

BARRAMENTOS EM MICRO COMPUTADORES (I)

Aquilino R. Leal – Professor da Universidade Estácio de Sá e do Instituto de Tecnologia ORT

O PROPÓSITO DESTA PUBLICAÇÃO É APRESENTAR UMA NOÇÃO DO QUE É UM BARRAMENTO BEM AS PRINCIPAIS CARACTERÍSTICAS DOS BARRAMENTOS TÍPICOS DA ATUALIDADE ASSIM COMO DOS PADRÕES MAIS ANTIGOS E A RESPECTIVA EVOLUÇÃO.

Definições e classificação dos barramentos

Um barramento (*omnibus*) é um conjunto de fios, condutores etc. que operam em paralelo tal qual as pistas de uma grande avenida cuja função é interligar duas localidades ou regiões geográficas. Semelhantemente às avenidas, os barramentos são utilizados para interligar os componentes de um computador - Figura 1

Em verdade um barramento nada mais é do que uma coleção de fios utilizada para transmitir sinais em paralelo, ou seja, simultaneamente. Em uma linguagem simples: um barramento é um caminho elétrico comum entre múltiplos dispositivos. Um exemplo típico é o barramento de sistema presente em todos os microcomputadores, o qual consiste em umas poucas dezenas de fios de cobre paralelos gravados na placa mãe (*motherboard*), com conectores regularmente espaçados onde serão inseridas as placas inerentes a dispositivos de E/S (I/O)¹.

Os barramentos também destinam-se a outras funções e/ou aplicações tais como interligar um microprocessador (CPU – Unidade Central de Processamento) a um ou mais coprocessadores ou a memórias locais etc. conforme ilustrado na Figura 1

Também não se pode ignorar que dentro da própria pastilha do microprocessador existem vários barramentos para interligar os componentes internos que o constituem. Uma aplicação típica é possibilitar que o conteúdo de um registrador² possa ser copiado para um outro registrador e vice-versa.

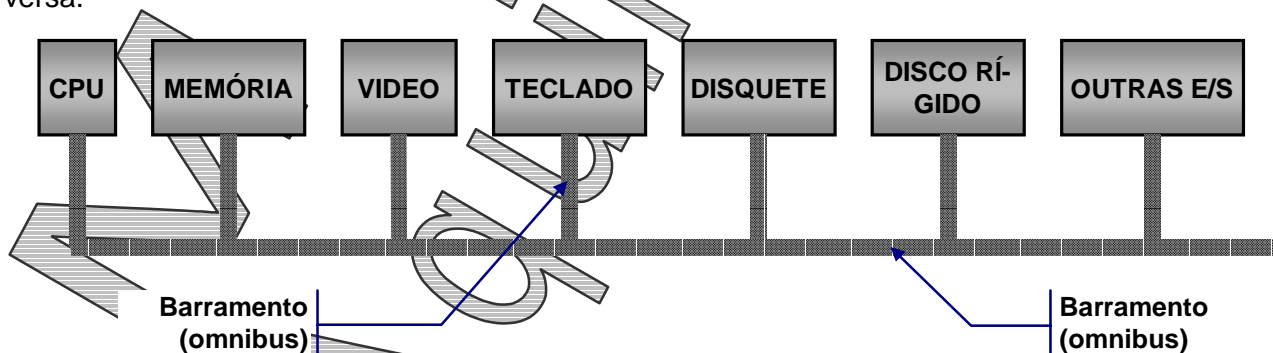


Figura 1

Na literatura os barramentos costumam ser representados por setas grossas, bidirecionais ou unidirecionais indicando o sentido do fluxo dos dados ou informações; em algumas situações associa-se a essas setas um número que diz quantas vias (fios, linhas etc.) o barramento em questão

¹ Entrada/Saída, em inglês *Input/Output*.

disponibiliza para o tráfego de informações (usualmente binárias).

Em verdade, o barramento de um sistema de computação é o elemento responsável pela interligação dos componentes, conduzindo, de modo sincronizado, o fluxo de informações (dados, endereços e controle) de uns para os outros de acordo com uma programação de atividades previamente definida pela **unidade de controle (UC – em inglês CU)** do microprocessador.

Conforme mencionado acima, basicamente existem dois tipos de barramentos:

- **barramento externo** → interliga os diversos componentes de um sistema computacional tais como UCP, ou CPU, memória, unidades de entrada/saída etc. (Figura 1) e
- **barramento interno** → interliga elementos no interior de um componente (pastilha), como, por exemplo, os registradores, ou registros, de um microprocessador.

Além da classificação acima, os barramentos também podem ser ordenados quanto ao tipo de informações que neles fluem. A saber:

- **barramento de dados (data bus)** → o barramento transporta as instruções e os valores (numéricos ou não) manipulados pelas instruções;
- **barramento de endereços (address bus)** → este tipo de barramento transporta valores que indicam a localização dos dados ou dispositivos de E/S etc. e
- **barramento de controle (control bus)** → responsável pela transferência de sinais (de controle) como *READ*, *WRITE*, *HOLD*, de início de operação aritmética, de interrupção, de sincronização, de reciclagem (*reset*) entre outros.

Algumas publicações mencionam a existência do **barramento de energia** (em inglês *power bus*), cuja coleção de fios é usada para distribuição de energia para os vários componentes de um sistema computacional. Outras publicações, ainda que não se concorde com elas, referem-se ao barramento de dados como **bus de direções** (Figura 2), melhor seria **barramento de direções** ou, utilizar de uma vez por todas o termo em inglês.

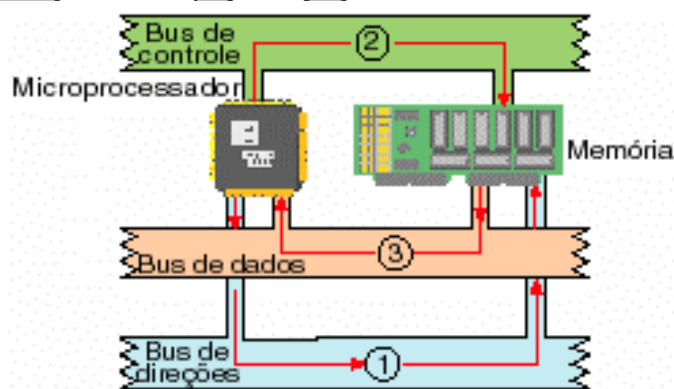


Figura 2

Quanto ao sentido do fluxo das informações pelas vias, os barramentos classificam-se em:

- **barramento unidirecional** → só pode transferir dados em um sentido; sendo tipicamente utilizados para interligar dois dispositivos um dos quais é sempre a origem e o outro é sempre o

² Unidade de armazenamento que faz parte do *hardware* interno da CPU, funcionando à mesma velocidade que esta.

destino;

- **barramento bidirecional** → pode transferir dados nos dois sentidos, mas não em ambos simultaneamente; eles são tipicamente utilizados quando qualquer um dos dispositivos pode ser a origem e qualquer outro pode ser o destino.

