Capítulo 6

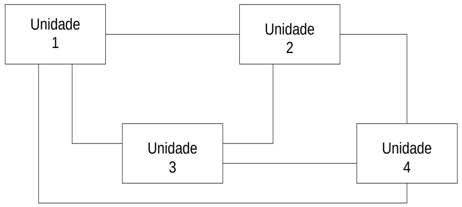
Barramento

6.1 Introdução

Existem basicamente duas formas para conectar as unidades de um sistema de computador (UCP, Unidade de Memória e a Unidade de E/S)

- ligação dedicada e
- barramento.

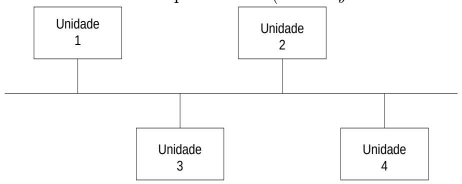
A ligação dedicada estabelece uma ligação única entre duas unidades. A principal vantagem deste tipo de conexão em comparação com ligações por barramento é que a velocidade de transferência de informação entre as duas unidades pode ser maior, uma vez que no projeto de interface as características específicas de cada uma das unidades são consideradas individualmente. Por outro lado, é difícil adicionar uma nova unidade ao sistema, pois para cada nova unidade é necessário um novo circuito de interface e novo conjunto de fios. A ligação dedicada é utilizada principalmente para a ligação de computadores com periféricos de E/S, podendo ser em formato serial ou em formato paralelo. Exemplos de ligação dedicada que já vimos no capítulo 5 são a ligação serial padrão RS232 e ligação paralela padrão Centronics.



A ligação por barramento utiliza um único conjunto de fios para interligar diversas unidades de um sistema de computadores. Embora não seja nada mais que um conjunto de fios projetados para interligar diversas unidades de um sistema de microprocessador, um barramento tem um papel importante tanto no aspecto econômico (evitar ligações dedicadas, ponto-a-ponto) quanto no aspecto funcional (a adição de uma nova unidade pode ser feita de forma bastante simples, basta *plugar* a nova unidade no barramento), desde que cada unidade do sistema obedeça rigorosamente às características de interfaceamento com ele. Como as unidades podem ser desenvolvidas por equipes distintas de empresas diferentes, impõe-se um rigor muito grande no projeto de um barramento, devendo-se levar em consideração as seguintes características (Fig. 10.2 do livro-texto, p. 764):

1. características mecânicas: dimensões dos barramentos (número de fios), dimensões das placas de interface, dimensões dos conectores e especificação do material e da resistência mecânica deste material.
2. características elétricas: níveis de tensão, de corrente, imunidade de ruído, capacidade de carga e a taxa de transferência de dados.
3. características funcionais: especificação das funções de sinais em cada fio/pino.
4. características procedurais: estabelecem o relacionamento entre os sinais e a sequência que eles devem ser ativados/desativados para executar uma dada operação.

Para uniformizar essas caracterizações, existem comissões de representantes de indústrias em entidades, como o EIA (*Electronic Industries Association*), o IEC (*International Electrotechnical Commission*) e o IEEE (*Institute of Electrical and Electronics Engineers*, para definir padrões de barramentos com o objetivo de orientar os fabricantes. Como exemplo de barramentos que surgiram através de comissões de fabricantes temos o padrão de barramento VME (IEC Com. 47b -¿ std. 821) e o padrão FUTURE BUS (projeto IEEE 896). A maior parte dos barramentos no mercada surgiram como barramentos próprios de equipamentos específicos, mas pela alta escala de aplicação eles se transformaramm em padrões *de facto*. Como exemplo disto temos o barramento S100, que depois de ser muito utilizado na indústria de microprocessadores de 8 bits passou por comissões do IEEE e se transformou no barramento padrão IEEE-696. Outro exemplo é o braamento MULTIBUS lançado pela INTEL Co. que se transformou na norma IEEE 796. O exemplo mais conhecido é o barramento do IBM PC, o IOCHANNEL, que foi adotado por diversos fabricantes de placas e de sistemas e padronizado por um conjunto de fabricantes como o padrão ISA (*Industry Standard Architecture*).



