

Capítulo 12

Barramentos da placa de CPU

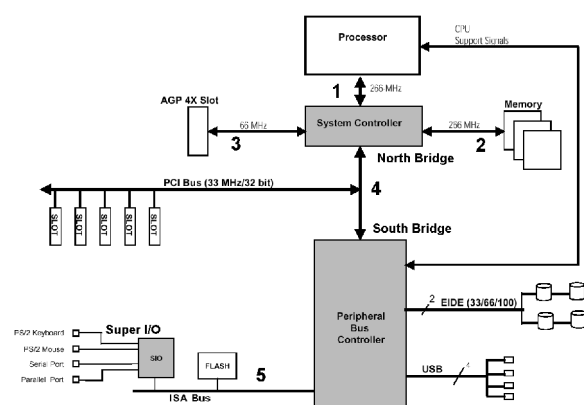
Barramentos internos e externos

Barramentos são conjuntos de sinais digitais através dos quais o processador transmite e recebe dados de circuitos externos. Alguns barramentos são usados para transmissões feitas entre placas, ou dentro de uma mesma placa. Existem vários barramentos nesta categoria:

- Barramento local
- Barramento da memória
- Barramento PCI
- Barramento ISA
- Barramento AGP
- Barramento AMR/CNR

Esses serão os barramentos estudados neste capítulo. Outros barramentos são usados para que o processador e a memória possam receber e transmitir dados para periféricos e dispositivos externos. Entre eles podemos citar os seguintes:

- Barramento SCSI
- Barramento USB
- Barramento Firewire
- Barramento IDE

**FIGURA 12.1**

Principais barramentos de uma placa de CPU.

A figura 1 mostra o diagrama em blocos de uma placa de CPU. Nele podemos ver o processador, as memórias, os dois principais integrantes do chipset (North Bridge e South Bridge – ponte norte e ponte sul), a memória, os slots PCI e AGP, o barramento ISA, as interfaces IDE e USB, o chip Super I/O e suas interfaces. Neste capítulo estudaremos detalhadamente os seguintes barramentos indicados nesta figura:

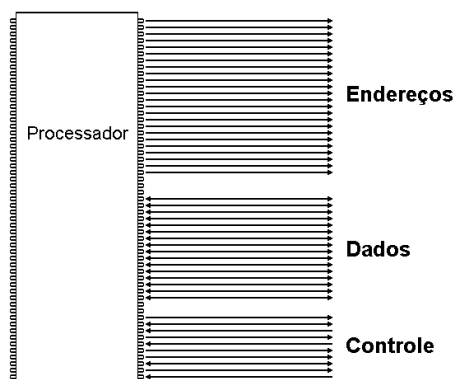
- 1) Barramento do processador, barramento local ou barramento de sistema. É aquele através do qual o processador faz contato direto com o chipset, e a partir daí, com os demais barramentos. Especificamente este barramento é ligado ao chip chamado ponte norte, north bridge ou system controller.
- 2) Barramento da memória. Ligado diretamente na ponte norte, dá acesso aos soquetes nos quais são instalados os módulos de memória.
- 3) Barramento AGP. É usado para a instalação de uma placa de vídeo AGP.
- 4) Barramento PCI. Através deles podemos usar placas de expansão PCI. A maioria das placas de expansão atuais usam este barramento.
- 5) Barramento ISA. Usado na ligação com placas de expansão antigas que seguiam este padrão. Já existem placas de CPU que não apresentam mais o barramento ISA, mas no seu lugar utilizam um barramento interno chamado LPC, usado na comunicação com o BIOS e com o chip Super I/O.

O barramento do processador

Através deste barramento o processador faz a comunicação com o seu exterior. Nele trafegam os dados lidos da memória, escritos na memória,

enviados para interfaces e recebidos de interfaces. Pode ser dividido em três grupos:

- Barramento de dados
- Barramento de endereços
- Barramento de controle

**FIGURA 12.2**

Barramento do processador.

Através do barramento de endereços o processador pode especificar qual a placa ou interface através da qual quer transmitir ou receber dados, e também especificar o endereço de memória no qual deseja ler ou armazenar dados. A maioria dos processadores modernos têm barramento de endereços com 36 bits, podendo assim endereçar até 64 GB de memória física. O barramento de dados tem 64 bits na maioria dos processadores modernos. O barramento de endereços é sempre unidirecional, ou seja, os bits são gerados pelo processador. O barramento de dados é bidirecional, ou seja, os dados são ora transmitidos, ora recebidos pelo processador.

O barramento de controle contém vários sinais que são necessários ao funcionamento do processador, bem como controlar o tráfego do barramento de dados. Alguns dos seus sinais são de saída, outros são de entrada, outros são bidirecionais. Existem sinais para indicação do tipo de operação (leitura ou escrita), sinais de especificação de destino/origem de dados (memória ou E/S), sinais de sincronismo, sinais de interrupção, sinais que permitem a outro dispositivo tomar o controle do barramento, sinais de clock, sinais de programação e diversos outros.

Na maioria dos casos, o barramento do processador é o mais veloz existente em uma placa de CPU, mas isto nem sempre ocorre. Por exemplo, uma

placa de CPU pode ter o processador operando com barramento de 100 MHz e as memórias operando a 133 MHz. Ter a memória mais rápida é vantajoso no caso de placas com vídeo onboard, já que estaria sendo acessada, ora pelo processador, ora pelos circuitos de vídeo. Neste caso o barramento da memória seria o de tráfego mais intenso do computador.

Velocidade do barramento do processador

Os barramentos de todos os processadores modernos operam de forma síncrona, ou seja, os eventos são sincronizados com um sinal de clock. Nos processadores antigos, cada operação era feita em dois, três ou mais períodos de clock. Nos processadores modernos, cada operação pode ser feita em um só período de clock. Infelizmente nem sempre as memórias e outros dispositivos são tão velozes a ponto de acompanhar esta velocidade, e assim são feitas prorrogações chamadas de wait states. Por exemplo, um Pentium III operando com 64 bits (8 bytes) e clock de 100 MHz externos, oferece uma taxa de transferência máxima de 800 MB/s (100 MHz x 8). Na prática este valor não é alcançado, pois nem todas as transferências podem ser feitas em um único ciclo. Por exemplo, todas as memórias modernas podem transferir dados a cada período de clock, porém o primeiro dado demora mais estar pronto, é o que chamamos de latência. Devido às latências das memórias e de outros dispositivos mais lentos, a taxa de transferência máxima nunca é obtida na prática.

A velocidade máxima teórica de um barramento é um ponto bastante importante, por isso vamos discutir o assunto com mais detalhes. Tanto é assim que ao lançar novos processadores (como o caso do Athlon/Duron e do Pentium 4 e Itanium), a Intel e AMD se preocuparam em oferecer taxas de transferência bem elevadas. Para calcular a taxa de transferência, basta multiplicar o clock externo do processador pelo número de bytes do seu barramento de dados. No caso de processadores que operam com DDR (Athlon e Duron) o clock deve ainda ser multiplicado por 2, e para processadores que operam com QDR (Quad Data Rate, como é o caso do Pentium 4 e do Xeon), o clock deve ser multiplicado por 4. A tabela a seguir mostra a taxa de transferência máxima teórica dos principais processadores produzidos nos últimos anos.

Processador	Clock externo	Transferências por ciclo	Taxa de transferência máxima teórica
Pentium / Pentium MMX	66 MHz	1	533 MB/s
K6-2 / K6-III	100 MHz	1	800 MB/s
Pentium II / Celeron	66 MHz	1	533 MB/s

Pentium II 350+	100 MHz	1	800 MB/s
Pentium III, Celeron-800+	100 MHz	1	800 MB/s
Pentium IIIB	133 MHz	1	1066 MB/s
Athlon / Duron	100 MHz	2	1600 MB/s
thlon / Duron	133 MHz	2	2133 MB/s
Pentium 4, Xeon	100 MHz	4	3200 MB/s

As latências das memórias impedem que essas taxas sejam obtidas na prática. Um outro fator pode ainda fazer com que a taxa real seja ainda menor. Por exemplo, se um processador Athlon ou Duron operando com barramento de 100 MHz (200 MHz efetivos usando DDR) for ligado a memórias PC133, o desempenho do seu barramento será menor que o de um processador que use barramento de 133 MHz, e não de 200 MHz.

Exemplo de barramento: Pentium MMX e Socket 7

Sempre encontramos nos manuais dos processadores, disponíveis no sites dos seus fabricantes, especificações detalhadas sobre seus barramentos. Apenas como ilustração, mostraremos a seguir algumas informações sobre o barramento do processador Pentium MMX, compatível com os utilizados por outros processadores para o Socket 7. A figura 3 mostra a numeração dos pinos do processador, bem como os nomes recebidos por cada um desses pinos.

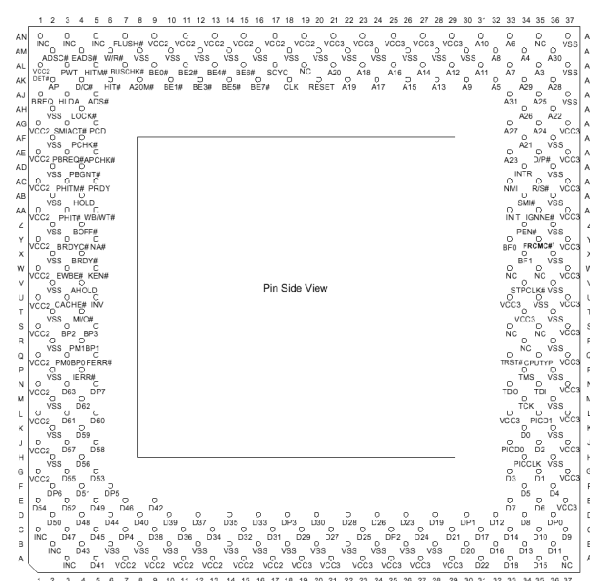


FIGURA 12.3

Pinagem do Pentium MMX.

Observe que vários pinos têm as indicações VSS, VCC2 e VCC3. Os 53 pinos VSS são ligados ao terra da placa de CPU (tensão de 0 volts). Os 28 pinos de VCC3 são para a tensão externa do processador, com 3,3 volts. Os 25 pinos de VCC2 recebem no caso do Pentium MMX, uma alimentação de 2,8 volts, porém em outros processadores esta alimentação tem valores diferentes, e normalmente menores. Esta tensão alimenta o núcleo do processador. O uso de vários pinos de alimentação serve para garantir uma melhor distribuição de correntes dentro do chip e ao longo da placa de CPU.

A figura 4 mostra os barramentos de dados e de endereços. São ao todo 29 pinos de endereços (A3 a A31), com os quais podem ser endereçadas 512 M células de 64 bits, totalizando 4 MB. Como cada célula de 64 bits pode ter seus bytes endereçados individualmente, este processador tem mais 8 sinais de controle (BE0-BE7) que fazem este endereçamento adicional. Por exemplo, o processador pode especificar uma célula de 64 bits qualquer e através dos sinais BE0-BE7, acessar apenas o valor de 8 bits armazenado no 3º byte desta célula, ou o valor de 16 bits armazenado no 5º e no 6º bytes, ou ainda acessar o valor de 32 bits encontrado no 2º, 3º, 4º e 5º byte. Pode ainda usar a célula inteira como um valor de 64 bits, ou então obter esses 64 bits, parte da célula atual e parte da célula seguinte. A figura 4 mostra ainda os 64 pinos que formam o barramento de dados (D0 a D63).

Address									
A3	AL35	A9	AK30	A15	AK26	A21	AF34	A27	AG33
A4	AM34	A10	AN31	A16	AL25	A22	AH36	A28	AK36
A5	AK32	A11	AL31	A17	AK24	A23	AE33	A29	AK34
A6	AN33	A12	AL29	A18	AL23	A24	AG35	A30	AM36
A7	AL33	A13	AK28	A19	AK22	A25	AJ35	A31	AJ33
A8	AM32	A14	AL27	A20	AL21	A26	AH34		

FIGURA 12.4

Barramentos de dados e de endereços do Pentium MMX.

Data									
D0	K34	D13	B34	D26	D24	D39	D10	D52	E03
D1	G35	D14	C33	D27	C21	D40	D08	D53	G05
D2	J35	D15	A35	D28	D22	D41	A05	D54	E01
D3	G33	D16	B32	D29	C19	D42	E00	D55	G03
D4	F36	D17	C31	D30	D20	D43	B04	D56	H04
D5	F34	D18	A33	D31	C17	D44	D06	D57	J03
D6	E35	D19	D26	D32	C15	D45	C05	D58	J05
D7	E33	D20	B30	D33	D16	D46	E07	D59	K04
D8	D34	D21	C29	D34	C13	D47	C03	D60	L05
D9	C37	D22	A31	D35	D14	D48	D04	D61	L03
D10	C35	D23	D26	D36	C11	D49	E05	D62	M04
D11	B36	D24	C27	D37	D12	D50	D02	D63	N03
D12	D32	D25	C23	D38	C09	D51	F04		

O mais complexo barramento do processador é o do controle, onde encontramos uma grande miscelânea de sinais (figura 5). Alguns são de fácil entendimento, bastando ter conhecimentos básicos sobre o funcionamento externo de um processador. Vamos apresentar alguns desses sinais, apenas para fins ilustrativos. Para obter mais detalhes específicos sobre o

funcionamento de cada um desses sinais, é preciso fazer o download do manual do processador de interesse. Esses manuais são encontrados nos sites dos fabricantes, como Intel, AMD e VIA.

Control					
A20M#	AK08	BREQ	AJ01	HIT#	AK06
ADS#	AJ05	BUSCHK#	AL07	HITM#	AL05
ADSC#	AM02	CACHE#	U03	HLDA	AJ03
AHOLD	V04	CPUTYP	Q35	HOLD	AB04
AP	AK32	D/C#	AK04	IERR#	P04
APCHK#	AE05	D/P#	AE35	IGNNE#	AA35
BE0#	AL09	DP0	D36	INIT	AA33
BE1#	AK10	DP1	D30	INTR/LINT0	AD34
BE2#	AL11	DP2	C25	INV	U05
BE3#	AK12	DP3	D18	KEN#	W05
BE4#	AL13	DP4	C07	LOCK#	AH04
BE5#	AK14	DP5	F06	M/IO#	T04
BE6#	AL15	DP6	F02	NA#	Y05
BE7#	AK16	DP7	N05	NMI/LINT1	AC33
BOFF#	Z04	EADS#	AM04	PCD	AG05
BP2	S03	EWBE#	W03	PCHK#	AF04
BP3	S05	FERR#	Q05	PEN#	Z34
BRDY#	X04	FLUSH#	AN07	PM0/BP0	Q03
BRDYC#	Y03	FRCMC# ¹	Y35	PM1/BP1	R04

FIGURA 12.5

Sinais de controle do Pentium MMX.

APIC	Clock Control	Dual Processor Private Interface	
PICCLK	H34 (2)	CLK	AK18 (2)
PICD0	J33	[BF0]	Y33
[DPEN#]		[BF1]	X34
PICD1	L35	STPCLK#	V34
[APICEN]			

Aqui vão portanto as descrições de alguns desses sinais:

Sinal	Descrição
A20M	Usado para dar acesso à chamada memória HMA, que consiste nos primeiros 64 kB localizados após o endereço 1M. É usado quando o processador opera no Modo Real, ou seja, o mesmo utilizado pelo MS-DOS.
ADS	Address Strobe. Indica que o processador está iniciando um novo ciclo. Os ciclos podem ser de leitura da memória, escrita na memória, leitura de I/O, escrita em I/O, etc.
AP e EADS	Usados para geração e checagem de paridade no barramento de endereços.
BF0 e BF1	Indica o multiplicador usado para definir o clock interno, em função do clock extenro (2x, 2,5x, 3x e 3,5x, no caso do Pentium MMX). Processadores como o K6-2 possuem ainda um pino adicional BF2 para a definição de multiplicadores maiores. Esses pinos são ligados aos jumpers da placa de CPU que fazem a programação dos multiplicadores.
BRDY	Indica ao processador se a atual transferência em modo burst já está finalizada ou se devem ser usados wait states.
CLK	Através desta entrada o processador recebe o seu clock externo.
CPUTYP	Usado para distinguir entre processador primário e secundário, em placas equipadas com mais de um processador.
D/C	Data/Code. Através deste sinal o processador indica se está fazendo um acesso a dados ou a instruções de programas.
DP0-DP7	Usados para geração e controle de paridade no barramento de dados. Cada um desses 8 sinais representa a paridade de 8 bits do barramento de dados.

FERR	Floating Point Erros. Usado para indicar a ocorrência de um erro de ponto flutuante, resultante de uma operação matemática inválida feita pela unidade de ponto flutuante, como uma divisão por zero ou raiz quadrada de um número negativo.
HOLD	Por esta entrada o processador recebe uma requisição para que entre em alta impedância (tristate), possibilitando que outro chip tome o controle do barramento. É usado em transferências de DMA e Bus Mastering.
HLDA	HOLD Acknowledge. O processador informa que está entrando em modo de tristate. A partir daí o barramento pode ser usado por outros chips.
M/IO	O processador indica se está acessando uma posição de memória ou uma interface de I/O (entrada e saída).
R/W	O processador indica se está realizando uma operação de leitura ou escrita.
RESET	Ao receber um comando nesta entrada, o processador realiza seu processo de RESET interno. Este sinal deve ser ativado mediante o botão RESET do gabinete, e também quando o computador é ligado (Power-on RESET).

Esses são apenas alguns dos sinais de controle deste processador, que são encontrados também nos processadores mais modernos. Nossa intenção aqui não foi descrever detalhadamente todos os sinais, mas apenas dar ao leitor uma idéia do tipo de função realizada por esses sinais.