À guisa de ilustração a Figura 3 apresenta a estrutura básica de um sistema computacional fundamentado em uma CPU essencialmente composta por uma **unidade de controle (UC)**, responsável pela busca das instruções na memória de programa, interpretação dessas instruções e geração dos sinais de controle e de sincronismo necessários para tornar o processo automático; **registros** (dispositivos de armazenamento temporário, onde a informação pode ser colocada em um determinado instante e de lá copiada quando isso se fizer necessário) e uma **unidade lógico-aritmética (ULA)** responsável pela execução de operações lógicas (E, OU, NOT etc.) e aritméticas (adição, subtração etc.) - cabe à unidade de controle estabelecer o tipo de função que a ULA deve desenvolver em cada momento. Esses três blocos constituintes da CPU comunicam-se através de barramentos internos: um unidirecional e outro bidirecional conforme indicam as setas apresentadas na Figura 3.

A comunicação da CPU, Figura 3, com o restante do sistema está sendo feita através de um barramento de dados bidirecional e de um barramento de endereços unidirecional (saída) – por questão de simplificação neste diagrama em blocos não foi representado o barramento de controle.

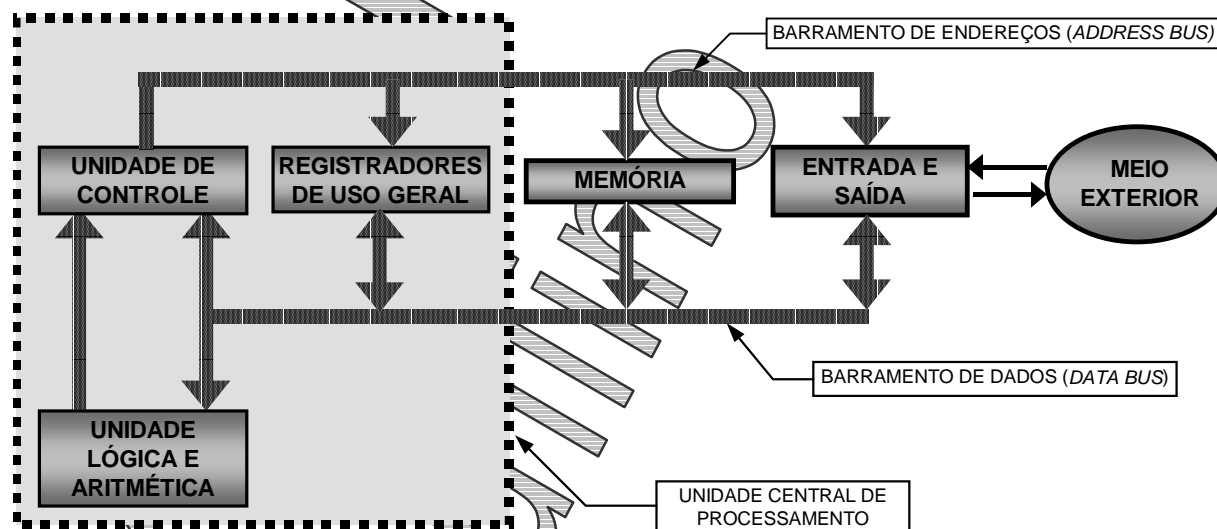


Figura 3

A título de ilustração a Figura 4 mostra, de forma bastante simplória, a estrutura interna de uma CPU típica interligada à memória do sistema através de um barramento específico (barramento local) aí designado, um tanto imprópriamente, por 'bus do sistema'.

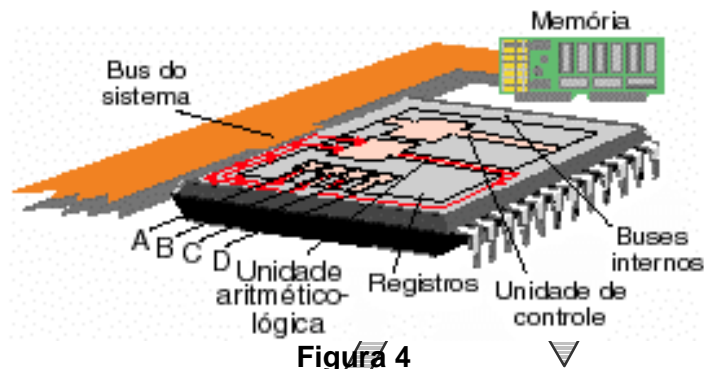


Figura 4

Como tanto a CPU, como a memória, dispositivos de entrada e saída podem colocar, ou retirar, informações no barramento de dados, havendo necessidade desses dispositivos conectarem-se e desconectarem-se eletricamente dos barramentos aos quais estão fisicamente ligados. Um barramento cujos dispositivos possuem essa capacidade (conexão ou não) é chamado **barramento tri state** pois cada linha pode assumir o valor lógico **0**, **1** ou estar **desconectada** (três estados, daí a designação *tri state*). Barramentos *tri state* são comumente usados quando os dispositivos ligados ao barramento potencialmente podem colocar informações neste último, ou sejam, têm a capacidade de “tomar” o barramento - essas conexões normalmente são feitas ou desfeitas em pouco tempo, algo em torno de uns poucos nanosegundos ou menos.

A **largura**, ou tamanho, de um barramento é uma unidade de medida que caracteriza a quantidade de informações (bits de uma forma geral) que pode fluir simultaneamente pelo barramento. No caso de fiação, conforme visto, consiste, a priori, na quantidade de fios paralelos existentes no barramento ao passo que, em circuitos impressos (placas), consiste nos traços impressos na placa com material condutor (filétes de cobre) por onde flui a corrente elétrica, em última estância, as informações. Esta largura, maior ou menor, também se constitui em um dos elementos que afetam a medida de desempenho de um sistema, juntamente com a duração (largura) de cada bit ou sinal - a taxa de transferência, ‘velocidade’ do fluxo de informações, é geralmente especificada em bits (ou Kbits ou Mbits etc.) por segundo e depende, fundamentalmente, da largura do barramento: quanto maior esta maior será a taxa.