Vimos ainda no capítulo 1 que quando várias unidades de um sistema computacional são interligadas por um barramento, distinguem-se em cada transferência, sob o ponto de vista do controle de barramento, duas classes de unidades:

Mestre de barramento: a unidade que controla a transferência; e

Escravos de barramento: são unidades que participam da transferência sob o controle de uma unidade-mestre.

Note-se que em cada transferência só pode existir um mestre de barramento. A escolha de um mestre de barramento dentre as potenciais unidades controladoras do barramento é conhecida por **arbitragem**.

Neste capítulo dedicamos ao estudo das características de estruturas de barramento e introduzimos a descrição dos barramentos VMEBus e NuBus. Mais especificamente, veremos:

- em que consistem as especificações mecânicas de um barramento,
- em que consistem as especificações elétricas de um barramento,

- como os acionadores/receptores podem afetar o comportamento de um sistema de barramento,
- como as terminações de linha podem afetar o comportamento de um sistema de barramento,
- como as unidades (cargas) conectadas ao barramento podem afetar o comportamento de um sistema de barramento,
- como um circuito de arbitragem gerencia o controle do barramento entre as potenciais unidades-mestres,
- quais são as políticas de arbitragem,
- como é o controle de transferência de dados no barramento,
- qual é a finalidade dos padrões de barramento e
- como são as especificações de dois padrões de barramento.

6.2 Características Mecânicas

As características mecânicas são determinantes no custo do projeto de um barramento. Entende-se como características mecânicas não só as dimensões das trilhas de interconexão como também as características mecânicas das conexões entre as unidades de uma sistema computacional e o barramento.

As conexões utilizadas nos barramentos do IBM PC é do tipo placa-conector (*edge-connector*), isto é as placas de interface (*daughterboard*) são inseridas diretamente nos soquetes disponíveis nas placas-mãe (*motherboard*), como ilustra a Fig. 10.3 do livro-texto (p. 765). Note que o espaçamento entre os contatos, o mecanismo de encaixe e o material dos contatos devem ser especificados detalhadamente.

Um outro tipo de conexão bastante popular é através de conectores de duas partes (*two-piece connector*). Estes conectores são menos suscpetíveis aos desgastes nos contatos e apresentam melhores contatos elétricos, porém são mais caros que os conectores PC. Fig. 10.4 do livro-texto (p. 766) apresenta um destes tipos de conectores – o conector definido pela norma DIN 41612.

6.3 Características Elétricas

Sob o ponto de vista elétrico, uma linha do barramento interliga o pino de saída de uma unidade com os pinos de entrada de outras unidades de um sistema computacional, permitindo que um sinal digital gerado por uma unidade seja distribuído entre várias outras com um nível lógico bem definido. Dependendo da frequência dos sinais, da carga que as unidades ligadas ao barramento representam e do comprimento das trilhas de barramento, um sinal gerado pelo acionador pode sobrepor com a sua reflexão nas "emendas" (*stub*) e nos extremos das linhas e chegar distorcido no receptor. Este efeito de reflexão pode, porém, ser evitado, se as linhas forem apropriadamente **terminadas** com resistores.

Portanto, ao caracterizar eletricamente um barramento, deve-se definir o número máximo de unidades que ele suporta, o número máximo de cargas que cada unidade representa, o nível de tensão destas unidades, etc.

Para contornar o problema da diversidade das unidades que se conectam a um barramento, procura-se "desacoplar" as caracterísitcas/comportamento do barramento e as propriedades elétricas de cada unidade através dos **acionadores** e **receptores** de barramento (Fig. 10.9 do livro-texto, p. 772). Estes *buffers* normalmente representam uma pequena carga para o barramento e apresentam uma grande capacidade de suprir/drenar altas correntes, como exemplifica a Tabela 10.3 do livro-texto (p. 772). O preço que se paga por este tipo de isolamento elétrico são adicionais atrasos, podendo levar ao problema de **contenção dinâmica**, como vimos nas pp. 244 –251 do livro-texto.