O barramento das memórias

Nas placas de CPU antigas, as memórias eram ligadas diretamente ao barramento do processador, através de chips chamados buffers bidirecionais. Esses chips tinham como único objetivo amplificar a corrente vinda do processador, permitindo que o barramento de dados fosse ligado a um número grande de chips de memória. Portanto a velocidade do barramento do processador era igual à velocidade do barramento das memórias.

Velocidade do barramento das memórias

Atualmente as memórias são ligadas ao processador através do chipset. A função do chipset nesta conexão não é de apenas aplicar corrente. A maioria dos chipsets possui registradores que permitem que a memória opere de forma assíncrona ao processador, ou seja, com um clock diferente. Alguns chipsets podem ter o processador operando a 100 MHz e as memórias a 66, ou 133 MHz. Outros podem ter o processador operando a 200 MHz e as memórias a 133. Existem vários outros exemplos de clocks diferentes. Nesses casos dizemos que a memória está operando de forma assíncrona ao processador.

Podemos encontrar barramentos de memória operando com diversas velocidades:

Tipo de memória	Clock	Transferências por ciclo	Taxa de transferência máxima teórica
-----------------	-------	-----------------------------	---

FPM, EDO	66 MHz	1/3	176 MB/s (*)
SDRAM PC66	66 MHz	1	533 MB/s
SDRAM PC100	100 MHz	1	800 MB/s
SDRAM PC133	133 MHz	1	1067 MB/s
DDR200	100 MHz	2	1600 MB/s
DDR266	133 MHz	2	2133 MB/s
DDR300	150 MHz	2	2400 MB/s
DDR333	166 MHz	2	2666 MB/s
DDR400	200 MHz	2	3200 MB/s
RDRAM	100 MHz	4	3200 MB/s (**)

(*) Memórias FPM e EDO gastam de 2 a 4 ciclos em Page Mode para fazer cada transferência, por isso consideramos uma média de 3 ciclos para cada transferência, ou 1/3 de transferência a cada ciclo.

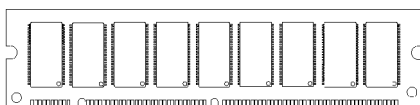
(**) Um módulo RDRAM opera com 1600 MB/s, porém são usados aos pares, resultando em 3200 MB/s. Note ainda que a DDR SDRAM mais veloz indicada na tabela é a DDR400, porém na época em que a RDRAM oferecia 3200 MB/s, a DDR mais veloz era a DDR266. Juntamente com a chegada de chips DDR mais velozes, chegarão também ao mercado chips RDRAM também com maior velocidade.

As taxas de transferência mostradas na tabela acima são meros limites teóricos, e nunca são obtidos na prática. São taxas momentâneas que vigoram apenas quando a transferência se dá em modo burst. Essas taxas não são sustentadas por períodos significativos, já que a cada 3 transferências em que usam um só ciclo, é exigida uma transferência inicial que dura 2 ou 3 ciclos (latência 2 ou 3), resultando em temporizações como 2-1-1-1 ou 3-1-1-1, o que resulta em 0,8 e 0,75 transferências por ciclo, em média. Mais tempo é perdido antes de cada transferência, ao serem usados os comandos de leitura e gravação, onde mostramos como são as formas de onda dos acessos aos vários tipos de memória). Finalmente um outro fator contribui para reduzir ainda mais o desempenho, que é a atuação da cache. A maioria dos acessos à memória passam pelas caches do processador, mas certos ciclos podem ser feitos no modo uncached. O processador estaria neste caso fazendo leituras e escritas diretamente na memória, e sem usar o modo burst (transferências da cache são feitas em grupos de 4 acessos consecutivos).

Os sinais de um barramento de memória

Existem algumas diferenças, principalmente no número de bits do barramento de dados e de endereços dos chips de memória. Já o barramento de memória não admite variações. Existe um padrão que deve ser seguido por todos os fabricantes. Por exemplo, os barramentos dos módulos SDRAM DIMM/168 devem ter sempre 64 ou 72 bits. Diferenças podem existir nos barramentos de endereços, já que módulos de maior

capacidade exigem mais bits para seu endereçamento. Entretanto, no mesmo soquete onde instalamos um módulo de 512 MB, também precisam ser suportados módulos de menores capacidades, o que resulta em variações no número de bits de endereços. Essas diferenças não são uma despadronização. O que ocorre é que o barramento deve ser compatível com módulos de diferentes capacidades. Outros sinais são padronizados em módulos de qualquer capacidade. Por exemplo, os sinais RAS e CAS dos módulos SDRAM DIMM/168 devem sempre ocupar os pinos 115 e 111, respectivamente. A figura 6 mostra a utilização de todos os pinos do barramento utilizado pelas memórias SDRAM DIMM/168.



PIN	SYMBOL	PIN	SYMBOL	PIN	SYMBOL	PIN	SYMBOL
1	Vss	22	NC	43	Vss	64	Vss
2	DQ0	23	Vss	44	NC	65	DQ21
3	DQ1	24	NC	45	SZ#	66	DQ22
4	DQ2	25	NC	46	DQMB2	67	DQ23
5	DQ3	26	Vdd	47	DQMB3	68	Vss
6	Vdd	27	WE#	48	NC	69	DQ24
7	DQ4	28	DQMB0	49	Vdd	70	DQ25
8	DQ5	29	DQMB1	50	NC	71	DQ26
9	DQ6	30	SO#	51	NC	72	DQ27
10	DQ7	31	NC	52	NC	73	Vdd
11	DQ8	32	Vss	53	NC	74	DQ28
12	Vss	33	A0	54	Vss	75	DQ29
13	DQ9	34	A2	55	DQ16	76	DQ30
14	DQ10	35	A4	56	DQ17	77	DQ31
15	DQ11	36	A6	57	DQ18	78	Vss
16	DQ12	37	A8	58	DQ19	79	CK2
17	DQ13	38	A10	59	Vdd	80	NC
18	Vdd	39	BA0	60	DQ20	81	WP
19	DQ14	40	Vdd	61	NC	82	SDA
20	DQ15	41	Vdd	62	NC	83	SCL
21	NC	42	CK0	63	CKE1	84	Vdd

PIN	SYMBOL	PIN	SYMBOL	PIN	SYMBOL	PIN	SYMBOL
85	Vss	106	NC	127	Vss	148	Vss
86	DQ32	107	Vss	128	CKE0	149	DQ53
87	DQ33	108	NC	129	SZ#	150	DQ54
88	DQ34	109	NC	130	DQMB6	151	DQ55
89	DQ35	110	Vdd	131	DQMB7	152	Vss
90	Vdd	111	CAS#	132	NC	153	DQ56
91	DQ36	112	DQMB4	133	Vdd	154	DQ57
92	DQ37	113	DQMB5	134	NC	155	DQ58
93	DQ38	114	S1#	135	NC	156	DQ59
94	DQ39	115	RAS#	136	NC	157	Vdd
95	DQ40	116	Vss	137	NC	158	DQ60
96	Vss	117	A1	138	Vss	159	DQ61
97	DQ41	118	A3	139	DQ48	160	DQ62
98	DQ42	119	A5	140	DQ49	161	DQ63
99	DQ43	120	A7	141	DQ50	162	Vss
100	DQ44	121	A9	142	DQ51	163	CK3
101	DQ45	122	BA0	143	Vdd	164	NC
102	Vdd	123	A11	144	DQ52	165	SA0
103	DQ46	124	Vdd	145	NC	166	SA1
104	DQ47	125	CK1	146	NC	167	SA2
105	NC	126	A12	147	NC	168	Vdd

*** 75%

FIGURA
12.6

Pinagem de um barramento de memória SDRAM DIMM/168.

A seguir apresentamos uma breve descrição dos mais importantes sinais de um barramento de SDRAM DIMM/168.

Sinal	Descrição
RAS	Row Address Strobe. Ligado diretamente aos pinos de RAS dos chips de memória encontrados nos módulos DIMM/168.
CAS	Column Address Strobe. Ligado diretamente aos pinos de CAS dos chips de memória.
WE	Write Enable. Também ligado nos pinos de mesmo nome nas memórias, é usado para indicar se a atual operação é de leitura (bit 1) ou escrita (bit 0).
CK0-CK3	São sinais idênticos e transmitem aos chips do módulo, os sinais de clock necessários aos seu funcionamento. Em módulos PC133, este clock é de 133 MHz.
DQ0-DQ63	São os 64 bits do barramento de dados do módulo.
CB0-CB7	São 8 bits adicionais usados para checagem de erros (Paridade ou ECC)
A0-A12	Barramento de endereços, especifica a célula de memória a ser acessada.

BA0, BA1	Selecionamento de banco. Cada chip de SRAM possui 4 bancos internos. Esses dois bits são usados para o selecionamento do banco desejado. O mesmo selecionamento é enviado a todos os chips do módulo.
S0, S1, S2, S3	Selecionamento de banco dentro do módulo. Existem módulos de 1 e de dois bancos. Normalmente os de 1 banco usam chips de um só lado, enquanto os de 2 bancos possuem chips em ambos os lados. O primeiro banco é sempre ativado pelos sinais S0 e S2, enquanto o segundo banco, quando existe, é ativado pelos sinais S1 e S3.
DQMB0-DQMB7	Cada um desses 8 bits de controle ativa o bits consecutivos do barramento de dados do módulo. DQMB0 habilita os bits DQ0-DQ7; DQMB1 habilita os bits DQ8-DQ15, e assim por diante, até DQMB7 que habilita os bits DQ56-DQ63. Através desses controles, o chipset, mediante comando apropriado do processador, pode acessar grupos de 8, 16, 32 ou 64 bits na célula endereçada. Isso é útil, por exemplo, quando é preciso gravar um dado em um único byte, ou em outras porções menores que os 64 bits do barramento de dados do módulo.
VCC	Alimentação de +3,3 volts.
VSS	Ligados ao terra

Células de memória em um módulo DIMM têm 64 bits, ou seja, 8 bytes. É interessante observar como essas células são endereçadas. São usados 3 grupos de sinais:

Sinais	Funcionamento
S0, S1, S2 e S3	Esses sinais fazem a distinção entre módulos de face simples e de face dupla. Nos módulos de face simples, devem ser ativados simultaneamente os sinais S0 e S2. Nos módulos de face dupla, os sinais S0 e S2 ativam os chips de uma face, e os sinais S1 e S3 ativam os chips da outra face. Mais especificamente: S0 ativa os bits DQ0-DQ31 da primeira face S2 ativa os bits DQ32-DQ63 da primeira face S1 ativa os bits DQ0-DQ31 da segunda face S3 ativa os bits DQ32-DQ63 da segunda face Portanto esses sinais podem representar até 2 combinações: S0 e S2 ativos S1 e S3 ativos
BA0, BA1	Esses sinais são enviados a todos os chips do módulo. Servem para selecionar um dos 4 bancos internos que um chip de SDRAM pode ter. Portanto esses sinais podem representar 4 combinações: Banco 0: BA1=0 e BA0=0 Banco 1: BA1=0 e BA0=1 Banco 2: BA1=1 e BA0=0 Banco 3: BA1=1 e BA0=1

A0-A11 ou A0-A12	Dependendo da capacidade do módulo, podem existir 12 bits (A0-A11) ou 13 bits (A0-A12) de endereçamento. Este grupo de bits é usado duas vezes para indicar a linha e a coluna desejada, juntamente com os sinais RAS e CAS. Podem portanto fornecer nas duas etapas, 24 ou 26 bits de endereços. Desta forma é possível endereçar 16M (2^{24}) ou 64M (2^{26}) células.
------------------	--

Sendo assim, a capacidade de um módulo de memória depende obviamente do número e da capacidade dos chips, mas também deve estar de acordo com o número de faces e do número de bits de endereço. Por exemplo, para módulos com 12 bits de endereços e face dupla, a capacidade máxima permitida é de:

$$16\text{M} \times 2 \text{ bancos por módulo} \times 4 \text{ bancos internos} \times 8 \text{ bytes por célula} = 1024 \text{ MB}$$

A tabela abaixo mostra as capacidades máximas dos módulos em função do número de bits de endereços e do número de faces:

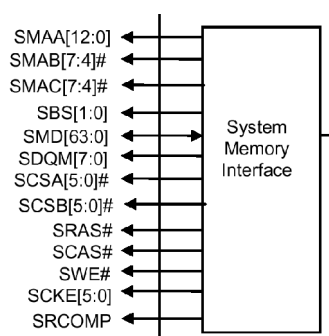
Endereços	Face	Capacidade máxima
A0-A11	Simples	512 MB
A0-A11	Dupla	1024 MB
A0-A12	Simples	2048 MB
A0-A12	Dupla	4096 MB

É claro que a utilização depende também da capacidade de cada chip. Quando não existem chips com capacidade suficiente para criar um módulo de alta capacidade e face simples, são usadas ambas as faces do módulo, com os chips disponíveis.

Praticamente todos os sinais do barramento de memória passam por todos os soquetes. Por exemplo, o mesmo barramento de dados DQ0-DQ63 passa por todos os soquetes e vai até o chipset. O mesmo ocorre com o barramento de endereços e sinais de RAS e CAS. Alguns sinais entretanto são distribuídos individualmente para cada soquete. É o caso dos sinais de clock, que por serem de frequência muito elevada, são fornecidos em “cópias individuais” para cada soquete. Isso evita que os sinais de clock sejam deteriorados, coisa que normalmente ocorre em pequena escala quando um sinal digital é ligado em muitos chips. A maioria dos chipsets possui vários sinais de clock, todos iguais e sincronizados, mas para serem enviados separadamente para cada soquete de memória.

Sempre que vários circuitos semelhantes são ligados em paralelo é preciso que algum sinal digital faça o seletor individual do circuito desejado,

caso contrário todos os circuitos tentariam fornecer dados ao mesmo tempo. Digamos por exemplo que um PC tem 256 MB de memória, formados por dois módulos DIMM/168 com 128 MB cada. O chipset precisa saber que endereços de 0 até 128 M pertencem ao primeiro módulo, e que endereços de 128 M a 256 M pertencem ao segundo módulo (note que a rigor o primeiro módulo termina no endereço 128M-1, enquanto o segundo começa exatamente no endereço 128M). Os sinais S0, S1, S2 e S3 de cada soquete de memória são usados para este propósito. O chipset possui vários bits de endereçamento através dos quais podem ativar os sinais S0, S1, S2 e S3 de cada soquete de memória, de forma individual. Através do endereço desejado pelo processador, o chipset sabe exatamente qual módulo de memória o contém, e ativa os sinais de selecionamento de banco adequados.



*** 35% ***

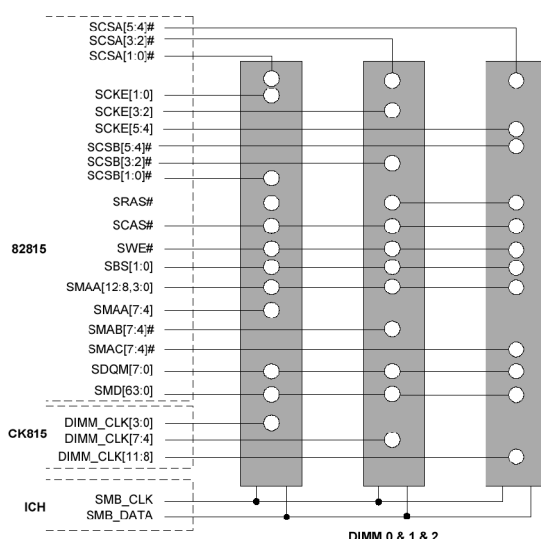
FIGURA 12.7

Sinais do controlador de memória do chipset Intel i815.

A figura 7 mostra os sinais do controlador de memória contido no chipset Intel i815. Esses sinais têm correspondência direta com o barramento dos soquetes DIMM/168 das memórias SDRAM. Por exemplo, os sinais SMAA [12:0] (o mesmo que SMAA0, SMAA1, ..., SMAA12) são o barramento de endereços das memórias. Os sinais SDQM[7:0] são ligados diretamente aos pinos DQMB dos soquetes de memória. Os sinais SRAS, SCAS e SWE são ligados diretamente aos pinos RAS, CAS e WE dos soquetes. Já os sinais SCSA[5:0] e SCSB[5:0] são usados para ligações individuais nos pinos S0, S1, S2 e S3 de cada soquete de memória. Este chipset permite usar até 3 módulos DIMM/168, que devem ter seus sinais de selecionamento ligados ao chipset da seguinte forma:

DIMM0	DIMM1	DIMM2
S0 = SCSA0	S0 = SCSA2	S0 = SCSA4
S2 = SCSB0	S2 = SCSB2	S2 = SCSB4
S1 = SCSA1	S1 = SCSA3	S1 = SCSA5
S3 = SCSB1	S3 = SCSB3	S3 = SCSB5

O arranjo é mostrado na figura 8. Note que cada soquete DIMM está ligado a sinais SCSA e SCSB apropriados, responsáveis pelo seu selecionamento. A maioria dos demais sinais são ligados em paralelo, aos três soquetes.

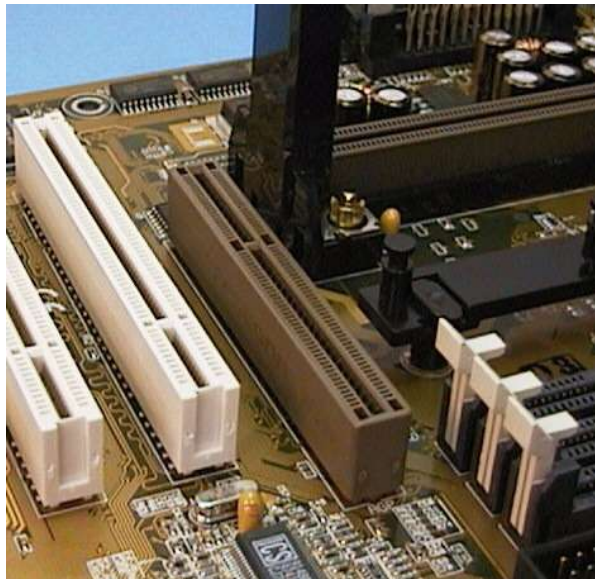
**FIGURA 12.8**

Ligação do barramento de memória ao chipset.