O intervalo de tempo requerido para mover um grupo de bits (tantos quanto a quantidade de bits definida pela largura do barramento) ao longo do barramento, é denominado **ciclo de tempo** do barramento ou, simplesmente, **ciclo de barramento** (*bus cycle*), semelhantemente ao que entendemos para o ciclo do processador e para o ciclo de memória.

Operação de um barramento e protocolos

Para entender o funcionamento do barramento de um computador é preciso enfatizar o aspecto de compartilhamento que caracteriza aquele componente, ou seja, como um barramento interliga diversos componentes, seja ele um barramento externo ou interno, as informações só podem fluir uma de cada vez, senão haverá colisão entre os sinais elétricos e o resultado, obviamente, será ininteligível qualquer que seja o destinatário.

Em outras palavras, se, por exemplo, a memória principal está enviando dados para a memória secundária (componente de E/S como disco magnético por exemplo), os demais componentes

têm de esperar a liberação do barramento para utilizá-lo. Este compartilhamento (um caminho para vários usuários) implica na necessidade de definição de regras bem explícitas de acesso ao barramento por um usuário (quando e como terminar o acesso) e de comunicação entre eles (como interrogar um componente destinatário, que resposta deve ser enviada, quanto dura a comunicação etc.). Estas regras são denominadas **protocolos**, sendo, no caso, **protocolos de barramento**, os quais são usualmente implementados através de sinais de controle e exata sincronização entre eles.

Portanto, o barramento não se constitui tão somente na tal fiação já mencionada mas também na unidade de controle do barramento que administra o acesso e as transferências.

Mas isso não é tudo. Devem existir especificações mecânicas e elétricas de modo que placas de terceiros caibam no bastidor e tenham conectores que encaixem naqueles da placa mãe, tanto em termos mecânicos quanto em termos elétricos.

Os fabricantes têm procurado criar uma padronização na definição desses protocolos e produtos visando com isso que componentes fabricados por qualquer um não tenham dificuldade de se conectar à CPU; infelizmente o sucesso total ainda está longe, ou seja, um só padrão em todo o mercado, o que é mais do que desejável.

Até há pouco tempo vários produtos e protocolos de barramento já estavam definidos, alguns tiveram pouca aceitação, outros não, alguns são proprietários (são definidos por uma única empresa que cobra *royalties* pelo seu licenciamento de uso) e outros não. À guisa de informação seguem alguns tipos de barramento entre os muitos existentes:

- UNIBUS → definido pela Digital Equipment Co. (DEC);
- ISA → *Industry Standard Architecture*, definido pela IBM para o PC-AT e adotado por toda a indústria, tendo sido um dos mais difundidos e utilizados;
- MCA → *Micro Channel Architecture*, definido pela IBM para os sistemas PS-2;
- EISA → *Extended ISA*, definido por um grupo de fabricantes para se opor ao MCA da IBM;
- SBI → *Synchronous Backplane Interconnect*, definido pela DEC;
- PCI → *Peripheral Component Interconnect*;
- USB → *Universal Serial Bus*;
- AGP → *Accelerated Graphics Port*;
- VLB → *VESA³ Local Bus*;
- IrDA → *Infrared Developers Association*.

O método de controle do acesso ao barramento pode ser feito utilizando o **método mestre/escravo** (*master/slave*) que é um dos métodos mais utilizados. Um dos componentes, em geral a CPU, é o mestre⁴, sendo o único que pode acessar o barramento seja para colocar informações (escrita) para um determinado componente (periférico), seja para obter informações (leitura) de um outro

³ Video Electronic Standards Association.

⁴ Em verdade, alguns dispositivos que se conectam a um barramento são ativos e podem iniciar transferências pelo barramento, enquanto outros são passivos e esperam por solicitações. Os ativos são chamados mestres e os passivos escravos. Quando, por exemplo, a CPU ordena que um controlador de disco leia ou escreva um bloco, a CPU está agindo com um mestre e o controlador de disco está agindo como escravo; entretanto mais tarde, o controlador de disco pode agir como mestre quando ele comanda que a memória aceite as informações que ele está lendo da unidade de disco.

componente. Os demais componentes são os escravos.

Esse método tem a vantagem de ser simples e barato de ser implementado, mas possui uma grande desvantagem: toda a comunicação é realizada via mestre (CPU), se há volume de dados envolvido nas transferências poderá surgir um gargalo no funcionamento do barramento como, por exemplo, na transferência de dados entre a memória secundária (discos, fitas etc.) e a memória principal (memória RAM). Uma alternativa consiste em se disporem múltiplos mestres no sistema e não somente um. Esse novo método utiliza um dispositivo controlador denominado **Acesso Direto à Memória** ou DMA (*Direct Memory Access*) o qual é utilizado pela maioria dos modernos sistemas. Ele se caracteriza pela possibilidade de um dos componentes que compartilham o barramento poder ganhar controle deste para transmitir/receber informações.

Esse último processo, ainda que libere o microprocessador para realizar outras tarefas enquanto uma transferência esteja sendo realizada pelo DMA (por exemplo entre a memória principal e a memória secundária), requer maior complexidade nos circuitos de controle do barramento e não componentes do sistema para administrar as diversas solicitações simultâneas de acesso ao barramento, bem como os correspondentes sinais de controle que são mais complexos.

Um outro elemento a considerar-se no aspecto de projeto de barramentos, refere-se ao modo pelo qual os eventos são coordenados no barramento, isto é a sua temporização. Existem duas técnicas: **operação síncrona** (barramento síncrono) e **operação assíncrona** (barramento assíncrono).

Na operação síncrona, os pulsos emitidos pelo relógio regem o surgimento/desaparecimento dos sinais nas diversas linhas do barramento, ou seja, o relógio sincroniza o funcionamento do barramento e a ocorrência e duração de todos os eventos. Para tal, o barramento disponibiliza uma linha para o relógio mestre por onde circulam os pulsos gerados por aquele dispositivo, sendo cada pulso, denominado **ciclo de relógio** ou **ciclo de barramento**, o qual corresponde ao inverso do valor da frequência do relógio (*clock*) de modo que todas as atividades gastam um número inteiro desses ciclos.