6.3.1 Configurações de Saída dos Acionadores

Quando mais de uma unidade é capaz de definir o nível lógico em uma linha de barramento, cuidados especiais devem ser tomados para evitar ligações que levem a contenção de barramento, como ilustra Fig. 10.10 do livro-texto (p. 773). Conforme a configuração de saída dos acionadores, distinguem-se duas classes de acionadores:

- passivos: tem como configuração de saída coletor (ou dreno) aberto (Fig. 10.12 do livro-texto, p. 776) e podem colocar uma linha de barramento para o nível lógico 0. O nível lógico 1 é definido por uma fonte externa (V_{cc}). O resistor delimitador de corrente (resistor *pull-up* para V_{cc}) deve ter (1) um valor tal que ele garanta o nível lógico 1 no barramento

$$R_{max} = \frac{V_{cc} - V_{OH}}{mI_{fuga} + nI_{IH}},$$

onde m e n são, respectivamente, as quantidades de acionadores e receptores, e I_{fuga} é a corrente que flui para dentro dos acionadores; (2) uma dissipação dentro da tolerância no nível lógico 0

$$R_{min} = \frac{V_{cc} - V_{OL}}{I_{OL} - nI_{IL}}.$$

Os acionadores passivos desempenham a função *wired-OR*, ou seja, se um dos acionadores drenar a corrente para o terra, a linha inteira fica no nível lógico 0. Este tipo de configuração é muito utilizado no circuito de arbitragem (Seção 6.3.4).

- ativos: tem como configuração de saída três-estados (Fig. 10.15 do livro-texto, p. 780) ou totem-pole (Figura 10.10 do livro-texto, p. 773) e definem ambos os níveis lógicos (0 e 1) no barramento. Portanto, diferentemente dos acionadores passivos, somente um acionador ativo pode acessar o barramento em cada instante para evitar o fenômeno de contenção.

Observação:

- Os dispositivos que integram os circuitos de acionadores e receptores são denominados *transceivers*, como 74LS242, 74LS243 e 74LS245.

Exercícios de Revisão

1. Resolver os exercícios 1, 2, 3, 4, 5, 6 e 9 do capítulo 10 do livro-texto.

6.3.2 Modelos de Linhas de Barramento

As transições dos estados de um sinal digital podem ser distorcidas em uma linha/trilha do barramento de acordo com as propriedades elétricas (resistência, capacitância e indutância) desta. Ao atingir um extremo do barramento, o sinal pode ainda ser refletido e interferir o sinal original. Estes fenômenos tendem a ser críticos à medida que se aumenta a frequência dos sinais e o comprimento das linhas. Faz-se então necessário o estudo dos efeitos de uma linha de transmissão nos sinais digitais de diferentes frequências.

Existem básicamente três modelos para analisar o comportamento de um sinal em uma linha:

- modelo resistivo é um modelo ideal de uma linha, considerando que o sinal seja contínuo (Fig. 10.18 do livro-texto, p. 784). Neste modelo, a capacitância e a indutância da linha são nulas.

- modelo resistor-capacitor RC é também conhecido como um circuito passa-baixo, isto é sinais de baixa frequência são pouco atenuados/distorcidos (Fig. 10.19 do livro-texto, p. 785). A **frequência de corte** para a qual o sinal é atenuado de 3 dB é dado por $f_c = \frac{1}{2\pi RC}$ e o tempo de subida (*rise time*) por $t_r = \frac{2.2 f_c}{2\pi}$. Este modelo é suficiente para analisar linhas de pequeno comprimento.

- modelo de linha de transmissão é recomendado para ser utilizado em sistemas onde o tempo de subida é muito menor do que o tempo de propagação (Fig. 10.20 do livro-texto, p. 785). A proporção recomendada é de 2 a 4 vezes menor.