Uma descrição completa de todos os barramentos de memória seria inoportuna neste momento. Procuramos portanto explicar o funcionamento do barramento de memória, tomando como exemplo a SDRAM. O barramento da DDR SDRAM tem sinais bastante parecidos, exceto pela taxa de dados dupla.

Barramento AGP

Este barramento foi lançado em 1997 pela Intel, especificamente para acelerar o desempenho de placas de vídeo em PCs equipados com o Pentium II e processadores mais modernos. Trata-se do Accelerated Graphics Port. É formado por um único slot, como o mostrado na figura 9. Observe que este slot é muito parecido com os utilizados no barramento PCI, mas existem diferenças sutis do ponto de vista mecânico. Fica um pouco mais deslocado para a parte frontal do computador, além de possuir uma separação interna diferente da existente no slot PCI. Desta forma, é impossível encaixar neste slot, uma placa que não seja AGP.

**FIGURA 12.9**

Slot AGP.

O AGP é um slot solitário, usado exclusivamente para placas de vídeo projetadas no padrão AGP. Muitos modelos de placas de vídeo são produzidas nas versões PCI e AGP (ex: Voodoo 3 3000 AGP e Voodoo 3 3000 PCI). A principal vantagem do AGP é a sua taxa de transferência, bem maior que a verificada no barramento PCI.

A figura 10 mostra uma placa de vídeo AGP. Observe a posição do seu conector, mais afastado da parte traseira da placa, o que não ocorre no padrão PCI.

**FIGURA 12.10**

Placa de vídeo AGP.

Placas de CPU com slot AGP começaram a se tornar comuns a partir de 1998. As primeiras placas de CPU a apresentar slot AGP foram as que usavam o chipset Intel i440LX, para Pentium II, e depois as que usavam o i440BX. Outros fabricantes de chipsets passaram a desenvolver produtos que também davam suporte ao barramento AGP. Placas de CPU para a plataforma Super 7 (K6, K6-2, etc.) também passaram a apresentar slot AGP. Atualmente todas as placas de CPU de alto desempenho apresentam um slot AGP. Por outro lado, muitas placas de CPU para PCs de baixo custo, tipicamente as que possuem vídeo onboard, não possuem slot AGP, com raras exceções.

AGP e vídeo onboard

Foram produzidas várias placas de CPU com vídeo onboard, sem slot AGP, entretanto com os circuitos de vídeo internamente ligados ao barramento AGP. Em outras palavras, essas placas possuem barramento AGP mas não possuem slot AGP. Elas têm os circuitos de vídeo embutidos, ligadas ao barramento AGP, porém não permitem que o usuário desative o vídeo onboard e instale uma placa de vídeo AGP. Como na maioria dos casos o vídeo onboard é de baixo desempenho (mesmo sendo AGP), o usuário que quiser melhorar o desempenho do vídeo precisa se contentar com uma placa de vídeo PCI.

Existem entretanto placas de CPU com vídeo onboard mas que possuem um slot AGP disponível para expansões. Placas de CPU com esta característica podem ser usadas para montar computadores simples, mas que podem posteriormente ser convertidos em modelos mais avançados, através da instalação de placas de expansão apropriadas.

As várias voltagens do AGP

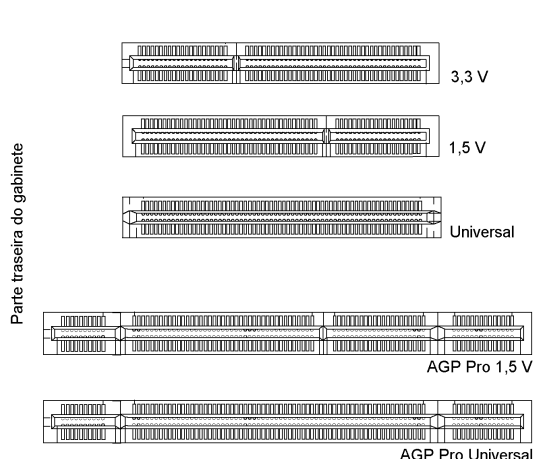
Desde que o barramento AGP foi criado, várias versões foram lançadas no que diz respeito à voltagem e velocidade. As primeiras versões operavam com 3,3 volts. As placas de CPU tinham slots AGP operando com 3,3 volts (a exemplo das memórias, chipsets e o barramento externo dos processadores). As placas de vídeo AGP também operavam com os mesmos 3,3 volts, de forma compatível com a placa de CPU. Inicialmente foi lançado o AGP de velocidade simples (AGP 1x), depois o AGP 2x e o AGP 4x, duas a 4 vezes mais velozes, respectivamente.

Para possibilitar a operação em modo 4x, os níveis de voltagem foram alterados para 1,5 volts. Surgiram então os slots AGP para 1,5 volts, capazes de operar exclusivamente com este nível de voltagem, e os slots AGP

universais, capazes de operar tanto com 1,5 como com 3,3 volts. Da mesma forma existem placas AGP de 3,3 volts, placas AGP de 1,5 volts e placas AGP universais.

A nova versão 3.0 da especificação AGP, que suporta o modo de transferência em 8x, opera com tensão de 0,7 volts, entretanto é utilizado o mesmo tipo de soquete para placas de 1,5 volts. Para manter compatibilidade total, tanto as placas de CPU quanto as placas de vídeo AGP 3.0 são capazes de operar tanto com 0,7 volts quanto com 1,5 volts. Ambas as placas são identificadas por novos sinais MB_DET e GC_DET, através dos quais as voltagens corretas são selecionadas.

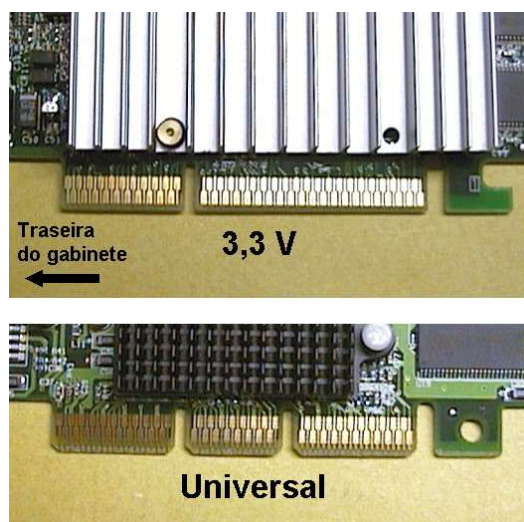
Note que essas tensões de 3,3 volts, 1,5 volts e 0,7 volts não se referem necessariamente ao funcionamento dos chips da placa. Elas se aplicam obrigatoriamente na comunicação entre a placa de vídeo e a placa de CPU, ao longo do slot. Uma placa de vídeo pode ter seus chips operando, por exemplo, com 2,5 volts mas usar tensões de 1,5 volts na comunicação com a placa de CPU. Portanto quando dizemos “placa AGP de 3,3 volts”, ou “placa AGP de 1,5 volts” ou “placa AGP de 0,7 volts”, estamos nos referindo apenas à voltagem usada pelos sinais digitais que trafegam ao longo do slot.

**FIGURA 12.11**

Os vários tipos de slots AGP.

A figura 11 mostra as diversas versões de slots AGP. O slot de 3,3 volts possui um chanfro localizado mais próximo da parte traseira da placa de CPU. O slot AGP de 1,5 volts tem o chanfro na posição inversa. Placas de vídeo AGP possuem conectores com chanfros correspondentes que se encaixam nos chanfros dos slots. Isto impede, por exemplo, que uma placa de 1,5 volts seja encaixada em um slot de 3,3 volts, e vice-versa.

Podemos ainda encontrar slots AGP universais e placas AGP universais. Um slot AGP universal não possui chanfro, e está preparado para operar tanto com 3,3 como com 1,5 volts. A placa instalada é reconhecida e o slot passa a operar com a voltagem apropriada. Da mesma forma encontramos placas AGP universais, com dois chanfros. Elas podem ser encaixadas tanto nos slots de 1,5 como nos de 3,3 volts.

**FIGURA 12.12**

Placas AGP com diferentes posições de chanfros.

Outro ponto importante é a velocidade de operação. As velocidades suportadas são 1x, 2x, 4x e 8x. Quando uma placa AGP é encaixada em um slot AGP de voltagem compatível (note que é impossível fazer o encaixe quando as voltagens não são compatíveis), prevalecerá a máxima velocidade que seja suportada simultaneamente pela placa e pelo slot. As primeiras placas de CPU com barramento AGP operavam com 3,3 volts e suportavam apenas o modo AGP 1x. Depois surgiram placas de CPU com chipsets capazes de operar em AGP 2x, também com 3,3 volts. Os slots AGP universais e os de 1,5 volts são encontrados nas placas capazes de operar em 4x. O modo 4x exige a tensão de 1,5 volts, o mesmo ocorrendo com o modo 8x.

Uma placa AGP 2x de 3,3 volts não pode ser conectada em um slot AGP de 1,5 volts, mas poderá ser encaixada em um slot AGP universal. Esses slots suportam o modo 4x, mas quando a placa de vídeo é 2x, a taxa de transferência será limitada pela placa de vídeo, apesar da placa de CPU poder chegar até 4x.

Versões do AGP

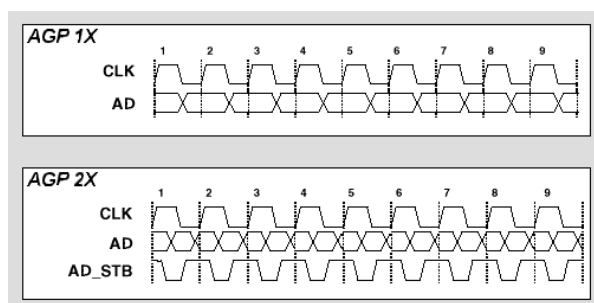
O barramento AGP versão 1.0 foi o primeiro a ser utilizado em placas de CPU e placas de vídeo. Esta versão oferecia os modos 1x e 2x, porém as primeiras implementações operavam apenas em 1x. A próxima especificação foi a AGP 2.0, que estendeu a velocidade para 4x, e finalmente a 3.0 que oferece transferências em até 8x. Além do aumento de velocidade, novas opções de voltagem foram introduzidas, bem como algumas outras modificações no funcionamento. Cada versão nova tem compatibilidade com as versões anteriores, desde que seja respeitado o tipo de conector. Por exemplo, uma placa de CPU compatível com AGP 3.0 e use slot de 1,5 volts, aceitará operar nos modos 4x e 8x. Placas de CPU AGP 3.0 universais suportam também operações em modos 1x e 2x.

AGP 1x, 2x e 4x

O barramento AGP é bastante semelhante ao PCI, mas com algumas modificações voltadas para placas de vídeo. Opera com 32 bits e 66 MHz. Na sua versão inicial (AGP 1x), cada clock realiza uma transferência de 32 bits (4 bytes). Como são 66 MHz (na verdade são 66,66 MHz), temos 66 milhões de transferências por segundo. Sendo as transferências de 4 bytes, o número total de bytes por segundo que podem passar pelo barramento AGP 1x é:

$$66,66 \text{ MHz} \times 4 \text{ bytes} = 266 \text{ MB/s}$$

Esta é uma taxa de transferência fantástica. Com ela é possível preencher todo o conteúdo da memória de vídeo cerca de 90 vezes por segundo (90 Hz), supondo uma resolução gráfica de 1024x768x32 bits. Isto é muito mais que os 30 Hz necessários para ter sensação visual de continuidade de movimentos. Portanto 90 Hz pode parecer um exagero, mas não é. O tráfego de dados no barramento AGP não é simplesmente a transferência de “frames” para a memória de vídeo. É preciso fazer continuamente a leitura de texturas que ficam na memória RAM da placa de CPU, para que sejam automaticamente e rapidamente aplicadas sobre os polígonos que formam as imagens tridimensionais. O tráfego de dados pelo barramento AGP tende a ser ainda mais elevado quando são usadas resoluções mais elevadas, quando são geradas imagens complexas e quando a resolução das texturas é muito elevada. Por isso existem versões novas do barramento AGP, capazes de operar com taxas ainda mais elevadas.

**FIGURA 12.13**

Transferências de dados no barramento AGP, modos 1x e 2x.

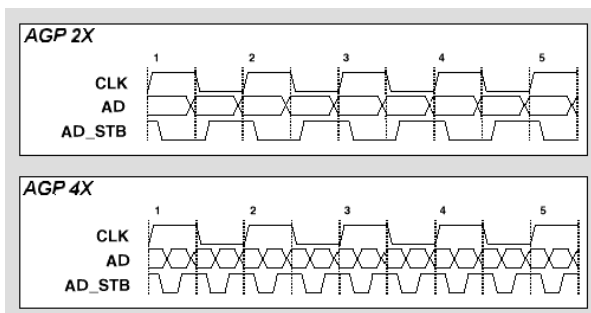
Desde a criação do barramento AGP, já era previsto o aumento da sua taxa de transferência, utilizando os modos 2x e 4x, e mais recentemente, 8x. O modo 2x também opera com 32 bits e 66 MHz, porém em cada período de clock, são feitas duas transferências, ao invés de apenas uma. A figura 13 compara as transferências de dados nos barramentos AGP 1x e 2x. Note que em ambos os casos, o sinal de clock (CLK) é o mesmo, mas no modo 2x é usado o sinal AD_STB para indicar a presença de dados válidos no barramento. Nos instantes em que o sinal AD_STB varia de 1 para 0, ou de 0 para 1, o barramento está pronto para fazer uma transferência. Como em cada ciclo de clock (indicados na figura pelos números 1, 2, etc.) existem duas transições de AD_STB, temos duas transferências a cada ciclo. Portanto a taxa de transferência no modo 2x é dada por:

$$66,66 \text{ MHz} \times 2 \times 4 \text{ bytes} = 533 \text{ MB/s}$$

O modo 4x utiliza um processo similar. A principal diferença é que o sinal AD_STB apresenta 4 transições a cada período de clock, portanto são feitas 4 transferências em cada ciclo. A taxa de transferência no modo 4x é então:

$$66,66 \text{ MHz} \times 4 \times 4 \text{ bytes} = 1066 \text{ MB/s}$$

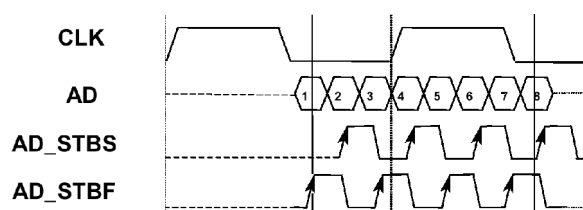
As primeiras placas de CPU com slot AGP possuíam suporte apenas para o modo 1x, bem como ocorria com as primeiras placas de vídeo AGP. Em 1999 já era comum encontrar placas de CPU e placas de vídeo, ambas capazes de operar no modo AGP 2x. Em 2000, praticamente todas as placas de CPU, e boa parte das placas de vídeo modernas operavam em AGP 4x.

**FIGURA 12.14**

Transferências AGP nos modos 2x e 4x.

AGP 8x

O modo AGP 8x faz parte da especificação AGP 3.0. Sua principal característica é o uso de taxas de transferência 8 vezes maiores que as oferecidas pelo AGP 1x. Em modo 8x, a taxa de transferência teórica máxima é de 2133 MB/s. Esta taxa é obtida com o uso de 8 transferências por ciclo, usando o mesmo clock básico de 66 MHz utilizado por todas as versões do AGP. Apesar do clock ser de 66 MHz, o barramento tem dois sinais complementares AD_STBS e AD_STBF, cujas transições são 4 vezes mais rápidas que o clock do barramento AGP. Os instantes de subida desses dois sinais marcam a transferência dos dados, como mostra a figura 15.

**FIGURA 12.15**

Transferência em AGP 8x.

Tabela de compatibilidade

Os chanfros existentes nos conectores AGP da placa de CPU e da placa de vídeo são os indicadores de compatibilidade entre essas placas. Todas as opções de compatibilidade são apresentadas na tabela abaixo:

Placa de CPU	Conector AGP	Compatibilidade
AGP 1.0, 3,3 volts	AGP 3,3 V	Opera com placas de vídeo AGP de 3,3 volts. As velocidades suportadas são 1x e 2x, porém isto ainda depende do chipset da placa de CPU e do chip gráfico da placa AGP.

AGP 2.0, 1,5 volt	AGP 1,5 volt	Suporta placas AGP de 1,5 volts. Os modos disponíveis são 1x, 2x e 4x, dependendo do chipset da placa de CPU e do chip gráfico da placa AGP.
AGP 2.0 universal (UAGP)	Universal (UAGP)	Suporta placas AGP de 1,5V e 3,3V, com velocidades de 1x, 2x e 4x, dependendo do chipset da placa de CPU e do chip gráfico da placa AGP.
AGP 3.0	AGP de 1,5 volt	Suporta apenas placas de vídeo AGP 3,0 com velocidades de 4x e 8x, dependendo do chipset da placa de CPU e do chip gráfico da placa AGP.
AGP 3.0 universal	AGP de 1,5 volt	Suporta taxas AGP 1x, 2x, 4x e 8x em modos AGP 2.0 e AGP 3.0.