Exemplificando: um relógio (*clock*) de 6,144Hz produz um ciclo de barramento de aproximadamente 162,8 ns ($1/6,144$) e se a escrita de um byte na memória utiliza três ciclos de barramento esse acesso tem a duração de 488,3 ns ($3 \times 162,8$ ns).

Na operação do tipo assíncrona não há relógio sincronizador nem eventos com duração certa de um ciclo de barramento pois não há relógio (no sentido de não haver pulso de relógio para o barramento). Nesta modalidade cada evento depende somente da ocorrência do evento anterior que pode ter duração em tempo diferente, ou não, da duração do próximo evento; deste modo os ciclos de barramento podem ter qualquer duração requerida e não precisam ser os mesmos entre todos os pares de dispositivos.

Na prática o barramento síncrono é simples de implementar e testar justamente devido à sua natureza inflexível (periodicidade) quanto ao tempo; em conseqüência qualquer atividade entre mestre e escravo somente pode realizar-se em quantidades fixas de tempo, o que acaba tornando-se uma desvantagem porque o barramento pode ter problemas como, por exemplo, ao trabalhar com

dispositivos que tenham tempos de transferências diferentes – nesta situação há necessidade da inserção de **estados de espera** (*wait states*) que nada mais são do que ciclos de barramento extras.

Esse inconveniente não ocorre com o barramento assíncrono que, por não depender de relógio com intervalos fixos de tempo, pode conviver com dispositivos que velocidade baixa e alta que utilizam tecnologia antiga e avançada, tirando vantagens das melhorias na tecnologia. Em contrapartida requer sinais adicionais para prover a devida sincronização dos eventos.

Para “fechar” por o protocolo completo (*handshake*)⁵ nesta técnica assíncrona é utilizado o par de sinais a seguir os quais são ativados em nível baixo:

- $\overline{\text{MSYN}}$ → **Master SYN**chronization - sincronização do mestre e
- $\overline{\text{SSYN}}$ → **Slave SYN**chronization - sincronização do escravo.

Esses dois sinais operam segundo um processo que basicamente consiste de quatro eventos:

- inicialmente $\overline{\text{MSYN}}$ é ativado pelo mestre para pedir uma ação a um escravo;
- o escravo ativa o sinal $\overline{\text{SSYN}}$ em resposta ao $\overline{\text{MSYN}}$, informando que ele está disponível;
- ao encerrar a tarefa, o sinal $\overline{\text{MSYN}}$ é desativado pelo mestre e, finalmente,
- o sinal $\overline{\text{SSYN}}$ é desativado pelo escravo em resposta à desativação do sinal $\overline{\text{MSYN}}$ originário do mestre.

Interfaceamento

Os sinais digitais oriundos da CPU freqüentemente não têm energia suficiente para alimentar um barramento, em especial se ele for relativamente longo e possuir muitos dispositivos a ele conectados. Por esse motivo a maioria dos mestres de barramento estão interligados a ele, barramento, através de pastilhas específicas chamadas **acionadores de barramentos** (*bus drivers*), essencialmente amplificadores digitais.

De forma similar, a maioria dos escravos estão conectados ao barramento através de **receptores de barramento** (*bus receiver*). Já para os dispositivos que podem agir tanto como mestre quanto como escravo, é utilizada uma pastilha chamada **transceptor de barramento** (*bus transceiver*).

Todas essas pastilhas de interfaceamento com o barramento são freqüentemente dispositivos *tri-state* para permitir que elas flutuem (sejam desconectadas – alta impedância) quando não é necessária a sua ação; em alguns casos também são usadas pastilhas do tipo **coletor aberto** que atingem efeito similar.

O *hardware*, no entanto, não para por aí. Assim como em uma CPU, um barramento também tem linhas de endereço, dados e controle. Entretanto, não existe necessariamente uma correspondência um-para-um entre os sinais do microprocessador e os sinais do barramento. Por exemplo, um

⁵ Deve estar claro que os *handshakes* completos são independentes do tempo. Cada evento é causado por um anterior e não por um pulso de relógio. Se um par mestre-escravo é lento, isto não afeta de forma alguma um par subsequente mestre-escravo que seja muito mais rápido.

microprocessador pode ter três pinos, ou terminais, cujos níveis lógicos codificam se ele está realizando uma leitura de memória, uma escrita na memória, uma leitura de E/S, uma escrita de E/S ou alguma outra operação, permitindo a codificação de até um máximo de oito possibilidades ($2^3 = 8$).

Por outro lado um barramento típico terá uma linha para leitura de memória, uma segunda linha para escrita em memória, uma terceira para leitura de E/S, uma quarta para escrita de E/S e assim por diante. Portanto, faz-se necessário um circuito decodificador entre essa CPU e tal barramento para converter o sinal codificado, no caso de três *bits*, em sinais separados que podem acionar as linhas do barramento.

A Figura 5 mostra um circuito decodificador comercial (CI⁶ 74LS138, tecnologia TTL) com uma tríade de entradas (**A**, **B** e **C**) em que apenas uma das oito saídas fica ativa (em nível baixo) de acordo com binário aplicado a essas entradas (as entradas **E_i** são utilizadas para selecionar a pastilha, possibilitando o seu cascadeamento).

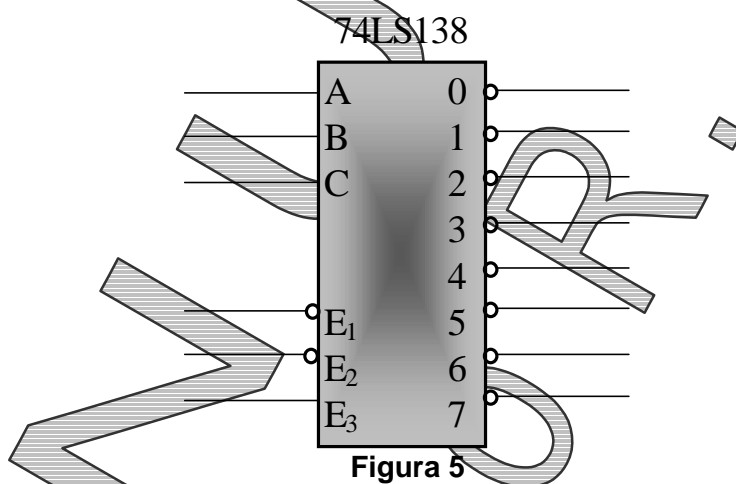


Figura 5

Barramento Local

Em um microcomputador de uso pessoal o seu principal barramento é o **barramento local**, ou seja, a via de comunicação que possibilita a comunicação do processador aos circuitos primordiais da placa-mãe como a memória RAM, a memória cache e o *chipset*⁷ - dentre os diversos circuitos presentes no *chipset* merecem destaque o controlador de memória RAM, o controlador de cache⁸ além do controlador de barramento no caso de existirem outros barramentos no micro, o que na prática certamente ocorre.