De acordo com o modelo simplificado de linha de transmissão, o atraso de propagação por unidade de comprimento pode ser expresso por

$$\sqrt{LC},$$

onde L e C são indutância e capacitância distribuídas por unidade de comprimento. Pela Lei de Ohm, a corrente que flui pela linha depende da impedância (caracterísitca) da linha que pode ser estimada através de

$$Z_o = \sqrt{\frac{L}{C}}$$

Rigorosamente, a impedância característica de uma linha, Z_o, é determinada pela geometria do barramento e pelas propriedades do dielétrico que separa as linhas e o retorno do terra. Tabela 10.6 do livro-texto (p. 786) mostra que a impedância característica das trilhas em uma placa de PC (*backplane*) é aproximadamente 100 Ω.

Um pulso que se propaga numa linha é refletido ao chegar no seu extremo. O sinal refletido pode distorcer significativamente o sinal original ao sobrepô-lo. Uma solução para isso é introduzir nos extremos cargas R_T capazes de dissiparem o sinal "remanescente", procurando anular o efeito do sinal refletido. Com uso da Lei de Ohms, pode-se derivar, de forma bastante grosseira, a tensão refletida V_r em termos da tensão incidente V_i, R_T e Z_o

$$V_r = V_i \frac{R_T - Z_o}{R_T + Z_o}, \tag{6.1}$$

Pela Eq.(6.1) é fácil ver que se $R_T = Z_o$, $V_r = 0$. Vale ressaltar aqui que nos sistema computacionais reais, as formas de onda numa trilha de barramento são muito mais complexas, devido à natureza não-linear dos dispositivos semicondutores e as "emendas" nos pontos de inserção das unidades ao barramento (*stubs*). Portanto, a determinação exata da impedância característica de uma linha não é uma tarefa trivial. Na prática, uma resistência de aproximadamente 100 Ω consegue atenuar bastante os efeitos de reflexão.

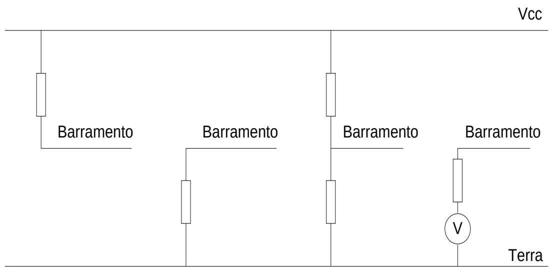
Quanto à configuração de terminação de linhas, pode-se

1. conectar o resistor entre V_{cc} e o barramento: esta configuração reduz a imunidade de ruído do nível lógico 0;

2. conectar o resistor entre o terra e o barramento: esta configuração reduz a imunidade de ruído do nível lógico 1;

3. conectar dois resistores, um entre o terra e o barramento e o outro entre V_{cc} e o barramento: é também conhecida como terminação paralela e tem a vantagem de não comprometer a imunidade de ruído da linha (Fig. 10.23 do livro-texto, p. 791);

4. conectar o resistor entre o barrmento e um regulador de tensão de valor aproximadamente igual à metade do V_{cc} e o terra: é também conhecida como terminação ativa e tem a vantagem da terminação não afetar o potencial da linha (Fig. 10.24 do livro-texto, p. 791).



Nos sistemas reais as cargas não são puramente resisitivas. As capacitâncias introduzidas pela unidades conectadas ao barramento podem alterar o tempo de propagação e a impedância característica (sem carga), Z_o, da linha. Isso significa que para determinar o valor de uma terminação deve-se fazer uso da impedância característica com carga, ao invés da impedância característica sem carga. Esta impedância com carga pode ser expressa em termos de Z_o

$$Z'_o = \frac{Z_o}{\sqrt{1 + \frac{C_d}{C_o}}},$$

onde C_d á capacacitânica introduziada pela carga, normalmente especificada pelos fabricantes e C_o é a capacitância intrínseca da linha que pode ser obtida por