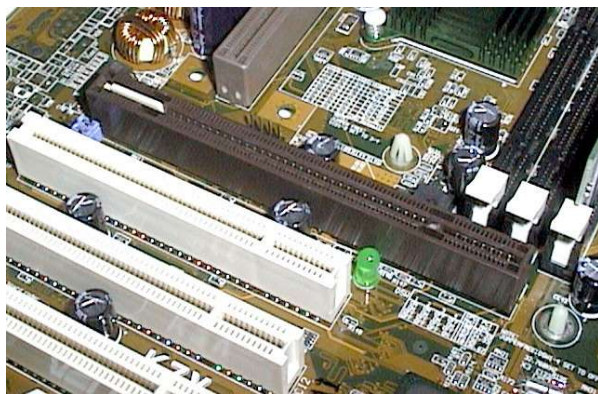
Apesar do padrão AGP 3.0 especificar uma tensão de 0,7 volts, não foi criado um soquete especial para esta nova voltagem. Este padrão utiliza o soquete de 1,5 volts, porém tanto a placa de CPU quanto a placa de vídeo AGP no padrão 3.0 são capazes de identificar a voltagem correta, graças a dois novos sinais introduzidos no barramento: MB_DET e GC_DET. Através do sinal MB_DET, a placa de vídeo AGP pode identificar se a placa de CPU tem slot AGP 3.0. Caso tenha, a placa de vídeo irá operar com 0,7 volts, e caso não tenha, irá operar com 1,5 volt. Da mesma forma, através do novo sinal GC_DET, a placa de CPU no padrão AGP 3.0 irá detectar se a placa de vídeo também é AGP 3.0. Caso seja, passará a operar com 0,7 volts, e caso não seja, usará tensões de 1,5 volts. A tabela abaixo mostra as voltagens que serão usadas na comunicação entre a placa de vídeo e a placa de CPU, em função dos tipos dessas placas e dos seus conectores.

Placa de CPU	Placa de vídeo				
	AGP 3,3V	AGP 1,5V	UAGP	AGP 3.0	UAGP 3.0
AGP 3,3 V	3,3 V	N.E.	3,3V	N.E.	N.E.
AGP 1,5V	N.E.	1,5 V	1,5 V	1,5 V	1,5 V
UAGP	3,3 V	1,5 V	1,5 V	1,5 V	1,5 V
AGP 3.0	N.E.	1,5 V	1,5 V	0,7 V	0,7 V
UAGP 3.0	N.E.	1,5 V	1,5 V	0,7 V	0,7 V

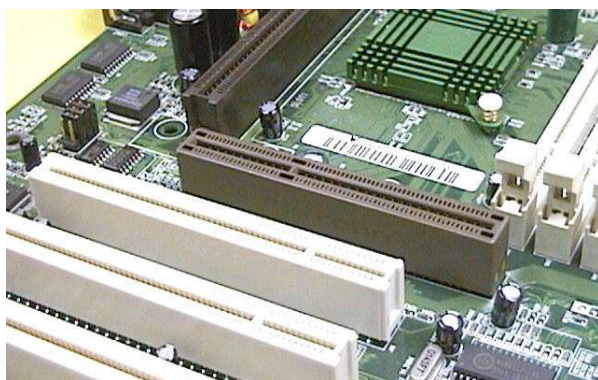
AGP Pro

O slot AGP Pro é uma versão ampliada do AGP, cuja principal característica é a maior capacidade de fornecimento de corrente. Seu slot é maior, com maior número de contatos, e nesses contatos adicionais existem mais linhas de alimentação. O maior fornecimento de corrente é necessário para as placas AGP de maior desempenho, muitas delas chegando a dissipar mais de 50 watts, possuindo inclusive um cooler sobre o seu chip gráfico, similar aos utilizados nos processadores. Podemos ver um slot AGP Pro na figura 16.

Comparando com o slot AGP comum, mostrado na figura 17, podemos observar que o AGP Pro é bem maior. Um slot AGP comum é um pouco menor que os slots PCI. O slot AGP Pro, por sua vez, é visivelmente maior que um slot PCI.

**FIGURA 12.16**

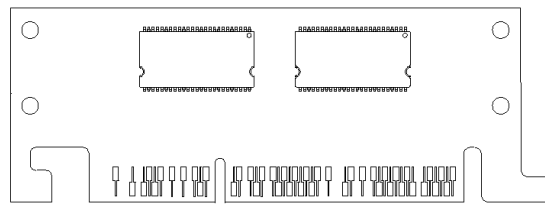
O slot AGP Pro é maior que os slots PCI.

**FIGURA 12.17**

O slot AGP comum é menor que os slots PCI.

Módulo de memória AGP

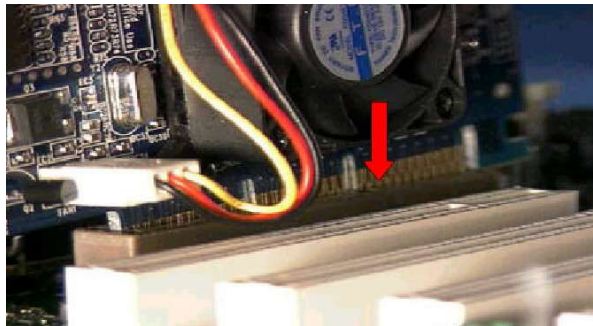
A Intel criou um módulo de memória que pode ser conectado diretamente em um slot AGP. São os módulos AIMM (AGP Inline Memory Module). Placas de CPU com vídeo onboard e que usam memória compartilhada têm como principal desvantagem a queda do desempenho do processador, já que o processador e os circuitos de vídeo concorrem pelo acesso à mesma memória. Uma solução para o problema é instalar uma placa de vídeo AGP, desde que exista slot AGP disponível. Uma outra solução é conectar no slot AGP, uma memória de vídeo exclusiva, que consiste em um módulo AIMM.

**FIGURA 12.18**

Módulo AIMM.

Mecanismo de retenção AGP

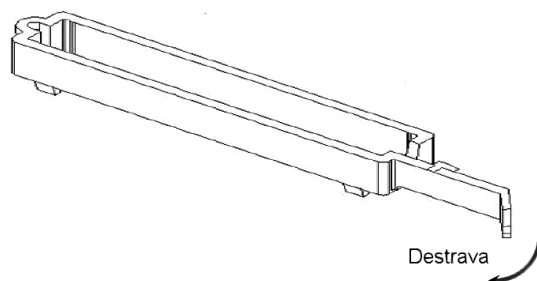
Muitas placas de CPU são acompanhadas atualmente de um mecanismo de retenção AGP. Trata-se de uma peça plástica que é encaixada sobre o slot e temo como objetivo travar a placa de vídeo ou o módulo AIMM. Este módulo, pelo fato de não usar travas como ocorre com os módulos SIMM e DIMM, pode afrouxar com o passar do tempo, devido à vibração. O mesmo pode ocorrer até mesmo com as placas de vídeo AGP, principalmente durante o transporte.

**FIGURA 12.19**

Esta placa AGP está mal encaixada, devido à vibração ou transporte.

O mecanismo de retenção AGP deve ser inicialmente encaixado sobre o slot AGP. Uma vez encaixado, não pode mais ser removido, a menos que seja inutilizado. Ao encaixar uma placa, devemos mover a alça do mecanismo, permitindo o encaixe da placa. Feito isto, liberamos a alça, que irá prender a placa no slot. Para remover a placa do slot, devemos antes mover a alça do mecanismo de retenção, destravando a placa. O mesmo procedimento é usado para encaixar e remover módulos AIMM.

Parte traseira
do PC



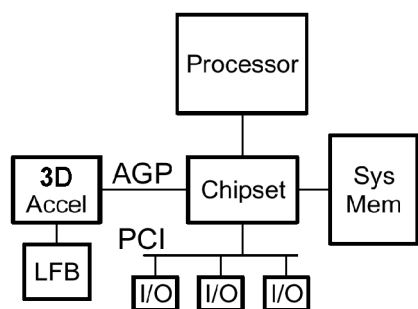
Destrava

FIGURA 12.20

Usando o mecanismo de retenção AGP.

Sinais do barramento AGP

Como vimos, o barramento AGP foi desenvolvido tomando como base o barramento PCI. Podemos dizer que o AGP é o resultado de melhoramentos feitos a partir do PCI, aumentando a velocidade e adicionando algumas funções específicas para operação de placas 3D.

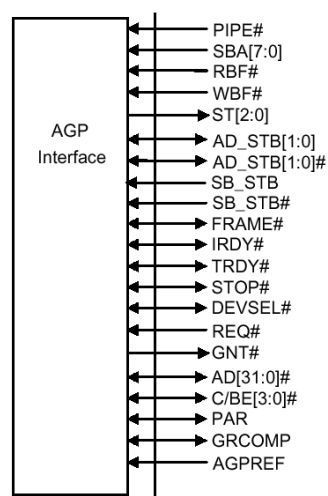


*** 35% ***

FIGURA 12.21

Relação entre os barramentos PCI e AGP.

Os barramentos PCI e AGP são totalmente independentes. A conexão entre o processador, a memória e barramento AGP é feita pelo chipset conhecido como ponte norte. Em geral temos o barramento AGP fisicamente representado por um slot, no qual é encaixada a placa de vídeo AGP. Na figura temos a placa representada como “3D Acell”. O módulo LFB (local frame buffer) é a memória de vídeo existente nesta placa. As placas AGP são capazes de utilizar tanto a sua própria memória local quanto a memória do sistema.



*** 35% ***

FIGURA 12.22

Sinais do barramento AGP.

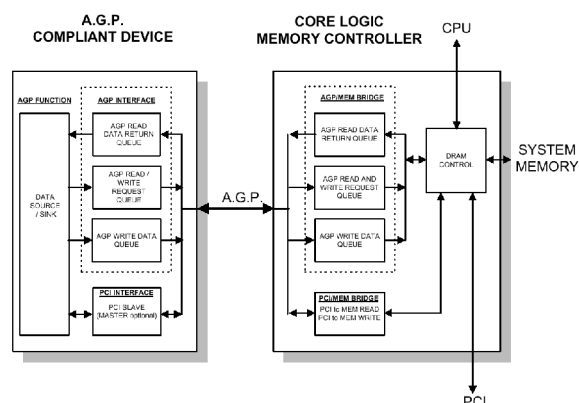
A figura 22 mostra os sinais do barramento AGP. São inteiramente gerados pelo chipset. Observe que existem sinais de entrada, outros de saída e outros bidirecionais. Os principais sinais são descritos a seguir.

Sinal	Descrição
AD0-AD31	Barramento de dados e endereços. Ao contrário de outros barramentos que usam sinais independentes para dados e endereços, o AGP utiliza um único barramento multiplexado. A cada operação, o barramento indicará o endereço, e a seguir os dados. Isso pode parecer causa de lentidão, mas não é. Na maior parte do tempo o barramento AGP necessita que seja indicado apenas o endereço inicial, e os dados vêm a seguir, em modo burst.
C/BE0-C/BE3	Esses 4 sinais têm dupla finalidade. Quando o barramento AD0-AD31 está fornecendo um endereço, os sinais C/BE0-C/BE3 indicam o comando que deve ser realizado (leitura, escrita, etc.). Nos períodos em que dados estão trafegando, esses 4 sinais indicam quais grupos de 8 bits devem ser levados em conta. Ao fazer transferências de 32 bits, os 4 sinais ficam ativados, mas podem ser ativados 1, 2 ou 3 sinais individualmente, permitindo acesso a trechos menores. Por exemplo, para escrever um pixel na tela em um modo gráfico de 256 cores (1 byte por pixel), um único sinal C/BE deve ser ativado para indicar o pixel correto.
IRDY	Initiator Ready. O dispositivo que iniciou a transferência indica que está pronto para receber dados. Desta forma é feita uma sincronização entre a placa de CPU e a placa de vídeo. Quando uma placa é mais lenta que outra, este sinal servirá para introduzir wait states, fazendo com que a mais rápida espere pela mais lenta, quando necessário.

TRDY	Target Ready. Tem função semelhante à do sinal TRDY, mas diz respeito ao dispositivo que está sendo acessado. Por exemplo, quando a placa de CPU quer ler dados da memória de vídeo, a placa de CPU opera como Initiator ou Master, e a placa de vídeo opera como Target (alvo). A placa de vídeo, que neste caso opera como Target, usará este sinal para indicar que está com um dado disponível, ou que está pronta para receber um novo dado. Este protocolo faz a sincronização de placas de desempenhos diferentes, mediante a introdução de wait states, quando necessário. Não adianta por exemplo operar em modo AGP 8x e a placa de vídeo usar memórias lentas. Isto fará com que sejam gerados wait states, resultando em desempenho inferior.
SBA0-SBA7	Os sinais AD0-AD31 podem ser usados para endereços e dados, ou podem ser usados exclusivamente para dados. Nesse caso os endereços são enviados através do barramento SBA0-SBA7. O ideal seria que esses endereços usassem um barramento também de 32 bits, mas são usados apenas 8 para que o número total de pinos do conector não seja exagerado. Este barramento opera em alta velocidade, assim como o barramento de dados. Em modo 2x, por exemplo, cada período de clock fornecerá 16 bits, e em 4x cada período fornecerá 32 bits. Desta forma temos barramentos de dados e endereços independentes. Apesar do barramento de endereços ter apenas 8 bits, isto não prejudica o desempenho, já que na maioria das operações o endereço é especificado apenas no início da transferência de uma longa sequência de dados.
AD_STB	Strobe para o barramento de dados. Este sinal é usado para sincronizar o tráfego através do barramento AD0-AD31.
SB_STB	Este sinal é usado para sincronizar as transferências no barramento SBA0-SBA7.
PIPE	Este sinal deve ser ativado durante o uso do barramento AD0-AD31. O número de ciclos nos quais este sinal fica ativo está relacionado com o número de transferências a serem efetuadas.
FRAME	Tem a mesma função que o sinal PIPE. A diferença é que FRAME é usado para transferências no modo PCI, enquanto PIPE é usado para transferências no modo AGP.

O AGP não é simplesmente baseado no PCI. O barramento AGP contém todos os sinais do barramento PCI, e pode ser portanto tratado como um dispositivo PCI de maior velocidade. Isso possibilitou aos fabricantes de placas de vídeo, converterem rapidamente suas placas de vídeo PCI para que usem o slot AGP.

Além de suportar os comandos e modos de operação do barramento PCI, o barramento AGP possui modos de operação próprios, mais adequados às operações relacionadas com vídeo 3D.

**FIGURA 12.23**

Seções AGP e PCI.

A figura 23 mostra o diagrama da conexão entre uma placa AGP e a placa de CPU. Note que existe uma seção específica para os comandos PCI e outras especializadas em comandos típicos do AGP.

Modos DMA e Execute

Para gerar uma imagem 2D, uma única estrutura de dados é necessária. É o frame buffer, uma representação linear do conteúdo da tela. Nas placas de vídeo 2D, toda a memória de vídeo é utilizada como frame buffer. Já a representação de imagens tridimensionais exige outras estruturas. O frame buffer é usado, assim como nas placas 2D, para manter uma imagem exata do que é apresentado na tela, porém a memória de vídeo de placas 3D necessita de outras estruturas de dados para uma representação tridimensional completa. O Z-Buffer é a área da memória de vídeo responsável pela representação da terceira coordenada (Z). As texturas são imagens bidimensionais que são aplicadas a polígonos no espaço tridimensional. O Stencil Buffer é uma outra estrutura de dados usada no processamento de imagens refletidas. As placas 3D baseadas no barramento PCI maninham todas essas estruturas de dados na sua própria memória de vídeo. O barramento AGP permite que essas informações sejam armazenadas tanto na memória de vídeo quanto na memória do sistema, localizada na placa de CPU.

Ao operar no modo DMA, todas as estruturas de dados envolvidas na formação das imagens 3D ficam armazenadas no frame buffer da placa AGP. A cada cena 3D representada, as texturas têm que ser carregadas na memória de vídeo, e tanto o Z-Buffer como o Stencil Buffer têm que ser calculados para cada posição e transferidos para a placa de vídeo. Este

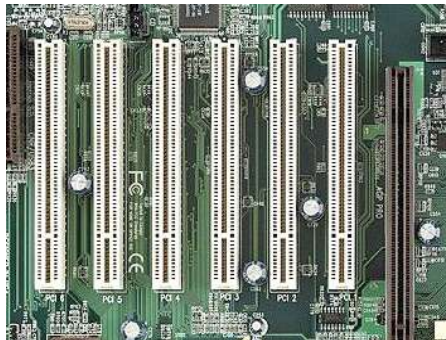
elevado tráfego de dados ao longo do barramento AGP pode prejudicar o desempenho geral.

O outro modo suportado pelo barramento AGP é o DIME (Direct Memory Execute). Neste modo as estruturas de dados podem ficar armazenados na memória do sistema. Tanto o frame buffer como o Z-Buffer, o Stencil Buffer e as texturas podem ficar na memória da placa de vídeo ou na memória do sistema. Decidir se cada estrutura fica em uma ou em outra memória tem impacto direto no desempenho gráfico. Por exemplo, é sempre melhor manter o frame buffer na memória de vídeo, e não na memória do sistema, já que esta área é constantemente lida para suas informações serem entregues ao monitor. Para imagens com uma modesta resolução de 800x600 com 16 bits por pixel e taxa de atualização de 75 Hz para o monitor, o tráfego de dados do frame buffer é superior a 500 MB/s. É um tráfego muito elevado em comparação com os 800 MB/s suportados por um barramento de memória PC100, e mesmo comparando com os 1066 MB/s oferecidos pelas memórias PC133. Este é um sério problema do vídeo onboard com memória compartilhada. Ao usar o frame buffer na memória da placa de vídeo, tanto a resolução como o número de cores e a taxa de atualização podem ser elevadas, sem sobrecarregar a memória do sistema.

As demais estruturas de dados podem ficar na memória de vídeo ou na memória do sistema, e impacto no desempenho dependerá da utilização. Mantê-las todas na memória de vídeo (modo DMA) resulta em alto desempenho, mas tem como desvantagem a perda de tempo realizando as todas as transferências. Note que apesar do padrão AGP oferecer o modo DIME, nem todos os chips gráficos o suportam, e nem todos os programas 3D o utilizam. Para compensar a ausência do modo DIME, a placa de vídeo deve ter uma generosa quantidade de memória de vídeo.