O barramento local é o mais rápido, pois os circuitos se comunicarão com o processador com seu desempenho máximo. Entretanto o barramento local não é padronizado: cada processador deverá utilizar o seu próprio modelo, de acordo com as suas características; aliás, é por este motivo que cada processador necessita de um modelo de placa-mãe diferente.

Como o barramento local é utilizado na comunicação do processador com os circuitos básicos e que requerem velocidade (especialmente a memória RAM e o cache de memória), esse barra-

⁶ Circuito Integrado.

⁷ Chipset é conjunto de circuitos integrados de apoio existentes na placa-mãe.

⁸ Memória estática de rápido acesso comparativamente à velocidade de funcionamento da CPU.

mento é totalmente transparente ao usuário. Simploriamente pode-se dizer que o barramento local está na placa-mãe e... nada mais.

Assim como ocorre com uma CPU típica, o barramento local (e a maioria dos barramentos de expansão) pode ser dividido em três grupos, a saber:

- **barramento de dados;**
- **barramento de endereços e**
- **barramento de controle.**

O processador Pentium, por exemplo, apresenta um barramento de dados de 64 bits (oito bytes⁹), o que quer dizer que o barramento local terá o seu barramento de dados de 64 bits. Como consequência, o acesso à memória será feito a 64 bits por vez. Da mesma forma, quando se diz que o processador trabalha externamente a 66 MHz (como o Pentium 200 por exemplo), isso significa que é esta a frequência de operação do barramento local.

No caso do barramento local típico de 64 bits a 66 MHz a taxa de transferência máxima entre o processador e memória (RAM e cache) será de 528 MB/s¹⁰ que é o resultado da expressão $[64 \text{ bits} \times 66 \text{ MB/s}] / 8 \text{ bits}^{11}$. No caso do barramento operar a 75 MHz, a taxa de transferência será de 600 MB/s (simplesmente $75 \times 8 \text{ MB/s}$). Já no caso de um barramento operando a 83 MHz (no caso de um *overclock*), a taxa passará a ser 664 MB/s. E, finalmente no caso dos processadores que operam externamente a 100 MHz, a taxa de transferência é de 800 MB/s.

Essa é uma ótima oportunidade para entender como um 6x86-PR200, que trabalha internamente somente a 150 MHz, consegue ser mais rápido que um Pentium 200: o Pentium 200, por operar a 66 MHz, “conversa” com a memória (RAM e cache) a 528 MB/s, enquanto o 6x86-PR200, por operar a 75 MHz, “conversa” com a memória a 600 MB/s.

Da mesma forma, não é recomendado o uso hoje em dia de processadores com barramento local abaixo de 66 MHz - como o Pentium-120, Pentium-150, K5-PR120 e 6x86-PR150+ por apresentarem baixo desempenho e um gargalo considerável – em verdade 'isso' nem mais existe!!

Aqui é válido ressaltar que o termo usual *slot* e barramento não são sinônimos. Assim como o barramento passa por todos os circuitos existentes no microcomputador, ele passa também por este *slot* (conector). O *slot* tem entradas e saídas para o caminho de dados, para o caminho de endereços e para o caminho de controle. Todos os *slots* do microcomputador são absolutamente iguais (em termos de função) e, portanto, não há *slots* especiais para esta ou aquela interface: qualquer interface periférica pode ser encaixada em qualquer um dos *slots* desde que, é obvio, ela esteja vaga e seja mecânica e eletricamente compatível. Para que uma simples placa de vídeo ou um disco rígido possam ser utilizados em qualquer micro, independentemente do processador instalado (ou seja, independentemente do modelo de barramento local empregado), são utilizados diversos modelos de barramentos de expansão conforme visto anteriormente. Todos esses modelos de barramento são

⁹ Aqui considerado como sendo um agrupamento de 8 bits.

¹⁰ Lê-se “528 mega bates por segundo”.

¹¹ A divisão por 8 se justifica para obter o resultado em bytes.

disponibilizados na placa-mãe do micro, através de conectores, chamados *slots*. A exceção fica por conta dos barramentos USB, *Firewire* e IrDA que são externos ao micro.

O grande inconveniente apresentado dos barramentos de expansão é a velocidade. Embora a maioria dos periféricos utilizados no micro seja lenta (como a impressora, o mouse, a unidade de disquete, a unidade de CD-ROM e o teclado), existem, a priori, três classes de periféricos bastante prejudicadas pela baixa velocidade: vídeo; discos rígidos e interfaces para rede local.

A princípio, pode parecer que, se o micro tiver esses periféricos integrados à própria placa-mãe, eles trabalharão à mesma velocidade da placa-mãe (isto é, à mesma velocidade do barramento local). Isso, porém, não é verdade.

No projeto da placa-mãe integradas, como em micros de arquitetura fechada, normalmente os periféricos integrados comunicam-se com o processador através de uma extensão do barramento de expansão, chamado barramento **X** (eXtension bus), ou seja, mesmo o periférico estando integrado na placa-mãe ele é tratado como se estivesse conectado a um dos *slots* de expansão.

I/O channel

O surgimento do PC (Figura 6) deu início a uma era de padronizações e uniformizações. Na época do seu lançamento (1981) o mercado era caracterizado por uma grande variedade de computadores pessoais e com eles diversos e incompatíveis barramentos, isso sem contar a incompatibilidade entre periféricos.

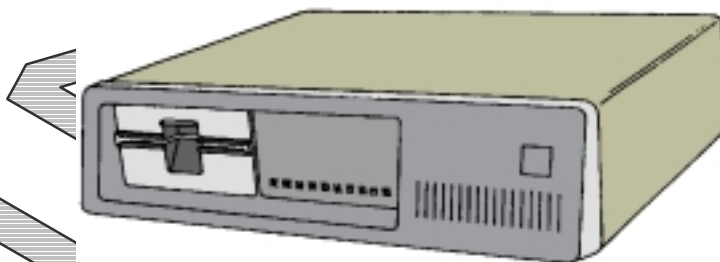


Figura 6

A arquitetura aberta do PC e a utilização de componentes comerciais permitiram uma proliferação de clones, tornando o mercado variado e competitivo e o melhor, relativamente compatível. Tudo isso graças à *big blue* (IBM).