$$C_o = \frac{t_{pd}}{Z_o}.$$

Neste caso, o tempo (de atraso) de propagação sem carga, t_{pd}, pode ser medido no laboratório. Este tempo pode também ser aumentado em consequência das cargas na linha, ou seja,

$$t'_{pd} = t_{pd}\sqrt{1 + \frac{C_d}{C_o}}.$$

Como é difícil prever o tipo de carga a ser conectada em uma linha de barramento, recomenda-se utilizar dispositivos que apresentam baixa carga capacitiva.

6.3.3 Inserção de placas ao vivo

Uma característica elétrica desejada de um barramento é suportar inserção ao vivo de uma nova unidade (placa), isto é o nível lógico no barramento não

seja afetado pelo transiente (na ordem de milisegundos) que aparece durante a conexão da unidade. Esta propriedade é útil para sistemas que precisam operar continuamente, como nos casos de controladores e servidores de rede.

Muitas vezes, o acesso difícil dos dispositivos dificulta a previsão da sequência de contato dos pinos da placa com os pinos do sistema e o instante em que estes contatos ocorrem. Portanto, é difícil prever como o nível lógico do barramento é afetado. O que conseguiu mostrar experimentalmente é que a remoção de uma placa ao vivo causa menos erros que a sua inserção.

Fig. 10.26 do livro-texto (p. 794) apresenta um modelo elétrico de inserção, onde a nova unidade a ser conectada é representada por uma capacitância. Através deste modelo, é fácil concluir que dispositivos de baixa capacitância introduz um transiente de largura menor. Alguns fabricantes fornecem acionadores apropriados para inserção ao vivo. Fig. 10.27 do livro-texto (p. 795) ilustra a configuração de saída destes acionadores. Note-se que ela é uma variante da de três-estados, onde o sinal de controle é a saída de uma porta lógica NOR com duas entradas. Uma entrada é o convencional sinal de habilitação e o outro é derivado de um circuito que assegura que o pino de saída só é habilitado quando a tensão V_{cc} da nova unidade atingir o seu nível operacional.

6.3.4 Circuito de arbitragem

O árbitro de um barramento recebe as requisições dos potenciais mestres de barramento e concede o uso a um deles segundo alguma estratégia pré-estabelecida. Uma política de arbitragem é chamada adequada (*fair*), quando todos os solicitantes tem igual chance para ganhar o acesso ao barramento. As políticas que dão preferência aos solicitantes de maior prioridade podem ser, por exemplo, inadequadas, no sentido de que o de baixa prioridade pode nunca ser atendido.

O projeto de um circuito de arbitragem de barramento pode ser complexo, uma vez que ele pode receber vários sinais de solicitação assincronamente e deve tomar uma decisão adequada perante estes sinais, mesmo que estes mudem de estado no intervalo de decisão (fenômeno conhecido por metaestabilidade). Mais ainda, ele deve evitar contenção do barramento permitindo somente o acesso ao mestre (de barramento) escolhido.

Fig. 10.28 do livro-texto (p. 796) ilustra as características básicas de um circuito de arbitragem. Neste caso particular, o circuito tem a função de controlar o acesso de uma unidade de memória (*dual-ported memory* – DPRAM) por dois processadores: *Processor 1* e *Processor 2*. Em outras palavras, o recurso (memória) é compartilhado por dois processadores e o conjunto de instruções (de um programa) que utiliza este recurso define uma **região crítica** cujo acesso é controlado por variáveis denominadas **semáforos**. Estes semáforos tem também o seu acesso compartilhado.