Barramento PCI

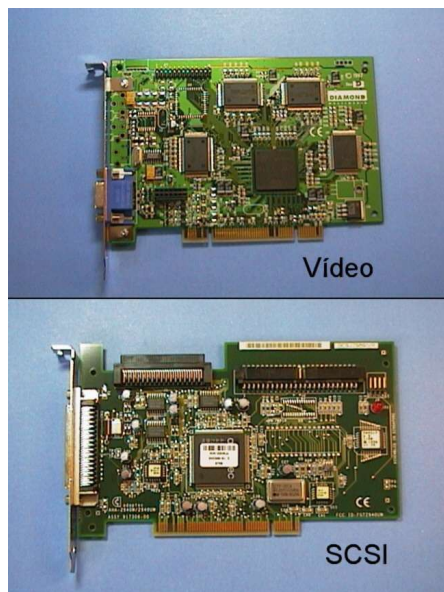
A figura 9 mostra os conectores usados no barramento PCI (Peripheral Component Interconnect). Nas placas de CPU modernas podemos encontrar 3, 4, 5 ou 6 slots PCI. Em algumas placas mais simples, tipicamente aquelas que têm “tudo onboard”, podemos encontrar apenas um ou dois slots PCI.

**FIGURA 12.24**

Slots PCI.

Nos slots PCI, conectamos placas de expansão PCI. Alguns exemplos típicos de placas de expansão PCI são:

- Placa de vídeo (SVGA)
- Placa de interface SCSI
- Placa de rede
- Placa digitalizadora de vídeo



*** 35% ***

FIGURA 12.25

Placas de expansão PCI: uma placa de vídeo e uma controladora SCSI.

É importante notar que Barramento PCI não é sinônimo de Slot PCI. O Barramento PCI é um conjunto de sinais digitais que partem do chipset e do processador, e atingem tanto as placas de expansão, através dos slots, como circuitos da placa de CPU. Por exemplo, as interfaces para disco rígido e as

interfaces USB embutidas na placa de CPU são controladas através do barramento PCI, apesar de não utilizar os slots.

Tipos de PCI

Na maioria das placas de CPU, o barramento PCI opera com 32 bits e utiliza um clock de 33 MHz. Podemos entretanto encontrar algumas variações. Em placas de CPU antigas podemos encontrar o barramento PCI operando com 25 ou 30 MHz. Em placas que operam com overclock externo, o clock do barramento PCI pode estar indevidamente programado para 37,5 MHz ou 41,6 MHz, resultando em mau funcionamento. Existem ainda as placas de CPU de alto desempenho, próprias para servidores, nos quais temos barramentos PCI de 64 bits e 66 MHz. A taxa de transferência máxima teórica do barramento PCI depende portanto das suas características. A tabela abaixo mostra as principais opções:

Clock	Número de bits	Taxa de transferência
33 MHz	32	133 MB/s
33 MHz	64	266 MB/s
66 MHz	32	266 MB/s
66 MHz	64	533 MB/s

Logo na primeira versão do PCI, apesar de operar com 32 bits, estava previsto o funcionamento com 64 bits. O slot PCI de 64 bits é diferente do 32 bits. Possui duas seções, sendo uma análoga ao de 32 bits e outra menor, com os sinais necessários ao funcionamento em 64 bits.

Parte traseira do gabinete

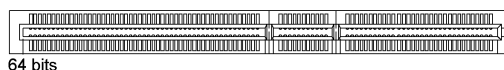
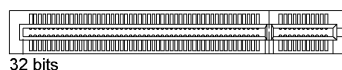


FIGURA 12.26

Slots PCI de 32 e de 64 bits.

Mais recentemente o barramento PCI sofreu uma revisão e agora suporta também a operação em 66 MHz, tanto em 32 como em 64 bits. Este aumento foi motivado principalmente pelas novas interfaces SCSI, capazes de operar com taxas de 320 MB/s. O barramento PCI de 66 MHz e 64 bits, com sua taxa de transferência máxima de 533 MB/s, suporta perfeitamente este tipo de interface. Obviamente só encontramos esta configuração em servidores e estações de trabalho de alto desempenho.

Além das opções usuais de 32 e 64 bits, 33 e 66 MHz, podemos encontrar em algumas placas de CPU antigas (e mesmo em algumas mais novas), alguns clocks incomuns em uso no barramento PCI. A tabela abaixo mostra algumas dessas opções e as respectivas taxas de transferência. Note que valores acima de 33 MHz são considerados overclock. Mesmo que a placa de CPU possa gerar esses valores, nem sempre as placas de expansão PCI irão suportá-los.

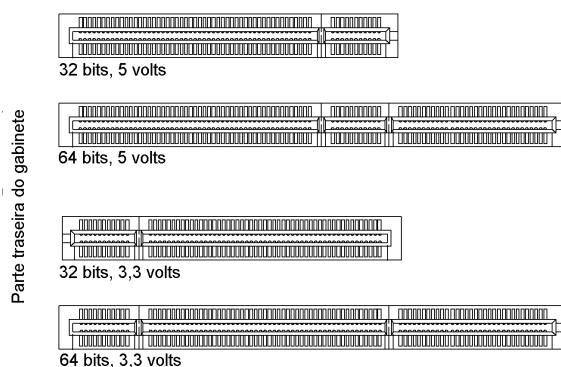
Clock	Número de bits	Taxa de transferência
25 MHz	32	100 MB/s
30 MHz	32	120 MB/s
33 MHz	32	133 MB/s
37,5 MHz	32	150 MB/s
41,6 MHz	32	166 MB/s
36,6 MHz	32	146,4 MB/s
38,3 MHz	32	153,2 MB/s

Nas placas de CPU Pentium antigas, o clock do barramento PCI era sempre igual à metade do clock externo do processador. Com clock externo de 66 MHz, o resultado era o barramento PCI operando a 33 MHz. No processador Pentium-75, por exemplo, o clock externo é de 50 MHz, portanto o barramento PCI passa a operar com apenas 25 MHz, já que tais placas são programadas para usar no barramento PCI, a metade do clock do processador. Da mesma forma no Pentium-90, Pentium-120 e Pentium-150, o clock externo é 60 MHz, portanto o barramento PCI opera com 30 MHz. Quando é feito overclock externo nessas antigas placas, obrigando o processador a operar com 75 ou 83 MHz externos, o barramento PCI opera com a metade desses valores, o que corresponde a 37,5 MHz e 41,6 MHz. Na maioria das placas de CPU com clock externo de 100 MHz, o barramento PCI opera com 1/3 do clock externo do processador, resultando em 33 MHz, mas pode resultar em valores maiores quando é usado overclock. Muitas placas de CPU mais recentes têm geradores de clock independentes para o processador e para o barramento PCI. Desta forma o barramento PCI pode ser mantido em 33 MHz, mesmo que o processador esteja usando overclock externo.

Voltagens do PCI

Na sua especificação original, o barramento PCI podia operar a partir de uma alimentação de +5 volts ou de +3,3 volts. Para impedir a conexão de placas incompatíveis, os slots PCI utilizam um chanfro que deve coincidir com outro chanfro localizado na placa de expansão. Nos slots de 3,3 volts, este chanfro fica mais próximo da parte traseira da placa de CPU. Nos slots

de 5 volts o chanfro fica no lado oposto. A figura 27 mostra os tipos de slots PCI em função da sua voltagem.

**FIGURA 12.27**

Slots de 5 volts e de 3,3 volts.

Os slots e placas de 5 volts são os mais usados. Praticamente não encontramos versões de 3,3 volts. Podemos entretanto encontrar algumas placas de expansão PCI que são universais, podendo funcionar tanto em slots de 5 como de 3,3 volts. Essas placas possuem dois chanfros no seu conector, como mostra a figura 28.

**FIGURA 12.28**

Placa PCI universal de 5 e 3,3 volts.

Master e Target

As transferências realizadas no barramento PCI são sempre feitas entre dois dispositivos. Um deles é o Bus Master ou Initiator, o outro é o Target (alvo). O Bus Master é aquele que toma a iniciativa de solicitar a transferência do Target. O Bus Master é o responsável pela geração de todos os sinais de endereços, controles e dados (no caso de escrita). Quando o computador é ligado, o processador da placa mãe é o Bus Master. Ele é o responsável por comandar a inicialização do hardware, incluindo o chipset e as placas de

expansão. Outros dispositivos também podem atuar como Bus Masters, como interfaces IDE e interfaces SCSI.

Para que um dispositivo opere como Bus Master, precisa inicialmente pedir permissão ao Bus Arbitrer, que faz parte do chipset. Do Bus Arbitrer partem várias linhas REQ (request) e GNT (Grant). Cada par REQ/GNT é ligado a um dispositivo PCI diferente. No caso dos slots, cada um deles tem um sinal REQ próprio, através do qual envia ao chipset o pedido de requisição para que tome o controle do barramento. Ao receber esta requisição, o Bus Arbitrer verifica se é possível dar a permissão e envia para o dispositivo que a requereu, o sinal GNT apropriado. Cada dispositivo e cada slot possui seu próprio sinal GNT. A maioria dos chipsets atuais possuem 4, 5 ou 6 pares REQ/GNT, através dos quais podem ser conectados 4, 5 ou 6 dispositivos PCI capazes de atuar como Bus Masters. Existem entretanto algumas placas nas quais apenas alguns slots PCI são Bus Masters, enquanto outros slots atuam apenas como Target. Ao passar por problemas de funcionamento de placas PCI, experimente a troca do slot. Consulte o manual da placa de CPU para verificar se todos os slots podem operar como Bus Masters. Placas de som, placas de rede, controladoras SCSI e digitalizadores de vídeo, bem como interfaces rápidas em geral, operam em modo Bus Master. Placas de vídeo PCI e modems operam como Target, portanto podem ser remanejadas para slots que não suportam Bus Mastering.

Sinais do barramento PCI

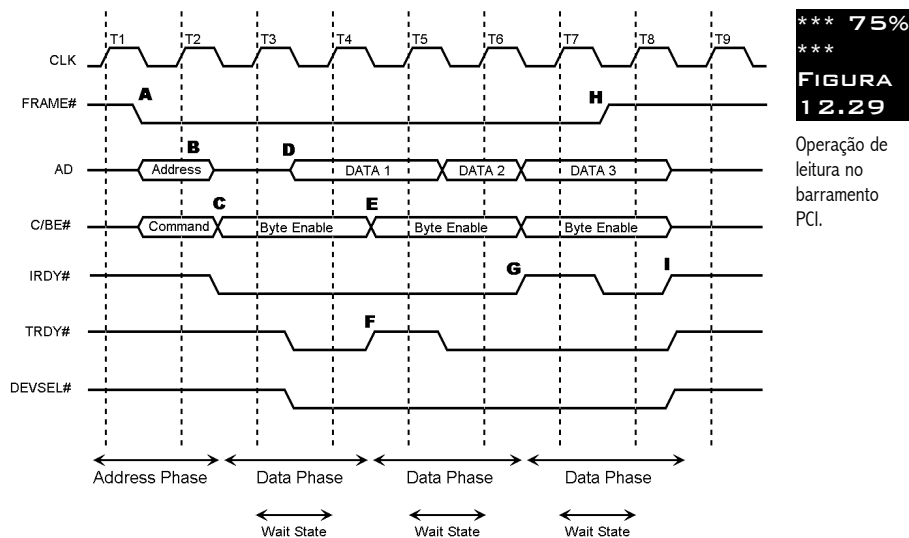
A seguir apresentamos alguns dos sinais de maior importância no barramento PCI, juntamente com sua descrição.

Sinal	Descrição
AD0-AD31	Barramento de dados e endereços multiplexados. No início de uma transferência, este barramento indica o endereço, e na fase seguinte, os dados. Como muitas transferências são feitas em modo burst, não existe queda de desempenho perceptível pelo fato de ser usado um único barramento para dupla função.
C/BE0-C/BE3	Durante a fase de endereço, esses 4 sinais indicam o comando a ser realizado (leitura, escrita, etc.). Na fase de dados, esses 4 bits indicam quais bytes dos 32 bits do barramento de dados devem ser levados em conta. Isso permite, por exemplo, acessar bytes individuais, apesar do barramento de dados ter 32 bits.
FRAME	O Bus Master ativa este sinal para dar início a um ciclo de transferência.
IRDY	Initiator Ready. Indica que o Master está pronto para ler ou enviar dados. Quando este sinal não é ativado, o Target irá esperar tantos wait states quanto forem necessários.

TRDY	Target Ready. Indica que o Target está pronto para receber dados (escrita) ou que o dado lido já está disponível (leitura). Quando este sinal não é ativado, o Master irá gerar tantos wait states quando forem necessários.
DEVSEL	Ativado pelo Target quando reconhece o seu endereço. Desta forma o Master pode saber se o dispositivo Target está ativo ou presente no barramento.
REQ	Requisição enviada ao Bus Arbitrer, para que o dispositivo se torne Bus Master. Cada dispositivo tem seu próprio sinal REQ.
GNT	Grant. Através deste sinal o Bus Arbitrer indica ao dispositivo solicitante que o barramento está liberado, permitindo assim que se torne Bus Master. Cada dispositivo tem seu próprio sinal GNT.
INTA, INTB, INTC, INTD	São linhas de interrupção a serem usadas pelos dispositivos PCI. Cada dispositivo e cada slot é ligado a um desses sinais, que podem ser compartilhados, ou seja, uma mesma linha INT pode ser usada por mais de um slot. O padrão PCI prevê o compartilhamento de interrupções.
AD32-AD63	Continuação do barramento de dados e endereços nos slots PCI de 64 bits.
C/BE4-C/BE7	Continuação do barramento de comando e habilitação de bytes nos slots PCI de 64 bits.
REQ64	Resuisição de transferência de 64 bits.
ACK64	Indica que o Target está apto a realizar transferência de 64 bits.

Transferências

Vários tipos de transferência podem ser feitas no barramento PCI. Essas transferências podem envolver um ou múltiplos dados. São chamadas de transactions (transações), mas aqui chamaremos apenas de transferências. A figura 29 mostra o diagrama de tempo de uma operação de leitura, na qual são lidos 3 dados consecutivos.



*** 75%

FIGURA
12.29

Operação de
leitura no
barramento
PCI.

Note que as operações são sincronizadas pelo clock. Durante o período em que o barramento AD traz endereços, temos a fase de endereços (address phase). Uma vez determinado o endereço, são feitas as transferências de dados, entrando então na fase de dados (data phase). Wait states podem ser gerados por solicitação do Target ou do próprio Master, através dos controles IRDY e TRDY. Os eventos que ocorrem nesta transferência são os seguintes:

- O Bus Master inicia a transferência ativando o sinal FRAME, que permanece ativo até que o Target termine sua última fase de dados. O Master também fornece o endereço (AD0-AD31) e o comando (C/BE0-C/BE3).
- O Target reconhece seu endereço e prepara-se para fornecer os dados.
- O Master para de indicar o comando nas linhas C/BE e passa a indicar os controles habilitadores dos bytes desejados.
- O Target ativa a linha DEVSEL para indicar que foi endereçado, fornece o primeiro dado e ativa a linha TRDY para indicar que o dado está pronto. Wait states podem ser gerados se necessário, bastando retardar a ativação de TRDY.
- O Master lê o dado e altera as linhas C/BE, se necessário.

- f) O Target desativa a linha TRDY enquanto busca o próximo dado, gerando mais um wait state. Isto pode ser necessário quando os circuitos do Target não são suficientemente velozes.
- g) Neste exemplo o Master ainda não está pronto para ler o próximo dado, portanto ele desativa o sinal IRDY, gerando mais um wait state que fará o Target manter os dados por mais um ciclo.
- h) Após receber o último dado, o Master finaliza a transferência, desativando a linha FRAME.
- i) O Target é desativado, liberando o barramento de dados e desativando os sinais TRDY e DEVSEL.

Este exemplo dá uma idéia geral de como ocorrem as transferências no barramento PCI. Outros tipos de transferência ocorrem de forma diferente, mas a idéia básica é a mesma.

Dispositivos de alta velocidade

A maioria dos dispositivos ligados ao barramento PCI são capazes de operar como Master. O primeiro deles é o processador da placa de CPU, que através do host bridge (faz parte do chipset), comunica-se com o barramento PCI. Chamamos de bridge um circuito que faz a interface entre dois barramentos. O host bridge é portanto um circuito que faz a comunicação entre o barramento do sistema e o barramento PCI. Através deste circuito o processador pode acessar qualquer outro dispositivo PCI, ou seja, qualquer um deles pode funcionar como Target em transferências nas quais o processador é o Master.

As transferências entre o processador e dispositivos de E/S eram feitas de duas formas nos PCs com arquiteturas antigas:

- E/S programada
- E/S por DMA

Esses dois métodos estão representados na figura 30. Na E/S programada, cabe ao processador obter os dados na interface e colocá-los na memória (nas operações de entrada), e obtê-los na memória e entregá-los à interface apropriada (nas operações de saída). A vantagem desta técnica é a simplicidade de circuitos, e a desvantagem principal é que deixa o

processador muito ocupado quando são transferidos muitos dados em seqüência.

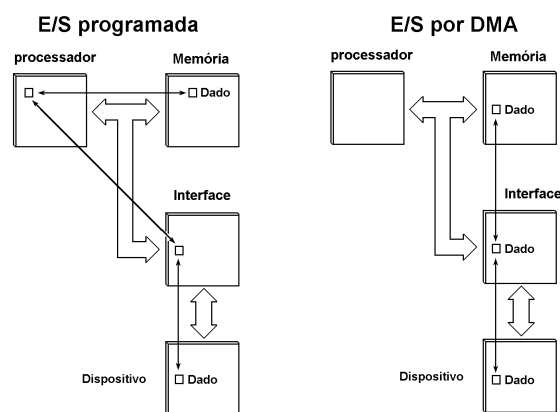


FIGURA 12.30

E/S programada e E/S por DMA.