O *slot* de expansão existente no "velho" IBM PC e IBM PC-XT tem a característica de ser um *slot* de 8 bits por utilizar somente 8 bits de dados (esse primeiro PC¹² fundamentava-se no 8088 que, apesar deste ser um microprocessador capaz de manipular internamente dados de 16 bits, comunica-se com o meio externo através de palavras de 8 bits). O barramento do IBM PC tornou-se padrão porque praticamente todos os vendedores de clones de PC o copiaram para permitir que muitas placas existentes de E/S de terceiros pudessem ser utilizadas em seus sistemas.

O barramento possui 62 pinhas sendo ele fisicamente fixado na placa mãe que possui cerca de meia dúzia desses conectores espaçados $\frac{3}{4}$ " um do outro nos quais podem ser inseridos cartões.

¹² Personal Computer.

Cada cartão possui uma borda que encaixa no conector. Esta borda tem 31 faixas (cobertas com ouro) de cada lado que fazem o contato elétrico com o conector (para registro: a IBM chama-o **barramento de I/O channel**). Na *Figura 7* estão representados quatro desses *slots* vistos por cima.

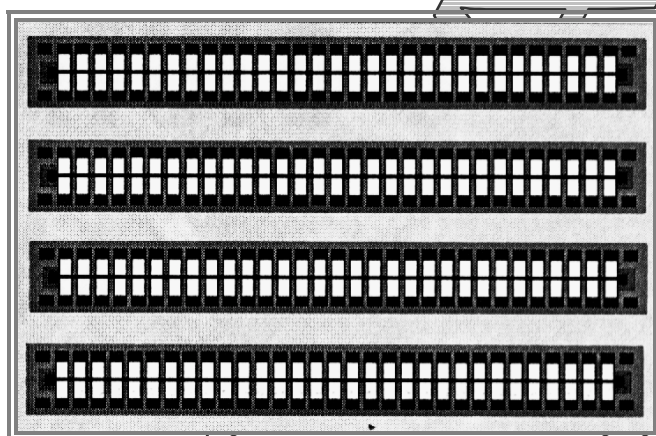


Figura 7

É interessante salientar que na época do primeiro PC e do PC XT, o *slot* era conectado diretamente ao barramento local do micro já que esses computadores utilizavam microprocessadores com baixos valores de frequência para os atuais padrões de frequências de operação.

Principais características:

- barramento de dados de 8 bits
- barramento de endereços de 20 bits
- frequência de operação de 8 MHz¹³.

ISA - Industry Standard Architecture

O lançamento do PC AT (*Advanced Technology*), *Figura 8*, marca uma nova geração nas arquiteturas de barramento. A linha AT estendeu o barramento de dados para 16 bits (o dobro do XT) e o de endereços para 24 bits. A compatibilidade com o antigo XT foi mantida porque os sinais adicionais foram colocados em um *slot* em separado. Com a evolução dos periféricos, começou-se a exigir uma eficiência maior na capacidade de comunicação da CPU com eles (e essa eficiência é uma função do barramento).

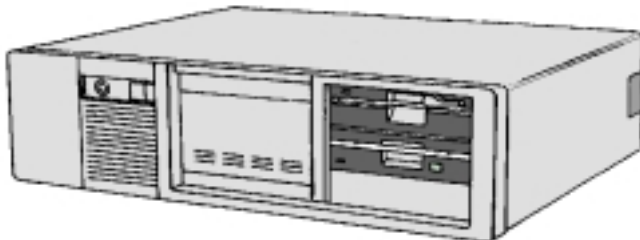


Figura 8

Com a introdução do micro AT da IBM, o barramento e o *slot* ISA aumentaram de tamanho,

¹³ O IBM PC original, assim como muitos clones, é acionado por um sinal de 14,31818 z (frequência correspondente do sinal de crominância utilizado nos televisores em cores da América do Norte e Japão – a IBM pensava que muitos consumidores usariam seus aparelhos de TV como dispositivo de vídeo o que, realmente não aconteceu). Esta frequência era levada para o 8088 padrão, cujo máximo é 5 MHz, sendo, então dividida por 3 para produzir um sinal de 4,77 MHz que na verdade serve como relógio mestre que determina o ciclo do barramento - alguns clones do PC trabalham a 8 MHz, usando uma versão mais rápida da pastilha 8088.

de forma a acompanhar as características do microprocessador 80286, ou seja:

- barramento de dados de 16 bits
- barramento de endereços de 24 bits (16M¹⁴)
- frequência de operação de 8 MHz.

Como o IBM PC AT é baseado no microprocessador 80286, que manipula dados de 16 bits e possui 24 linhas de endereços surge o problema de como manter a compatibilidade de *hardware* entre ele e o XT, baseado no 8088 que externamente manipula dados de 8 bits e possui 20 linhas de endereço? E mais, como aproveitar no AT as interfaces disponíveis no XT? Afinal de contas muitos dos potenciais consumidores teriam hesitado em comprar o AT porque nenhuma das muitas placas disponíveis para o PC não funcionaria na nova máquina.

Para resolver esta situação foi criado um barramento atendendo ao padrão ISA. Neste padrão, o *slot* de expansão (onde, é claro, irão ser encaixadas as interfaces) é totalmente compatível com o *slot* de expansão do XT (8 bits como vimos acima). Mas se o AT possui mais linhas de dados, mais linhas de endereços e mais linhas de controle, como é possível tal compatibilidade?

O *slot* do padrão ISA é dividido em dois, como se fossem dois *slots* independentes; em uma parte é um *slot* com barramento igual ao do XT e em outra, menor, localizam-se as linhas de dados, endereços e controles adicionais. Conseqüentemente, o *slot* ISA tem acesso a 16 bits de dados e até 16 MB de endereços, fora as linhas de interrupção e os canais de DMA a mais. O segundo conector do barramento PC AT contém 36 linhas. É interessante notar que qualquer interface pode ser colocada em qualquer *slot*, seja ela mais antiga (projetada originalmente para XT) ou seja projetada para o *slot* ISA; porém algumas placas mãe possuem, em separado, um *slot* de 8 bits sem a sua “continuação” ISA e alguns leigos acreditam que interfaces de 8 bits só podem ser conectadas em tal *slot* específico. Isso não é verdade! Esse(s) *slots* em separado só existe(m) porque algumas interfaces mais antigas têm uma espécie de rebarba na placa de circuito impresso que fica exatamente na linha dos contatos de borda, tornando impossível seu encaixe no *slot* ISA.