Para evitar problemas de sincronização e *race conditions*, o conteúdo dos semáforos deve ser lido e atualizado em um único ciclo de instrução. Particularmente, MC68000 provê a instrução TAS (*test and set*) com tal função. O diagrama de tempo é esquematizado na Fig. 10.29 do livro-texto (p. 798). Note que os sinais UDS*/LDS* são ativados duas vezes no ciclo com AS* ativo, a primeira é para a leitura e a segunda é para a escrita. Ainda mais, a segunda ativação está condicionada a ativação do sinal DTACK*. O circuito de arbitragem apresentado na Fig. 10.30 do livro-texto (p. 800) provê os necessários sinais DTACK*. Fig. 10.33 do livro-texto (p. 803) mostra os detalhes desta parte do circuito.

O acesso mutuamente exclusivo do barramento é garantido por um conjunto de flip-flops (Fig. 10.31 do livro-texto, p. 801). No caso, cada unidade processadora é conectada ao circuito de arbitragem por duas linhas: uma linha de requisição/solicitação ($REQUEST$) e uma linha de habilitação ($ENABLE$) (Seção 6.3.4). Observe que a frequência de amostragem do sinal de requisição é determinada por *arbitration clock*.

Basicamente distinguem-se duas políticas de arbitragem: centralizada e distribuída.

Árbitro Centralizado

Quando o *hardware* usado para passar o controle de uma unidade para outra estiver "altamente" concentrado num único local. Este local pode ser uma outra unidade ligada ao barramento.

O árbitro centralizado pode ser por

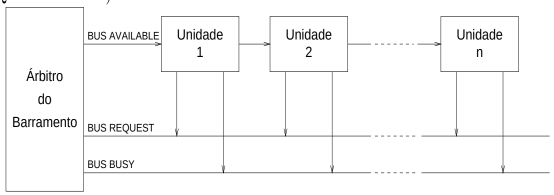
- *daisy chaining*
- *polling*
- requisições independentes

Árbitro Centralizado por *Daisy Chaining*

As n potenciais unidades-mestres são ligadas em cadeia (*daisy chain*) ao barramento onde se destaca um circuito de arbitragem – Árbitro do Barramento. Este circuito reúne a maior parte da lógica do barramento e utiliza dois sinais de entrada

- BUS BUSY, que indica que o barramento está sendo controlado por uma unidade-mestre, e
- BUS REQUEST, que é usado pelas potenciais unidades-mestres para solicitar o controle do barramento.

e um sinal de saída, BUS AVAILABLE, para indicar a disponibilidade do barramento. Este sinal é gerado se o barramento não estiver sendo ocupado (BUS BUSY inativo) e houver uma solicitação do barramento (BUS REQUEST ativo).



O sinal BUS AVAILABLE é propagado de uma unidade para a outra se a unidade precedente não estiver solicitando o barramento. Quando uma unidade estiver solicitando o barramento e recebe o sinal BUS AVAILABLE, ele assume o controle do barramento forçando o sinal BUS BUSY para o seu nível ativo e deve retirar o seu sinal BUS REQUEST.

A principal vantagem deste esquema é a sua simplicidade, um número muito pequeno de sinais é necessário e é independente do número de unidades-mestres.

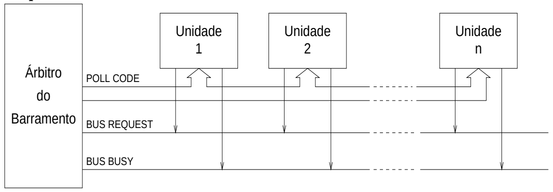
Entre as desvantagens citamos

- susceptibilidade a falhas. Se uma falha ocorrer numa lógica de propagação do sinal BUS AVAILABLE pode levar a que este sinal não atinja as demais unidades da cadeia ou, num pior caso, pode levar a que duas unidades tenham o controle do barramento.

- prioridade fixa por posição. Se a unidade mais próxima fizer solicitações muito frequentes, as unidades mais afastadas "nunca"conseguirão ganhar o controle do barramento.
- velocidade de arbitragem mais lenta, dependente da configuração.
- facilidade na ruptura da cadeia. Cuidado deve ser tomado ao se retirar uma unidade para evitar a ruptura da cadeia.