Na E/S por DMA, um circuito especial chamado controlador de DMA toma o controle do barramento do processador e realiza a transferência, gastando o mínimo tempo possível. A transferência de um dado através de DMA dura um único ciclo, sem monopolizar o barramento. Enquanto na E/S programada o processador pode ficar ocupado entre um dado e outro, testando se o próximo dado já pode ser transferido, na E/S por DMA o processador pode executar outras tarefas enquanto o controlador de DMA se encarrega da transferência. O controlador de DMA pode tomar o controle do barramento do sistema graças a dois sinais existentes nos processadores: HOLD e HLDA (Hold Acknowledge). Quando a interface tem um dado pronto para entregar, envia um comando para o controlador de DMA (DMA Request), que por sua vez ativará o sinal HOLD do processador. O processador termina a execução do ciclo em andamento e entra em tristate, “congelando” sua atividade, e envia para o controlador de DMA o sinal de HLDA, reconhecendo o estado de HOLD. A partir daí o controlador de DMA, que antes estava em tristate, passa a gerar os sinais de endereço, dados e controle, passando a ser o “mestre” do barramento. O controlador de DMA envia para a interface, um sinal de DMAACK (DMA Acknowledge, ou reconhecimento de DMA). A interface pode então entregar o dado para o barramento e enviando para a memória (nas operações de entrada), ou então obter o dado da memória (nas operações de saída).

Quanto mais lento é um periférico, mais adequada é a transferência por E/S programada, e quanto mais rápido, mais adequada é a transferência por

DMA. Vejamos alguns exemplos, considerando o uso de um barramento externo de 100 MHz para o processador. Supondo que as memórias operem com temporização 2-1-1-1, teríamos cerca de 80 milhões de transferências de 64 bits por segundo. Considere alguns tipos de interface e o “tempo” entre transferências consecutivas:

Interface	Taxa de transferência	Intervalo entre transferências, medido em acessos à memória
Teclado	30 caracteres/s	2.600.000
Modem, 56k bps	5 kB/s	16.000
Placa de som	176 kB/s	1.800
Disquetes	45 kB/s	1.700
Paralela padrão	150 kB/s	533
Rede, 10 Mbits/s	1,25 MB/s	256
USB 1.1	1,5 MB/s	200
Paralela EPP/ECP	2 MB/s	40
Rede, 100 Mbits/s	12,5 MB/s	25
IDE, PIO Mode 4	16,6 MB/s	19
IDE, ATA-33	33,3 MB/s	9
Firewire	50 MB/s	6
USB 2.0	60 MB/s	5
IDE, ATA-66	66,6 MB/s	5
IDE, ATA-100	100 MB/s	3

Veja por exemplo o caso do teclado, que fornece uma taxa máxima de 30 caracteres por segundo (quando usamos o REPEAT). Entre a chegada de dois caracteres consecutivos, transcorrem cerca de 2,6 milhões de ciclos. Não teria sentido o processador ficar monopolizado entre a chegada de dois caracteres consecutivos, testando milhares de vezes se o próximo caractere já chegou. Para isso são usadas interrupções. O processador não testa se chegou o próximo caractere, ele continua executando outras tarefas, e a interface de teclado gera uma interrupção quando uma tecla é pressionada. Para atender à interrupção o processador precisa salvar o seu contexto (armazenar o conteúdo de todos os registradores internos), atender à interrupção e retornar ao processamento original, o que consome entre 50 e 100 ciclos. Como o intervalo neste caso é de 2,6 milhões de ciclos, vale a pena para o processador executar outras tarefas e ser interrompido quando chegar um caractere. Por isso podemos, por exemplo, tranquilamente digitar um texto enquanto o processador de textos faz correção gramatical ou salva o arquivo automaticamente. Portanto a E/S programada com o uso de interrupções é um método bem adequado para a operação do teclado.

Um modem de 56k bps fornece um dado a cada 16.000 ciclos. O uso de E/S programada com interrupções também é adequado aqui. Quando chegar um carácter serão gastos entre 50 e 100 ciclos para atender à interrupção que lê o carácter (ou que o obtém da memória para transmitir). Por isso tipicamente os modems operam com E/S programada, sincronizada por interrupções.

A situação da placa de som é mais delicada. Operando com o som de qualidade máxima, transcorreriam 1800 ciclos entre duas transferências consecutivas, o que tornaria possível a operação por E/S programada, com interrupções. Entre dois dados consecutivos, descontando 100 ciclos para atendimento da interrupção, sobriam ainda 1700 para outros processamentos. O problema é que estamos levando em conta um barramento de 100 MHz, e quando as placas de som foram criadas, o barramento usado era o do 386, em versões tão lentas como 16 MHz e usando 2 ciclos por transferência. Nesse caso ocorriam cerca de 150 ciclos entre duas transferências consecutivas, o que tornava inviável o funcionamento por E/S programada com interrupções, já que cada atendimento de interrupção tem uma latência de cerca de 100 ciclos. Uma solução seria usar E/S programada sem interrupções, porém este método não permite que o processador realize outras tarefas. Por isso foi escolhida para as placas de som, a E/S por DMA. O processador não se envolve com a transferência, apenas a ativa no início e é informado da sua finalização, depois que todos os bytes foram transmitidos ou recebidos. Até hoje as placas de som operam com E/S por DMA, apesar dos barramentos atuais suportarem esta operação com E/S programada por interrupções.

Outro caso clássico é o da interface de disquetes. São cerca de 1800 ciclos entre bytes consecutivos, mas nos PCs originais, que faziam transferências lentas como 4 MB/s, o intervalo entre bytes consecutivos era de pouco mais de 100 ciclos. As interfaces de disquete da época usavam DMA ou E/S programada sem interrupções (também chamada de pooling, já que o processador passa os intervalos interrogando a interface para checar se os dados já chegaram). Para permitir maior desempenho, a IBM optou por utilizar para a sua interface de disquetes, a E/S por DMA, característica que é mantida até nos PC atuais, por questões de compatibilidade.

A interface paralela padrão opera nos PCs modernos com cerca de 500 ciclos entre bytes consecutivos, mas nos PCs antigos este período era reduzido a cerca de 20 ciclos, inviabilizando a E/S programada por interrupções. As opções viáveis eram DMA e E/S programada por pooling, e este último foi o adotado pela IBM. A interface de impressora usa interrupções apenas para indicar condições anormais, como buffer cheio,

falta de papel e outros erros. Devido à operação por pooling, o processador fica totalmente ocupado quando envia dados à impressora. Podemos inclusive notar algumas paradas no processamento normal durante a impressão. Interfaces paralelas modernas, operando nos modos EPP/ECP transmitem e recebe dados a cada intervalo de 40 ciclos, e a única opção viável de funcionamento é a E/S por DMA. A E/S programada por pooling deixaria o processador monopolizado durante a impressão.

Interfaces de rede a 10 Mbits/s oferecem 256 ciclos entre dados consecutivos (barramento de 100 MHz), ou um número menor ainda nos barramentos mais lentos. Por isso as interfaces de rede operam com E/S por DMA ou com E/S programada por pooling. Para não monopolizar o uso do processador ao operar com pooling, as interfaces de rede utilizam um buffer que armazena um bloco inteiro de dados. Assim consegue uma utilização mais eficiente. Ao transmitir dados, o processador os coloca rapidamente no buffer e dispara a transmissão, ficando assim liberado. A interface de rede vai interromper o processador quando terminar a transmissão do bloco de dados. Da mesma forma durante a chegada de dados a interface os armazena no seu buffer e interrompe o processador quando o bloco inteiro estiver pronto. O uso de buffers para acelerar o desempenho, ocupando menos o processador, e interrupções apenas para sinalizar o término da operação, é possível apenas para interfaces orientadas por blocos, ou seja, que transmitem sempre blocos de dados, como é o caso da interface de rede. Em contraste temos os dispositivos orientados a bytes ou caracteres, como os modems, teclados e impressoras, que requerem atenção individual do processador para cada byte que transita. Interfaces de rede de 100 Mbits/s podem operar com os mesmos métodos usados pelas interfaces de 10 Mbits/s.

Dispositivos mais rápidos não suportam a E/S programada por interrupções. Devem usar DMA ou E/S programada por pooling. Veja por exemplo o caso da interface IDE operando em PIO Mode 4. Com apenas 19 ciclos entre dados consecutivos, tem tempo suficiente para fazer um pooling: receber o dado, guardar na memória, incrementar os contadores, testar se chegou o último dado e caso não tenha chegado, voltar ao pooling para esperar o próximo dado. A seguir mostramos um trecho de programa (linguagem Assembly) no qual é feita uma leitura por pooling. O método é usado tanto para interfaces IDE como para outras que operem desta forma. As diferenças ficam por conta do número de bytes de cada bloco, dos endereços de dados e status e do bit indicador de dado pronto.

```
...  
MOV CX, 512          ; Vai ler 512 bytes  
MOV DI, BUFFER       ; DI aponta para o destino
```

```

POOLING:      IN AL, STATUS      ; Interroga a interface
               TEST AL,BIT       ; Testa o bit de dado pronto
               JZ POOLING        ; Volta se não pronto
               IN AL, DADO       ; Lê o dado
               STOSB             ; Armazena no buffer
               DEC CX            ; Decrementa contagem
               JNZ POOLING       ; Volta se não chegarm 512
FIM:          ....              ; Continua o processamento

```

A programação da E/S por DMA é um pouco mais complexa. É preciso programar o controlador de DMA indicando o canal a ser usado (cada canal é ligado a uma interface), o endereço de memória para os dados e o número de bytes. A seguir a interface recebe o comando a ser realizado. Terminados esses preparativos, o processador está livre para executar outras tarefas. O controlador de DMA e a interface farão todo o trabalho. Nos instantes em que dados estiverem prontos, o controlador de DMA coloca o processador em HOLD e faz seu acesso à memória. Terminada a transferência do bloco, a interface gera uma interrupção para avisar seu término ao processador.

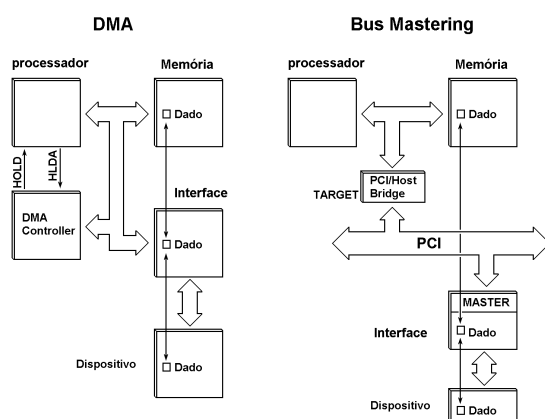
Interfaces ainda mais rápidas não conseguem operar com E/S programada por pooling. Os 9 ciclos entre transferências consecutivas em uma interface ATA-33 são um período muito pequeno para fazer o pooling, mesmo operando a 100 MHz. Mais crítica seria a situação nos barramentos de 66 MHz.

A solução para o problema seria utilizar para esses dispositivos mais rápidos, a E/S por DMA. Não poderia neste caso ser usado o controlador de DMA padrão do PC. Este controlador opera nos mesmos padrões do barramento ISA, por questões de compatibilidade. No IBM PC XT operava mediante um clock de 5 MHz, e no AT, com 8 MHz. Seria possível utilizar controladores de DMA mais rápidos, mas não seriam 100% compatíveis com o padrão PC, baseado no controlador 8237A. Usando este controlador, os acessos à memória seriam feitos com os 8 MHz do barramento ISA, com ciclos extremamente longos para os padrões atuais, inviabilizando o seu uso com os dispositivos modernos.

Para usar DMA com os dispositivos de E/S mais modernos, a arquitetura do PC deveria ser modificada para incluir um novo controlador de DMA, ligado diretamente ao barramento do processador, e operando com clocks mais elevados, por exemplo, usando o mesmo clock do processador. Tecnicamente isto seria possível, porém ao invés disso foi adotada uma solução mais simples e mais elegante, que é utilizar o método Bus Mastering no barramento PCI.

Bus Mastering

O barramento PCI foi construído de tal forma que seus dispositivos podem operar como Bus Masters. Um Bus Master pode realizar o mesmo papel que um controlador de DMA, já que assume o total controle do barramento. É como se cada dispositivo PCI capaz de operar como Bus Master tivesse um controlador de DMA embutido.

**FIGURA 12.31**

DMA x Bus Mastering.

OBS: Na figura 31 a sincronização entre o controlador de DMA é feita pelos sinais HOLD e HLDA. Processadores modernos utilizam controles semelhantes para fazer este controle, apesar de não serem chamados de HOLD e HLDA.

OBS: O barramento PCI não utiliza operações de DMA. Entretanto operações de DMA que ocorrem no barramento ISA convertidas em transferências no barramento PCI, nas quais o Master é o PCI/ISA Bridge, e o Target é o PCI/Host bridge.

A figura 31 mostra o esquema das transferências por DMA e por Bus Mastering. Assim como ocorre com o DMA, o Bus Mastering não monopoliza o processador. A transferência de dados entre a memória e um periférico, usando o barramento PCI, tem as seguintes características:

a) A interface opera como Bus Master, enquanto o PCI/Host Bridge opera como Target. O Master pode ser uma interface de rede, uma interface IDE, uma interface SCSI, uma interface Firewire ou qualquer outra que esteja preparada para operar neste modo. O PCI/Host Bridge, atuando como Target, fará os acessos à memória através do barramento do processador, quando necessário.

b) Mesmo que o barramento PCI esteja operando com seu tráfego máximo (133 MB/s, com 32 bits e 33 MHz, por exemplo), o barramento do sistema não será exigido 100% do tempo. O PCI/Host Bridge tomará conta do

barramento do processador apenas quando necessário, deixando este barramento livre na maior parte do tempo para que o processador tenha acesso à memória, realizando seus processamentos normais. Explicando de forma simples, com o barramento PCI exigindo 133 MB/s e o barramento do processador sendo capaz de operar com 66 MHz e 64 bits (533 MB/s), estaríamos ocupando apenas 25% do tráfego total do barramento do processador, que continua com 75% do tráfego para si próprio. Levando em conta que a maioria dos acessos são feitos na cache L2, o processador continua operando normalmente enquanto uma transferência da memória por Bus Mastering está em andamento.

c) Para evitar que uma transferência monopolize o barramento, o seu controlador pode suspender temporariamente uma transferência muito longa, dando chance a outros Bus Masters de terem acesso ao barramento PCI. Transferências suspensas podem continuar de onde pararam depois que outros Bus Masters foram atendidos.

Plug and Play nos barramentos PCI e ISA

Para que interfaces e placas de expansão funcionem, precisam que sejam configurados os recursos de hardware a serem usados, e que sejam instalados os drivers apropriados. Esses recursos de hardware são:

- Endereços de memória
- Endereços de E/S
- Linhas de interrupção
- Canais de DMA

Nos PCs antigos cabia ao usuário ou ao técnico instalador, a configuração desses recursos de forma manual, através de jumpers, microchaves ou de programas de configuração específicos para cada dispositivo. Um grande esforço da Microsoft, Intel e outros fabricantes foi feito no sentido de possibilitar a configuração automática desses recursos, sem que o usuário precise intervir. As configurações são feitas de forma automática pelo BIOS da placa de CPU e/ou pelo sistema operacional. Cabe ao sistema operacional providenciar a instalação dos drivers corretos para cada dispositivo instalado.

O barramento PCI é totalmente Plug and Play (PnP). Isto significa que qualquer dispositivo PCI é beneficiado pela configuração automática oferecida pelo Plug and Play. Já o barramento ISA não possui recursos PnP nativos, ou seja, as placas de expansão ISA antigas, bem como os dispositivos das placas de CPU ligadas ao barramento ISA (interfaces seriais

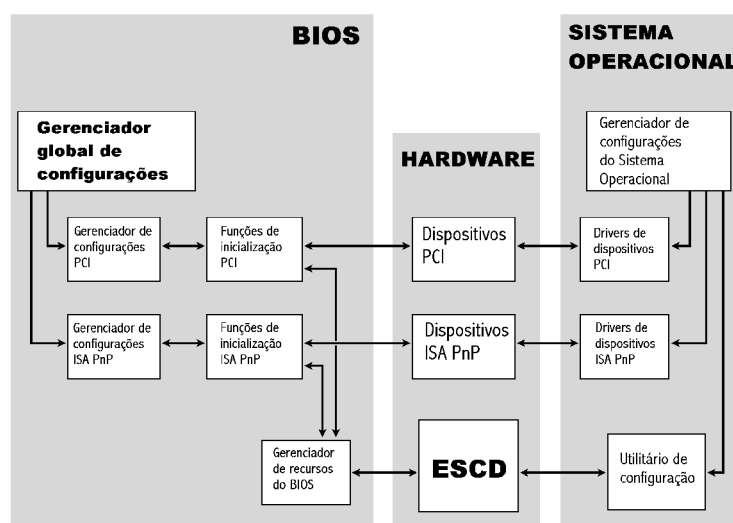
e paralelas, por exemplo), não possuem recursos de configuração automática.

A arquitetura Plug and Play foi criada de forma que não apenas dispositivos PCI possam usá-la. Novos dispositivos ISA puderam passar a utilizar as mesmas configurações automáticas disponíveis nas placas PCI. Placas de expansão ISA produzidas a partir de 1995, bem como interfaces existentes nas placas de CPU modernas e ligadas ao barramento ISA, também são Plug and Play.