Como visto, o barramento ISA foi especificado a partir da adição de um conector de 36 pinos (2 x 18) adjacente ao de 62 pinos (2 x 31) no antigo projeto PC XT - Figura 9. Com isso, possibilitaram-se recursos para barramento de dados de 8 ou 16 bits, recursos para endereçamento de 16 MB de memória, ciclos de barramento mais rápidos e outros.

A parte do conector ISA de 62 pinos compatível com XT apresenta algumas alterações:

- Em primeiro lugar, o pedido de interrupção IRQ 2¹⁵ foi substituído pela IRQ 9, pois a IRQ 2 passou a ser usada para fazer o cascadeamento dos controladores de interrupção. Porém, nada impede que placas baratas possam usar a IRQ 2, pois o BIOS¹⁶ e os sistemas operacionais automaticamente a redirecionam por software para a IRQ 9.
- Em segundo lugar, o conector de 62 pinos possui sinais que controlam o acesso ao primeiro

¹⁴ 2²⁴ = 16M.

¹⁵ Interrupt Request.

¹⁶ Basic Input Output System.

megabyte da memória RAM, enquanto seus análogos do conector de 36 pinos controlam o acesso aos 15 MB restantes.

- Finalmente, em terceiro lugar, o sinal de relógio passa a ser de 8 MHz, valor este intimamente ligado à velocidade de comunicação entre a CPU e as placas de expansão.

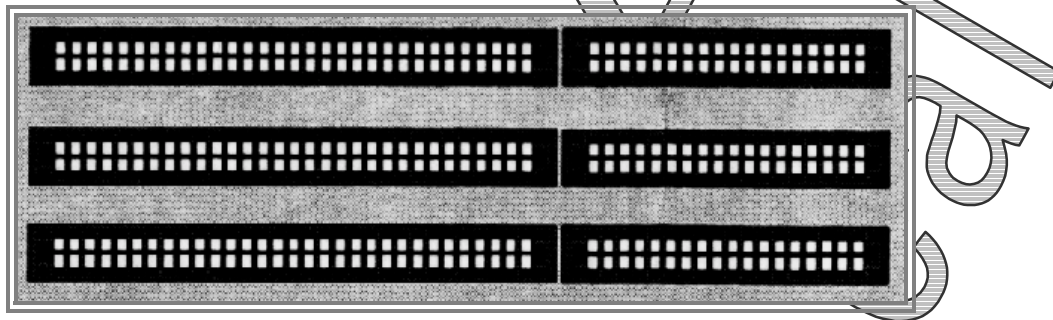


Figura 9

O barramento ISA, então, nada mais é que a padronização do barramento especificado para o PC AT. Originalmente, ele estava sincronizado com a CPU mas, quando a velocidade da CPU excedeu 10 MHz, os periféricos (que operavam a até 8 MHz) não puderam acompanhá-la. Por isso, o barramento ISA trabalha a 8 MHz, usando um barramento de dados com 16 bits, o que dá uma velocidade de comunicação teórica de 16 MB/s - na prática, a velocidade é bem menor.

Imagine-se uma placa de vídeo ISA em um sistema moderno, um Pentium-200 por exemplo. Embora o barramento local trabalhe a 66 MHz, a comunicação do processador com a placa de vídeo será feita somente a 8 MHz: algo em torno de 12% da velocidade com que o processador se comunica com o barramento local!

Dito de outro modo: embora o processador consiga processar os dados rapidamente, o vídeo apresentará os dados na mesma velocidade em que apresentaria em um velho 286. Horrificante, não? Isto é o que se constitui como gargalo para o sistema computacional.

A frequência máxima de 8 MHz é utilizada por um simples motivo técnico: as placas antigas não funcionariam em micros novos caso a frequência de operação do barramento ISA fosse maior. Em outras palavras, se o barramento ISA não fosse limitado a 8 MHz não se poderia utilizar, por exemplo, uma placa de som antiga, supostamente instalada em um antigo 286, no novo Pentium. Como as placas periféricas ISA são para trabalhar a somente 8 MHz, em geral os circuitos utilizados em placas ISA não conseguem trabalhar a uma frequência muito maior do que essa.

Para comunicação do barramento ISA com o barramento local do micro (que são bastante diferentes hoje em dia; basta pensar que o barramento ISA trabalha a 16 bits e o barramento local de um Pentium e superiores operam a 64 bits), há um circuito próprio para o interfaceamento, chamado controlador de barramento ISA. Esse circuito está integrado ao *chipset* da placa-mãe e trata de converter todas as informações de um barramento para outro.

Algumas placas mãe ainda têm o barramento ISA. À primeira vista parece ilógico ao se pensar em termos de desempenho, porém é totalmente aceitável quando se trata de periféricos lentos tais como uma placa de som ou uma placa fax/modem, por exemplo. A maioria dos periféricos é de funcionamento lento. Além disso, o que é extremamente importante, é possível aproveitar, em um

micro moderno, periféricos utilizados em micros mais antigos, como a já mencionada placa de som.

A Figura 10 mostra o aspecto de uma placa ISA típica – notar os entalhes de encaixe para o *slot* ISA (Figura 9).

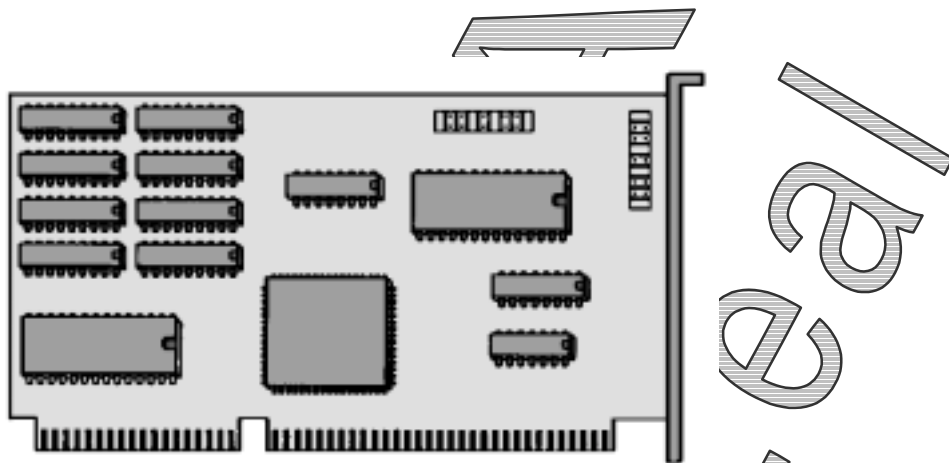


Figura 10

Por causa da lentidão do barramento ISA para periféricos que exigem velocidade (em especial, a placa de vídeo e o disco rígido), outros tipos de barramento deveriam ser criados. Naquela época a IBM não estava preocupada com isso pois o sucessor do AT era o seu PS/2: um microcomputador com arquitetura fechada utilizando um barramento proprietário chamado Arquitetura Microcanal (MCA – **M**icro **C**hannel **A**rchitecture) que contornava tais problemas de desempenho.