Árbitro Centralizado por *Polling*

Neste esquema a potencial unidade-mestre que requer o barramento força o sinal BUS REQUEST para o seu nível ativo. Recebendo este sinal o árbitro do barramento inicia uma contagem de *poll counter*. Todas as unidades que estiverem solicitando o barraamento comparam o valor nestas linhas com o seu código de *poll*. Quando nestas linhas estiver um código que coincida com uma das unidades solicitantes, esta força a linha de BUS BUSY indicando que ocupou o barramento.

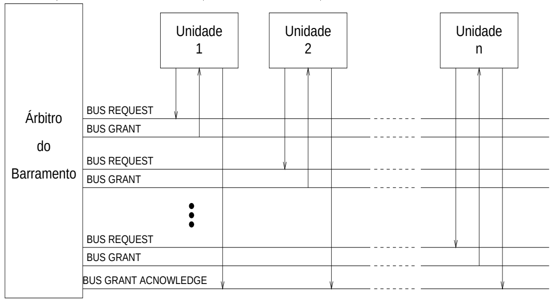


Este esquema evita os problemas de prioridade fixa por posição e permite um total controle centralizado do estado de ocupação do barramento. A forma de contar pode ser feita de diversas maneiras, cada uma estabelece um tipo de prioridade, por exemplo a contagem crescente e decrescente. Se ele for zerado, a prioridade é estabelecida de acordo com o código de *poll* das unidades, sendo o código zero o mais prioritário. Se a contagem continuar a partir do código da unidade atendida, é dada igual oportunidade de ocupação do barramento às diversas unidades independentemente do seu código. Essa forma caracteriza o que se chama prioridade por *round robin*.

A principal desvantagem deste esquema é a limitação do número de unidades que podem ser adicionadas ao barramento.

Árbitro Centralizado por Requisições Independentes

Este esquema utiliza um par de linhas para cada potencial unidade-mestre do barramento, uma linha de solicitação (BUS REQUEST) e uma linha de habilitação/reconhecimento (BUS GRANT).



Quando uma unidade precisar do barramnto ele envia ao árbitro do barramento o seu sinal de BUS REQUEST, o árbitro seleciona a próxima unidade a ter sua solicitação atendida e lhe envia o sinal de BUS GRANT. A unidade atendida retira a sua solicitação e ativa o sinal BUS GRANT ACKNOWLEDGE para avisar às demais unidades que o barramento está ocupado. Após completar a sua transferência a unidade desativa o sinal BUS GRANT ACNOWLEDGE e árbitro remove o sinal BUS GRANT e seleciona a próxima unidade que requer o barramento.

Como todas as solicitações estarão presente em paralelo no árbitro do barramento, a resposta a uma solicitação é mais rápida que nos dois esquemas anteriores. No árbitro está localizada toda a política de alocação do barramento, podendo ser implementado um esquema de prioridade pré-especificada, prioridade adaptativa ou mesmo um esquema de *round robin*.

A maior desvantagem deste tipo de arbitragem é o número limitado de dispositivos que podem ser mestres de barramento. Este número é função do número de pares de linhas de BUS REQUEST e BUS GRANT. Outra desvantagem é a maior complexidade do circuito de árbitro.

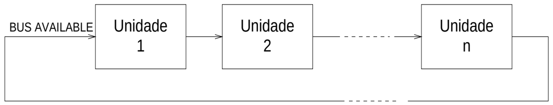
Árbitro Distribuído

Dá-se o nome de árbitro distribuído quando o *hardware* usado para passar o controle de uma unidade para outra estiver altamente distribuído pelas unidades ligadas ao barramento. As diferentes formas do mecanismo distribuído para a arbitragem do acesso ao barramento são

- *daisy chaining*,
- *polling* e
- requisições independentes.