A distribuição de recursos de hardware de forma automática depende de três fatores:

- 1) Que exista uma área no sistema que indique quais são os recursos de hardware disponíveis e quais estão em uso por quais interfaces. Esta área existe em todas as placas de CPU modernas, e chama-se ESCD (Extended System Configuration Data). Não existe local padrão para essas informações, mas normalmente ficam localizadas no BIOS, em Flash ROM, ou no chip CMOS.
- 2) Que cada dispositivo PCI ou ISA PnP informe os recursos de hardware de que necessita, e que possa ser programado para utilizar os recursos a ele destinados por um Gerenciador de Recursos.
- 3) Que os dispositivos não Plug and Play (Legacy Devices, ou dispositivos de legado) possam ser configurados de forma manual, porém de forma compatível com os dispositivos Plug and Play.

A arquitetura Plug and Play tem seu funcionamento dependente do BIOS e do sistema operacional. A figura 32 mostra como o BIOS e o sistema operacional interagem para acessar os recursos Plug and Play. Estamos indicando na figura como é feito o acesso a dispositivos de hardware dos tipos PCI e ISA PnP.



*** 75%

FIGURA

12.32

Arquitetura de um sistema Plug and Play.

Quando o computador é ligado, o Gerenciador Global de Configurações, que faz parte do BIOS PnP, ativa o Gerenciador de Configurações PCI e o Gerenciador de Configurações ISA PnP. Cada um desses módulos comandará suas funções de inicialização, que por sua vez enviarão comandos de inicialização apropriados para os dispositivos de hardware PCI e ISA. Durante esta inicialização, todos os dispositivos são interrogados para checar quais são os recursos de hardware necessários (Memória, E/S, IRQ e DMA). As funções de inicialização por sua vez pedem ao Gerenciador de Recursos do BIOS que obtenha através das informações contidas no ESCD, quais recursos podem ser destinados a esses dispositivos. Note que alguns dispositivos são inicializados e configurados nesta etapa, outros são configurados pelo sistema operacional. Tipicamente os dispositivos que precisam estar ativos antes do boot (interfaces de disco, por exemplo) são ativados e configurados pelo BIOS. Dispositivos que funcionam apenas sob o Windows ou outro sistema Plug and Play podem ser configurados posteriormente por este sistema.

O sistema operacional também tem um gerenciador de configurações que engloba as mesmas funções do gerenciador do BIOS, porém com funções mais avançadas. Uma das funções que este gerenciador executa e o BIOS não, é o carregamento dos drivers dos dispositivos. O BIOS não carrega drivers, apenas distribui os recursos de hardware. Em função dos dados existentes no ESCD, o sistema operacional também pode identificar os recursos usados por cada dispositivos. Pode ainda interrogar os dispositivos

que não forma configurados pelo BIOS e distribuir recursos de hardware (Memória, E/S, IRQ e DMA) em função dos dados existentes no ESCD. O sistema operacional também pode empregar utilitários para ler e alterar as informações do ESCD, bem como reprogramar os recursos de hardware usados pelos dispositivos Plug and Play (como é o caso do bom e velho Gerenciador de Dispositivos do Windows).

Como vemos, a arquitetura Plug and Play consiste muito mais em procedimentos de software que elementos de hardware. Por isso foi fácil integrá-la às placas ISA de fabricação mais recente.

Conexões com outros barramentos

A ligação entre barramentos diferentes é feita através de circuitos chamados de bridge (ponte). Esses circuitos ficam localizados no chipset. Como sabemos, o chipset é normalmente formado por dois componentes:

- System Controller ou Northbridge
- Periperal Controller ou SouthBridge

No System Controller encontramos entre vários outros circuitos, a ligação entre o barramento PCI e o barramento do processador (PCI/Host Bridge). No Peripheral Controller encontramos, além de vários outros circuitos, a conexão entre o barramento PCI e outros barramentos mais lentos (como o ISA e o LPC).

Uso de interrupções no barramento PCI

O barramento PCI utiliza 4 linhas de interrupção que podem ser redirecionadas para as interrupções convencionais do PC (IRQs). Um componente do chipset chamado Redirecionador de interrupções é encarregado de fazer este redirecionamento.

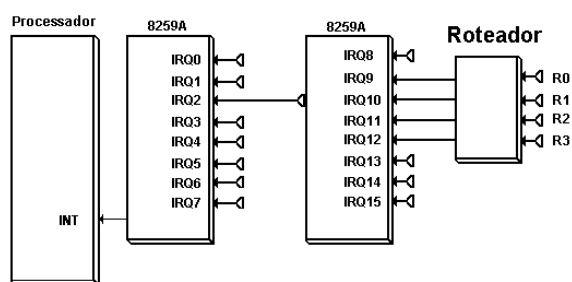


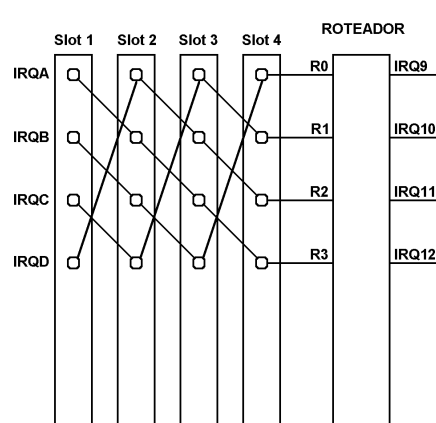
FIGURA 12.33

Ligação do roteador de interrupções do barramento PCI.

A figura 33 mostra como funciona o roteador de interrupções. Chegam a ele 4 linhas de interrupção do barramento PCI: R0, R1, R2 e R3. Este roteador é programado para redirecionar essas 4 linhas para linhas de IRQ apropriadas. No exemplo da figura 33 essas interrupções estão redirecionadas para IRQ9, IRQ10, IRQ11 e IRQ12. Esta escolha de IRQs é feita de forma automática pelo BIOS, mas pode ser alterada através do CMOS Setup, na seção PCI Configuration. Nela podemos escolher entre as várias IRQs disponíveis para serem associadas às 4 linhas de interrupção que chegam do barramento PCI.

As interrupções no barramento PCI podem ser compartilhadas, ou seja, dois ou mais dispositivos podem usar a mesma interrupção. Caberá à rotina do sistema operacional que faz o atendimento a cada interrupção, determinar qual foi o dispositivo que a gerou.

Curiosa é a forma como as linhas de interrupção são ligadas aos slots PCI. Cada slot tem 4 pinos de interrupção: IRQA, IRQB, IRQC e IRQD. Entretanto a ligação com as entradas do roteador não é direta, e sim, alternada. A figura 34 mostra um método típico de ligação.



*** 35% ***

FIGURA 12.34

Conexões de INTA, INTB, INTC e INTD em slots PCI.

A forma de conexão parece confusa, mas é na verdade bastante interessante e faz com que o uso das interrupções seja distribuído uniformemente entre as disponíveis. Sempre que uma placa PCI precisa utilizar uma única interrupção, deve ser obrigatoriamente a IRQA. Se uma placa precisar ter mais de uma interrupção pode utilizar as demais linhas, mas a a IRQA sempre deverá ser usada. Aparentemente isto faria com que a linha IRQA ficasse congestionada, mas não fica graças à forma como as linhas IRQA, IRQB, IRQC e IRQD são levadas até o roteador de interrupções. Note que no esquema da figura 34, a IRQA do slot 1 está ligada em R3, que por sua

vez é direcionada para IRQ12. A IRQA do slot 2 está ligada em R2, que por sua vez é direcionada para IRQ11. A IRQA do slot 3 está ligada em R1, que por sua vez é direcionada para IRQ10, e finalmente a IRQA do slot 4 está ligada em R0, que por sua vez é direcionada para IRQ9. Se cada placa utilizar a IRQA ao mesmo tempo, serão geradas interrupções diferentes IRQ9, IRQ10, IRQ11 e IRQ12.

Quando existem mais de 4 slots PCI, o roteamento obrigatoriamente faz com que uma mesma entrada do roteador esteja ligada a duas linhas INTA, de dois slots PCI diferentes. Por exemplo, em uma placa com 6 slots PCI, as IRQs poderiam estar roteadas da seguinte forma:

R0: INTA/1, INTB/2, INTC/3, INTD/4, INTA/5, INTB/6
R1: INTB/1, INTC/2, INTD/3, INTA/4, INTB/5, INTC/6
R2: INTC/1, INTD/2, INTA/3, INTB/4, INTC/5, INTD/6
R3: INTD/1, INTA/2, INTB/3, INTC/4, INTD/5, INTA/6

Neste caso é comum dizer que “os slots 1 e 5 compartilham a mesma IRQ” e “os slots 2 e 6 compartilham a mesma IRQ”. Isso estaria correto se cada slot usar uma única IRQ, mas sabemos que quando mais de uma IRQ é usada, outras IRQs também serão compartilhadas.

Vimos que a distribuição das IRQs a serem usadas pelo roteador pode ser configurada pelo CMOS Setup, de forma manual, ou ser deixada a cargo do BIOS e do sistema operacional. Podemos checar através do Gerenciador de Dispositivos, quais linhas estão sendo usadas pelo roteador de interrupções. Clicando em Computador / Propriedades / IRQs, podemos ver a lista das IRQs em uso. Aquelas indicadas com “Portador de IRQs para redirecionamento de IRQs PCI” são as que estão em uso pelo roteador.

Nem todos os dispositivos ligados ao barramento PCI utilizam as linhas IRQA, IRQB, IRQC ou IRQD. Por exemplo, as interfaces IDE primária e secundária, apesar de serem dispositivos PCI, utilizam as interrupções IRQ15 e IRQ14, respectivamente, que não passam pelo roteador, sendo ligadas diretamente ao controlador de interrupções. Da mesma forma, as interfaces USB que também são dispositivos PCI podem estar diretamente ligadas a uma interrupção, ou então usarem uma das IRQs disponíveis para os slots PCI.

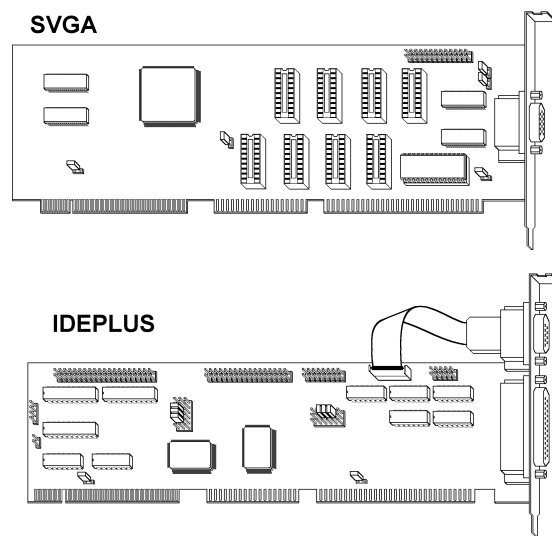
Barramento VLB

Antes do surgimento do barramento PCI, alguns outros barramentos forma usados nos PCs, oferecendo taxas de transferência mais elevadas. O barramento MCA e o EISA foram dois padrões adotados entre o final dos anos 80 e o início dos anos 90. O MCA (Microchannel Architecture) era usado em PCs IBM PS/2 e teve várias versões, de 16 e 32 bits, operando a 10 e 16 MHz. O barramento EISA (Enhanced ISA) foi desenvolvido por diversas empresas que precisavam de um barramento mais rápido mas não podiam usar o MCA, que era barramento proprietário da IBM. O EISA opera com 32 bits e usa clocks entre 6 e 8,33 MHz.

Algumas placas de CPU chegaram a utilizar barramentos locais de alta velocidade para expansões de memória. Permitiam a instalação de uma placa especial de memória, mas infelizmente esses barramentos eram proprietários. Significa que uma placa de CPU com um barramento local proprietário para expansão de memória deveria obrigatoriamente usar uma placa de expansão de memória do mesmo fabricante. Como esses barramentos não eram padronizados, não foram usados em larga escala pela indústria de placas para PCs. A necessidade deste tipo de barramento cessou com a proliferação dos módulos de memória, que permitiam obter elevadas capacidades de memória em pouco espaço.

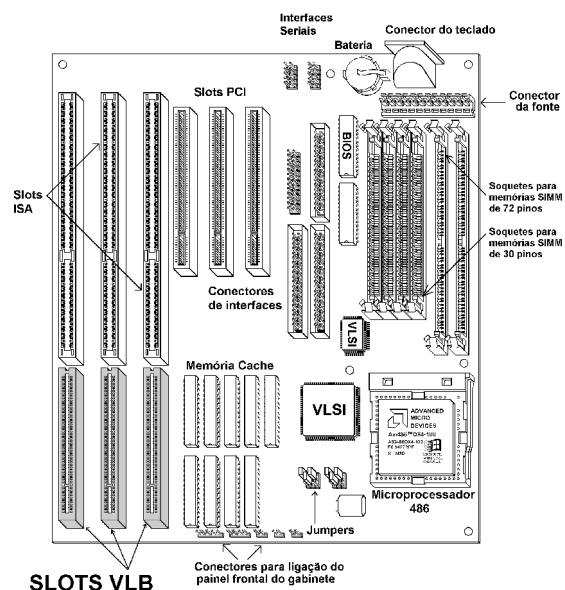
A necessidade de barramentos mais rápidos voltou a ser grande quando as placas de vídeo passaram a operar com altas resoluções e elevado número de cores. As antigas placas VGA de 16 bits operavam de forma satisfatória em modo texto, e com gráficos de 640x480 com 256 cores, quando toda a memória de vídeo ocupava apenas 300 kB. Já com a resolução de 1024x768 com 16 milhões de cores, a memória de vídeo ocupa cerca de 2 MB. Para transferir integralmente uma tela nesta resolução para uma placa de vídeo ISA, seria necessário um tempo de cerca de 0,25 a 0,5 segundo. A movimentação da tela seria extremamente lenta, o que criou a necessidade de um novo barramento mais veloz, próprio para a placa de vídeo.

Foi então que surgiu o VESA Local Bus (VLB), criado pela Video Electronics Standards Association. Este barramento era representado fisicamente por um conector adicional que ficava alinhado com os slots ISA. Neste barramento era feita a reprodução quase fiel dos sinais de dados, endereço e controle do processador 486.

**FIGURA 12.35**

Placas SVGA VLB e IDEPLUS VLB.

Entre 1994 e 1995 eram comuns as placas SVGA VLB e IDEPLUS VLB, mostradas na figura 35. Usando uma placa SVGA VLB era conseguido um desempenho gráfico bastante superior. A placa IDEPLUS VLB tinha como vantagem a maior taxa de transferência da interface IDE, já que suas outras interfaces (seriais, paralela, drives de disquetes e joystick) operavam com taxas de transferência bem menores. Também foram produzidas placas digitalizadoras de vídeo e controladoras SCSI no padrão VLB. Infelizmente o barramento VLB era totalmente baseado no barramento local do 486. A chegada dos processadores Pentium e suas placas de CPU equipadas com slots PCI, juntamente com a extinção dos processadores 486, fez com que o barramento VLB também caísse em desuso.

**FIGURA 12.36**

Uma placa de CPU com slots VLB.

Os slots VLB foram muito utilizados em placas de CPU 486 com clock externo de 33 MHz, portanto operavam também com 33 MHz. São slots de 32 bits, e a 33 MHz oferecem uma taxa de transferência teórica máxima de 133 MB/s. Esta taxa variava de acordo com o clock externo do processador. Por exemplo, o 486DX2-50 operava externamente a 25 MHz, portanto o barramento VLB acompanhava este clock, e a taxa de transferência resultante era de 100 MB/s. A tabela a seguir mostra alguns processadores e as taxas obtidas nos seus barramentos VLB.

Processador	Clock externo e do VLB	Taxa de transferência
486DX-33, 486DX2-66, 486DX4-100, 5x86-133	33 MHz	133 MB/s
486DX2-80	40 MHz	160 MB/s
486DX-25, 486DX2-50, 486DX4-75	25 MHz	100 MB/s

Note que a maioria das placas de expansão VLB não suportava operar acima de 33 MHz. Para suportar 40 MHz era preciso utilizar wait states que eram programados através de jumpers nas placas VLB.

O barramento VLB tinha várias desvantagens que contribuíram para que não fosse prolongado o seu uso depois da criação do PCI. As placas eram extremamente longas e maus contatos no conector eram bastante comuns.

Não oferecia o recurso Plug and Play, como ocorre com o barramento PCI. O fato de não ter sido criado sob a liderança da Intel e da Microsoft também contribuiu para que não fizesse tanto sucesso no mercado.

Barramento ISA

O barramento ISA (Industry Standard Architecture) surgiu no início dos anos 80. Foi criado pela IBM para ser utilizado no IBM PC XT (8 bits) e no IBM PC AT (16 bits). Apesar de ter sido lançado há muito tempo, podemos encontrar slots ISA em praticamente todos os PCs produzidos nos últimos anos. Apenas a partir do ano 2000 tornaram-se comuns novas placas de CPU que aboliram completamente os slots ISA.

No tempo em que não existiam barramentos mais avançados, as placas de CPU possuíam 6, 7 e até 8 slots ISA. Depois da popularização do barramento PCI, as placas de CPU passaram a apresentar apenas 2 ou 3 slots ISA. As raras placas produzidas atualmente que possuem slots ISA, apresentam apenas um ou dois desses slots.

Os slots ISA são utilizados por várias placas de expansão, entre as quais:

- Placas fax/modem
- Placas de som
- Placas de interface para scanner SCSI
- Interfaces proprietárias
- Placas de rede

Note que estamos falando principalmente de modelos antigos, pois a maioria dos fabricantes de placas de expansão já adotou definitivamente o padrão PCI, e não fabricam mais novos modelos ISA. De qualquer forma, a presença de slots ISA em uma placa de CPU é útil caso seja necessário aproveitar placas de expansão antigas.