Por tratar-se de uma arquitetura fechada os demais fabricantes ficam aliçados do processo. A única padronização de *slots* que poderia ser utilizada por todos os fabricantes sem problemas de compatibilidade era o *slot* ISA, justamente por ser uma arquitetura aberta - exatamente o padrão utilizado por todos os fabricantes. A partir desse momento, todos os micros que tinham arquitetura aberta foram apelidados de "AT" (AT-386, AT-486 etc.), pois utilizavam o *slot* ISA igual ao PC AT original.

Durante muito tempo padrão o ISA continuou sendo a única alternativa para os fabricantes de *hardware*. Basta reparar que todas as placas mãe 386 e muitas das placas mãe 486 não têm nenhum outro tipo de *slot*, a não ser o ISA. A Figura 11 mostra, de forma bem simplificada, uma placa mãe típica para a CPU 386 (situada canto inferior esquerdo da figura) podendo-se ver em relativo destaque a presença de seus oito *slots* padrão ISA.

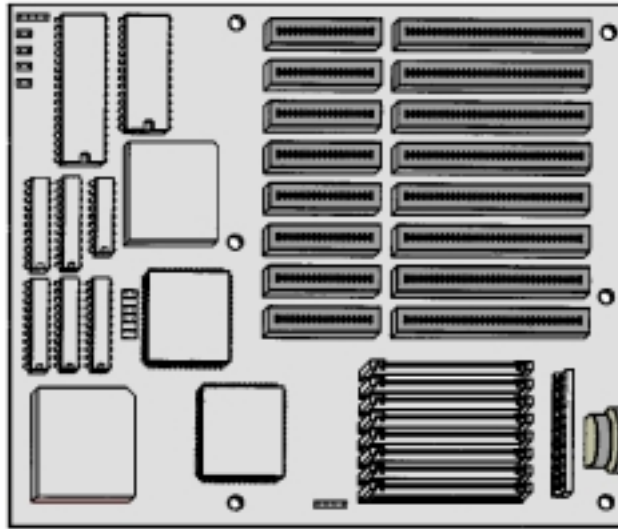


Figura 11

A utilização de algum outro tipo de slot ainda não era possível pois não havia o menor consenso entre os diversos fabricantes de placas mãe e micros existentes no mundo quanto à padronização utilizar. Dessa forma os fabricantes preferiram continuar utilizando o barramento ISA, ainda que, tecnicamente, esse barramento de expansão fosse de inferior performance e ultrapassado.

Para a identificação dos terminais (pinagem) do slot ISA recomendamos a leitura do artigo "Alimentando projetos a partir do PC", Revista nº 335 – dezembro/2000.

MCA – Micro Channel Architecture

O barramento MCA será colocado apenas como referência histórica de modo a compreender-se o que ocorria na época do conhecido e consagrado 386.

Com o surgimento do microprocessador 8386 em 1985, o microprocessador mais revolucionário de todos os tempos, a IBM, como era de se esperar, resolveu projetar e lançar um microcomputador que utilizasse este microprocessador, isto é, era a hora de começar tudo de novo porém agora dando um basta aos fabricantes de clones de PC que tinham pego uma grande parte do mercado. A "big blue" também decidiu não continuar com a sua linha IBM PC, optando por criar uma nova linha de microcomputadores compatíveis com o IBM PC porém com uma arquitetura fechada e proprietária. Esta nova linha chamava-se PS/2¹⁷ e como possuía uma arquitetura fechada (e logicamente patenteada) nenhum fabricante poderia seguir os mesmos moldes.

Diversas interfaces que antes estavam disponíveis através de placas periféricas vinham agora na mesma placa de circuito impresso em que o próprio microcomputador era construído; existindo diversos modelos de PS/2, cada um com conjunto de periféricos bem definido.

A arquitetura com que o PS/2 era construído chama-se Arquitetura Micro Canal ou MCA (*Micro Channel Architecture*). Esse procedimento da IBM fez com que outros fabricantes que quisessem construir microcomputadores compatíveis teriam de formular o seu próprio padrão; como não havia qualquer tipo de padronização e muito menos entendimento entre os diversos fabricantes... Contudo

¹⁷ *Personal System/2* – Sistema Pessoal/2.

com o passar do tempo, e depois de vários atropelos pelo mercado, a linha PS/2 ficou praticamente morta - os computadores PS/2 de outros fabricantes, como o Aptiva usam, na verdade, uma arquitetura semelhante à do AT.

A empresa Compaq resolveu, então, criar um computador fundamentado no microprocessador 80386. Uma das boas soluções que a Compaq apresentou no seu computador foi a colocação das memórias no barramento local, em vez de conectá-las ao barramento de expansão, que trabalhava a 8 MHz. Isso permitiu uma maior velocidade e mais espaço de endereçamento no acesso à memória.

Mas para um processador novo que manipula palavras binárias de 32 bits e acesso direto à memória até 4GB (32 linhas de endereço) de memória, qual padrão deveria ser utilizado? Coube à própria Compaq liderar um grupo de fabricantes (Epson, HP, NEC, Olivetti, Tandy, Zenith, AST entre outros) na tentativa de continuar com a estrutura modular que preservasse sobretudo a compatibilidade, coisa que não acontecia na arquitetura MCA da IBM - foi uma espécie de reação da indústria de computadores pessoais à IBM.

O novo padrão criado por esse grupo de fabricantes chama-se EISA (*Extended Industry Standard Architecture*) ... Isso porém será visto na próxima publicação.

MINIQUILINO