinais é semelhante à dos sinais gerados pelo próprio microprocessador operando no modo mínimo. O ciclo só é finalizado quando as linhas de status voltam ao estado inativo 111. Por isso, estas linhas são desativadas no T3 ou em um TW, em função de haver "wait states" ou não.

2.3.1.2 - Ciclos de leitura.

São executados sempre que o 8086 (8088) precisa ler um dado em uma posição de memória ou em uma porta de I/O. A figura 2.7a ilustra um ciclo de leitura em memória ou I/O do 8086 operando no modo mínimo. A figura 2.7b ilustra um ciclo de leitura em memória ou I/O do 8086 operando no modo máximo. Durante uma leitura ocorre uma transferência de um dado em um periférico ou posição de memória para um registrador interno, de 8 ou 16 bits, do 8086. As diferenças entre leituras do 8086 e do 8088 são análogas às diferenças entre as escritas, por isso os ciclos de leitura do 8088 não estão ilustrados.



(a)



(b)

Figura 2.7 - Ciclos de leitura do 8086 nos modos mínimo e máximo.

Observa-se que o sinal DEN# é ativado durante T2 nos ciclos de leitura, enquanto o mesmo sinal é ativado durante T1 nos ciclos de escrita. Isto ocorre para evitar conflitos no barramento quando este sinal é usado para habilitar buffers bidirecionais conectados no barramento de dados. Esta aplicação será abordada na seção 2.3.2.

Após a ativação do sinal RD#, o 8086 (8088) espera que o dispositivo periférico ou de memória sendo acessado coloque o dado a ser lido pelo microprocessador no barramento de dados e endereços multiplexado. Por isso estas linhas são colocadas em alta impedância pelo microprocessador no início de T2, logo após a transferência de endereços efetuada durante T1 e início de T2.

2.3.1.3 - Ciclos de atendimento de interrupção.

Quando o 8086 (8088) recebe um pedido de interrupção pelo seu pino INTR, ele responde com um ciclo de atendimento de interrupção (se as interrupções estiverem habilitadas). A figura 2.8 ilustra um ciclo de atendimento de interrupção de um 8086 operando no modo mínimo.

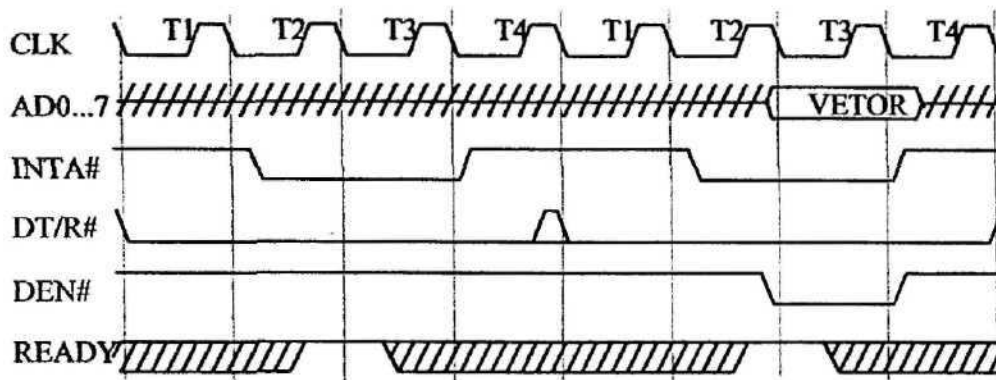


Figura 2.8 - Ciclo de atendimento de interrupção do 8086 no modo mínimo.

No modo máximo, o ciclo é semelhante ao da figura 2.8, porém os sinais INTA#, DT/R# e DEN# são gerados por um dispositivo externo (8288) a partir dos sinais de status S0#, S1# e S2# e do sinal de relógio CLK.

O ciclo de atendimento de interrupção dura pelo menos 8 períodos do relógio CLK. Neste intervalo, o sinal INTA# é ativado ($INTA\# = 0$) duas vezes. Após a segunda ativação, o microprocessador lê nos 8 bits menos significativos do barramento de dados, AD0...7, um byte que corresponde ao número ou tipo de interrupção solicitada (esse número é chamado de **vetor de interrupção**). Isso permite ao microprocessador distinguir até 256 tipos diferentes de interrupções. Cabe a um dispositivo **controlador de interrupções** externo (por exemplo o 8259) responder ao segundo pulso INTA# e colocar o vetor de interrupção no barramento. A seguinte sequência de eventos ocorre após a leitura do vetor pelo microprocessador:

- O microprocessador multiplica internamente o vetor lido por 4 (desloca duas vezes para a esquerda).
- O microprocessador lê 4 bytes em endereços consecutivos da memória, começando no endereço linear dado por $4 \times \text{vetor}$ previamente calculado.
- O microprocessador salva o contexto de execução (registrador de flags FR, seguido de CS e IP) na pilha e carrega o CS com os dois primeiros bytes lidos e o IP com os dois últimos bytes lidos, deste modo desviando a execução para uma subrotina cujo endereço é dado pelos 4 bytes.

Esse mecanismo "reserva" o primeiro Kilobyte de memória (endereços lineares 00000H até 003FFH) para os endereços das 256 rotinas de tratamento de interrupções. A figura 2.9 ilustra o espaço de memória com estas posições reservadas. Observa-se que o intervalo de endereços de memória de 00000H até 003FFH (ou qualquer subintervalo deste) podem ser usados normalmente, desde que não se use interrupções no sistema (ou pelo menos não se use as interrupções cujos endereços fiquem no subintervalo).

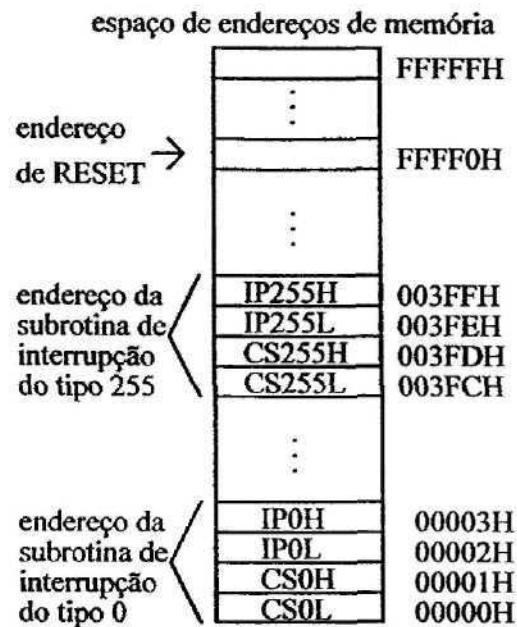


Figura 2.9 - Tabela de endereços de interrupções na memória.

As interrupções podem ser usadas para diversas finalidades como tratamento de teclado, controle de comunicações seriais, etc. e podem ser livremente definidas pelo projetista do sistema. Existem contudo algumas interrupções reservadas pela INTEL para usos específicos, como por exemplo a interrupção tipo 0, que está associada à divisão por zero. Os tipos reservados serão estudados posteriormente no capítulo de software do 8086.