As placas fax/modem e as placas de som foram as que mais demoraram para adotar o padrão PCI. O motivo desta demora é que o tráfego de dados que elas utilizam mal chega a ocupar 5% da capacidade de transferência de um slot ISA. Já as placas de vídeo, placas de rede, interfaces SCSI e digitalizadoras de vídeo operam com taxas de transferência mais elevadas, por isso foram as primeiras a serem produzidas no padrão PCI.



A figura 37 mostra exemplos de placas de expansão ISA. Observe que algumas delas utilizam um conector simples (8 bits), enquanto outras utilizam um conector duplo (16 bits). Da mesma forma, os slots ISA podem apresentar um único conector (ISA de 8 bits) ou dois conectores (ISA de 16 bits). Placas ISA de 8 bits podem ser encaixadas, tanto em slots ISA de 8 bits como em slots ISA de 16 bits. Placas ISA de 16 bits devem ser encaixadas obrigatoriamente em slots ISA de 16 bits (exceto em raríssimos casos de placas VGA antigas, de 16 bits, mas que se comportam como placas de 8 bits ao serem encaixadas em um slot de 8 bits). Os slots ISA de 8 bits eram encontrados em placas de CPU muito antigas.

Observe que Barramento ISA não é sinônimo de Slot ISA. O Barramento ISA é um conjunto de sinais digitais que partem do chipset e do processador, e atingem tanto as placas de expansão, através dos slots, como circuitos da placa de CPU. Por exemplo, as interfaces para drives de disquete, interfaces seriais e interface paralela embutidas na placa de CPU são controladas através do barramento ISA, apesar de não utilizarem os slots.

Sinais do barramento ISA

A seguir apresentamos os principais sinais do barramento ISA e suas descrições. A maioria deles foram originados no barramento de sistema dos processadores 8088 e 80286.

Sinal	Descrição
DATA0-DATA16	Barramento de dados, com 16 bits

Address 0 – Address 23	Barramento de endereços, com 24 bits
Reset Driver	Sinal que é enviado para todo o barramento quando o processador é resetado, provocando assim o RESET dos demais circuitos do computador.
IRQ3 – IRQ15	Linhas de requisição de interrupção. Através delas as diversas placas de expansão e demais circuitos podem interromper o processador. Algumas linhas não estão disponíveis no barramento ISA, pois são usadas internamente pela placa de CPU. É o caso da IRQ0, usada pelo alarme, IRQ1 usada pelo teclado, IRQ2 usada para conexão entre os dois controladores de interrupção e IRQ13 usada pela unidade de ponto flutuante.
DMAREQ 0 – DMAREQ7	Requisições de DMA. Através dessas linhas uma interface pode solicitar transferências por DMA. O canal DMA4 não está disponível no barramento, pois é usado para conectar os dois controladores de DMA.
DMAACK0 – DMAACK7	Sinais de reconhecimento de interrupção, um para cada canal de DMA.
Real Memory Read	Indica leituras na memória, quando o processador opera no modo real.
Real Memory Write	Indica escritas na memória, quando o processador opera no modo real.
Memory Read	Indica leituras na memória, tanto no modo real como no protegido.
Memory Write	Indica escritas na memória, tanto no modo real como no protegido.
I/O Read	Indica leitura em dispositivos de E/S
I/O Write	Indica escrita em dispositivos de E/S
Terminal Count	Indica que foi finalizada uma operação de DMA
ALE	Address Latch Enable. Indica ao barramento que os endereços são válidos e podem ser capturados pelas interfaces.
OSC	Sinal de 14,38 MHz, usado pelas placas de vídeo antigas para gerar o sinal de vídeo composto no padrão NTSC.
Memory 16-bit Select	Indica que o atual ciclo de memória deve operar com 16 bits.
I/O 16-bit Select	Indica que o atual ciclo de E/S deve operar com 16 bits.
I/O Channel Check	Indica que ocorreu um erro de paridade na memória.
I/O Channel Ready	Usado para introduzir Wait States nos ciclos em andamento.

LPC - substituto do ISA em placas modernas

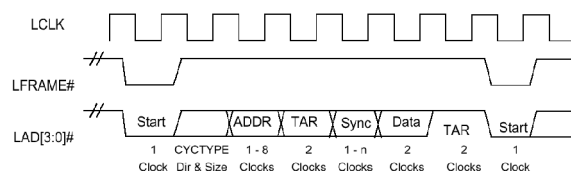
As placas de CPU modernas não utilizam mais slots ISA. Ainda assim possuem circuitos internos que precisam estar disponíveis, apesar de serem originalmente ligados ao barramento ISA. São as interfaces seriais, a interface paralela, a interface para drives de disquetes, o CMOS, a interface de teclado, a interface para mouse, o PC Speaker e o BIOS. Note que nas placas modernas, com exceção do BIOS, todos esses circuitos fazem parte do chip conhecido como Super I/O. Seria preciso manter um barramento ISA interno apenas para a ligação desses dispositivos.

Outro problema é que o barramento ISA possui muitos pinos. Os slots de 16 bits têm ao todo 98 pinos. Mesmo descontando pinos que normalmente não

O LPC tem ao todo 13 pinos, sendo 7 de uso obrigatório e 6 opcionais, usados apenas quando é preciso utilizar todos os seus recursos, o que nem sempre é o caso. Os pinos de uso obrigatório são:

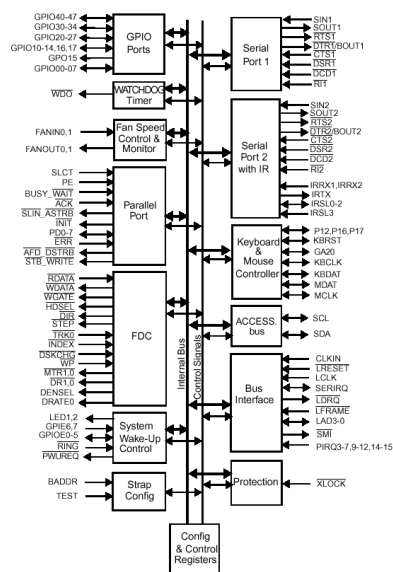
Sinal	Descrição
LAD0- LAD3	Barramento serial para comandos, endereços e dados, todos multiplexados.
LFRAME	Indica o início de cada ciclo.
LRESET	Reseta os dispositivos do barramento.
LCLK	Clock de 33 MHz para os dispositivos do barramento.

Os demais 6 sinais são necessários para dispositivos que usam DMA, interrupções e gerenciamento de energia. As operações de leitura e escrita em memória e E/S no barramento LPC são feitas em 8 bits, assim como ocorre com todas as interfaces que pretende atingir. Pode ainda realizar operações de DMA com 8, 16 e 32 bits, e operações de Bus Mastering com 8, 16 e 32 bits.

**FIGURA 12.39**

Operação do barramento LPC.

A figura 39 mostra o diagrama de tempo de uma transferência no barramento LPC. As transferências são comandadas pelo sinal de clock LCLK. O sinal LFRAME é ativado em nível 0, indicando o início de um ciclo. A seguir os comandos, endereços, dados e demais informações são transmitidas através de LAD0-LAD3, em seqüências de 4 bits a cada clock.



*** 35% ***

FIGURA 12.40

Diagrama de um chip Super I/O para barramento LPC.

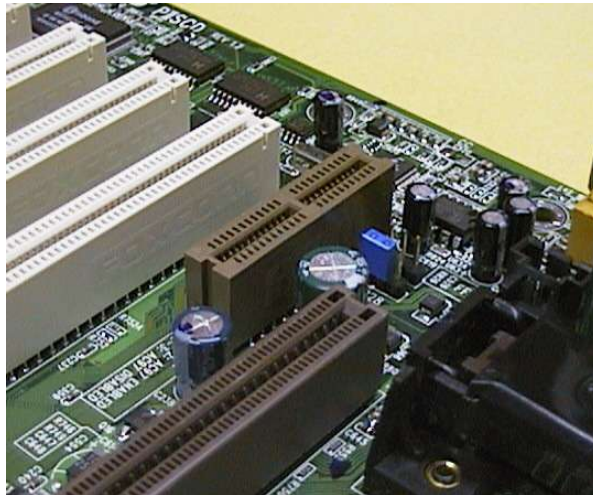
A maioria dos fabricantes de chips Super I/O produzem modelos para o padrão LPC. A figura 40 mostra o diagrama do chip PC87360, produzido pela National Semiconductor. Possui diversas interfaces, e no seu diagrama podemos ver os sinais de conexão com o barramento LPC (Bus Interface). Note que existem ainda disponíveis os sinais PIRQ3-PIRQ7, PIRQ9-PIRQ12, PIRQ14 e PIRQ15. Partindo do Super I/O, esses sinais são enviados ao South Bridge, o chip no qual estão integrados os controladores de interrupções. Através dele as interfaces existentes no Super I/O podem interromper o processador.

Barramentos AMR, CNR e ACR

Muitas placas de CPU modernas possuem conectores para a instalação de um riser card. São os slots AMR, CNR e ACR. Um riser card é uma placa de interface especial, cujo principal objetivo é a redução de custo. A idéia básica dessas placas é dividir cada interface em duas partes. Uma parte, totalmente digital e de baixo custo, é embutida no chipset. A outra parte, mais voltada para funções analógicas, fica no riser card. A comunicação entre o chipset da placa de CPU e o Riser Card é feita em um formato serial, utilizando um reduzido número de pinos.

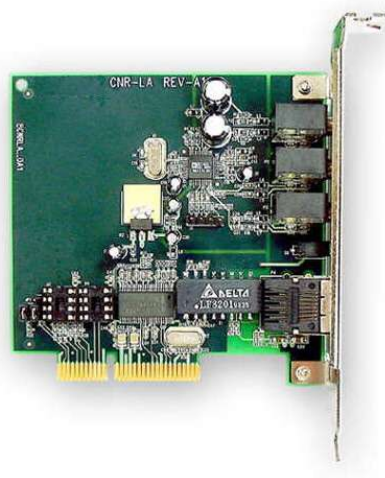
O primeiro padrão de riser card foi o AMR (Audio Modem Riser). Destinava-se a ser usado apenas com circuitos de som e modem. Para utilizar

essas placas é preciso ter no chipset, os circuitos de áudio AC'97 e de modem MC'97. Muitos chipsets modernos possuem tais circuitos. Os circuitos de som AC'97 são relativamente simples, mas com boa qualidade. Os circuitos MC'97 são similares aos existentes nos soft modems. Toda a parte digital desses dispositivos fica localizada no chipset, e a parte analógica fica em uma placa de expansão AMR, que deve ser instalada no slot apropriado. A figura 41 mostra um slot AMR.

**FIGURA 12.41**

Slot AMR em uma placa de CPU.

Depois do AMR, a Intel criou um novo padrão, o CNR (Communications Network Riser). O tipo de slot é idêntico ao usado pelo padrão AMR. Neste slot podemos instalar riser cards com funções de áudio, modem e rede. As placas AMR e CNR têm formatos semelhantes, como a que vemos na figura 42.



*** 35% ***

FIGURA 12.42

Uma placa de Rede/Áudio CNR.

Portanto o CNR é um padrão similar ao AMR, porém suporta funções de rede. Existe uma diferença entre as localizações dos slots AMR e CNR. Normalmente as placas de CPU possuem um ou outro tipo, mas não ambas. O slot à esquerda de todos os slots PCI é o CNR (olhando a placa de CPU pela frente do gabinete). O slot localizado à direita é um AMR.

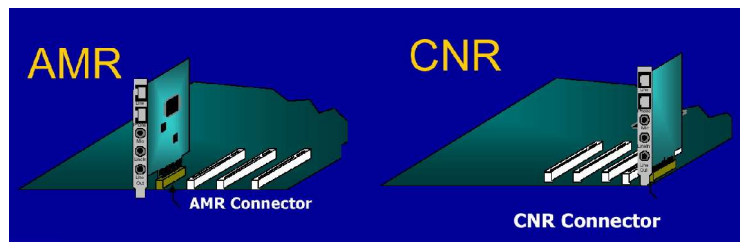


FIGURA 12.43

Localização dos slots AMR e CNR.

Podemos então considerar que usar uma placa AMR ou CNR é o mesmo que utilizar uma placa de som simples, ou um soft modem, ou uma interface de rede comum. A diferença é que parte dos circuitos ficam no chipset (SouthBridge e Super I/O) e parte fica no riser card. Existem vários tipos de riser card no mercado: modem, áudio, áudio+modem, áudio+rede, modem+rede, modem+áudio+USB, etc.

O padrão ACR, promovido pela AMD e outros fabricantes de modems e produtos de comunicação, é compatível com o AMR, e também oferece funções de rede, USB e comunicação em banda larga. Seu slot possui mais pinos, e é similar ao slot PCI, porém com uma fixação mecânica diferente.

Na figura 44 vemos uma placa ACR. É bem parecida com uma placa PCI, entretanto não pode ser encaixada em um slot PCI. Note que o chanfro existente no conector da placa fica na posição simétrica em relação à dos slots PCI de 5 volts comuns nas placas de CPU. O conector ACR existente na placa de CPU é do mesmo tipo usado pelos slots PCI, mas além da posição ser invertida, o conector é ligeiramente deslocado para a parte traseira do gabinete, o que impede o encaixe de placas ACR em slots PCI, e vice-versa. A localização do conector ACR na placa de CPU é a mesma do conector CNR, ou seja, à esquerda dos slots PCI.

**FIGURA 12.44**

Uma placa ACR.

A idéia básica do ACR é a mesma do AMR e do CNR: produzir interfaces simples, com a parte digital localizada no chipset e a parte analógica localizada no riser card.

Muitos fabricantes estão produzindo riser card dos tipos AMR/CNR e ACR. Diversos chipsets da Intel, VIA e SiS estão embutindo circuitos como o áudio AC'97 e modem MC'97, restando apenas instalar o riser card apropriado para ter acesso a esses recursos. As interfaces USB também já estão presentes nos chipsets, faltando apenas rotar seus dados para o riser card. A tendência é que os chipsets passem a utilizar embutidos também os circuitos de rede e de comunicação em geral. Mesmo quando os circuitos não estão embutidos no chipset, os barramentos AMR, CNR e ACR podem ser usados, pois existem diversos chips independentes, de baixo custo, que podem ser utilizados pelos fabricantes de placas de CPU.

Note ainda que a maioria das placas de CPU com som onboard, utilizam os circuitos de áudio AC'97. Ao invés de utilizarem um riser card, os fabricantes acrescentam na própria placa de CPU os circuitos que estariam no riser card de áudio, e usam os tradicionais conectores de áudio na parte traseira da placa de CPU. Desta forma o áudio AC'97 pode ser utilizado, sem que seja preciso instalar um riser card.

Ainda é muito difícil encontrar riser cards no comércio, porém seus fabricantes apostam que nos próximos anos serão as opções mais comuns para soft modems e outras interfaces de baixo custo.

Velocidades dos principais barramentos

O desempenho de uma placa conectada a um barramento depende de vários fatores, entre os quais, a taxa de transferência. Esta por sua vez, depende do número de bits, do clock e do número de transferências feitas a cada ciclo. A tabela que se segue mostra as características dos barramentos ISA, PCI e AGP.

Barramento	Bits	Clock	Transferências por ciclo	Taxa de transferência
ISA	16	8 MHz	1/2	8 MB/s
PCI 33 MHz 32 bits	32	33 MHz	1	133 MB/s
PCI 33 MHz 64 bits	64	33 MHz	1	266 MB/s
PCI 66 MHz 32 bits	32	66 MHz	1	266 MB/s
PCI 66 MHz 64 bits	64	66 MHz	1	533 MB/s
AGP 1x	32	66 MHz	1	266 MB/s
AGP 2x	32	66 MHz	2	533 MB/s
AGP 4x	32	66 MHz	4	1066 MB/s
AGP 8x	32	66 MHz	8	2133 MB/s

O barramento ISA utiliza um clock de 8 MHz, e realiza transferências de 8 ou 16 bits. Usando 16 bits, teoricamente poderia transferir 16 MB/s (8 MHz x 2 bytes), mas cada transferência utiliza 2 ciclos de clock, como era exigido pelas placas de expansão do início dos anos 80, que eram muito lentas. Portanto realiza em média, meia transferência a cada ciclo. Desta forma, a taxa de transferência obtida com o ISA é de apenas 8 MB/s.

O barramento PCI mais simples utiliza um clock de no máximo 33 MHz, com transferências de 32 bits. Isto resulta em uma taxa de transferência igual a 132 MB/s (33 MHz x 4 bytes). As versões de 64 bits e 66 MHz resultam em taxas mais elevadas, chegando até 533 MB/s.

O barramento AGP não está ligado ao PCI, e sim, ao barramento externo do processador, apesar de ter muitas características similares às do PCI. No chamado modo AGP 1x, em cada ciclo AGP é feita uma transferência, resultando em uma taxa de 266 MB/s. Como já mostramos, os modos AGP 2x, AGP 4x e AGP 8x fornecem 533 MB/s, 1066 MB/s e 2133 MB/s, respectivamente.

Essas comparações mostram como uma placa de vídeo PCI simples opera com taxa de transferência mais lenta (133 MB/s) que um modelo AGP. Muitos modelos de placas de vídeo são atualmente produzidos nas versões AGP e PCI, sendo que as versões PCI destinam-se a upgrades, ou seja, melhorar o sistema de vídeo de PCs antigos. Com o passar do tempo, serão cada vez mais raras as placas de vídeo PCI.

Note que todas essas taxas de transferências são limites máximos teóricos, que na prática não são atingidos. Devemos levar em conta as latências dos barramentos, que necessitam do fornecimento inicial de endereço e comando, uso de wait states e outros eventos que causam overhead, resultando em taxas de transferência efetivas bem menores. De qualquer forma, os limites máximos teóricos mostrados na tabela servem como comparação do potencial de cada um desses barramentos.

//////// FIM //////////