Árbitro Distribuído por *Daisy Chaining*

O estado do barramento é determinado por transições na linha de BUS AVAILABLE que atravessa, em cadeia, todas as unidades do barramento. Quando uma transição chega a uma unidade que está solicitando o barramento, esta impede a propagação da transição para as demais unidades na cadeia. E quando não há nenhuma unidade necessitando do barramento, um pulso se desloca pela linha BUS AVAILABLE continuamente, passando por todas as potenciais unidades-mestres.

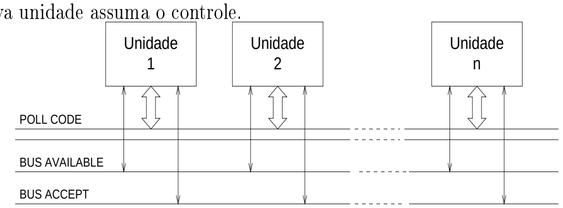


Assim como no árbitro centralizado, o método *daisy chaining* é dependente da posição da unidade na cadeia, e se uma unidade falhar, todo o barramento fica comprometido. Um problema que aparece com este método é a susceptibilidade a ruídos, por se tratar de um *hardware* sensível a transições de sinais.

Árbitro Distribuído por *Polling*

A unidade que libera o barramento é quem coloca o *poll code* nas linhas, juntamente com a ativação da linha BUS AVAILABLE. Se existe alguma unidade solicitante do barramento com código igual ao que está nas linhas, ela ocupa o barramento ativando a linha BUS ACCEPT. Em resposta, a primeira unidade-mestre retira o *poll code* das linhas e libera a linha BUS AVAILABLE. A nova unidade-mestre desativa o sinal BUS ACCEPT e assume o controle do barramento. Se não há nenhuma unidade com o mesmo código nas linhas solicitando o uso do barramento o sinal BUS ACCEPT não

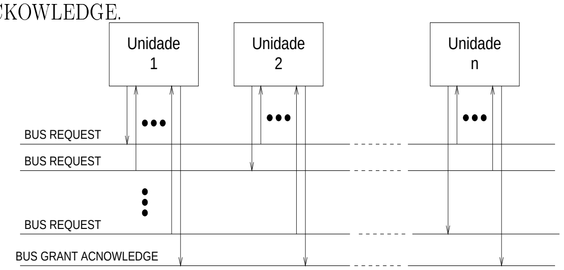
é ativado depois de esgotar um certo tempo pré-determinado. A unidade-mestre gera, então, um novo código de acordo com alguma política de alocação (prioridade ou *round robin*). E, assim sucessivamente, até que uma nova unidade assuma o controle.



Este método requer que somente uma unidade assuma o controle do barramento na iniciação do sistema. O *hardware* de alocação é o mesmo em todas as unidades e é igual àquele do árbitro centralizado por *polling*, encarecendo assim a opção por este método. Porém, a confiabilidade do sistema é maior, uma vez que a falha numa unidade não compromete o funcionamento do sistema como todo.

Árbitro Distribuído por Requisições Independentes

Este método requer que cada unidade ligada ao barramento possua uma linha de BUS REQUEST própria. O conjunto de todas as linhas de BUS REQUEST é acessível por todas as unidades. Quando uma unidade libera o barramento, ele desativa a linha BUS GRANT ACKNOWLEDGE, todas as unidades que precisam do barramento analisam o estado das linhas de BUS REQUEST. Aquele que se reconhecer como o de mais alta prioridade, assume o controle do barramento acionando a linha de BUS GRANT ACKOWLEDGE.



A lógica de prioridade em cada unidade é mais simples que no mesmo método centralizado, entretanto o número de ligações e conectores é maior. A diferença de fase no relógio entre as diversas unidades (*clock skew*) limita este método a sistemas pequenos. Além disso, este método é muito susceptível a ruídos, devido ao grande número de linhas.

Exercícios de Revisão

1. Resolver os exercícios 12, 13, 14, 15, 17, 20 e 21 do capítulo 10 do livro-texto.
2. Por que a configuração de saída OC (*open collector*) evita a contenção em uma linha?
3. Quando se aplica a teoria de linha de transmissão na análise do comportamento de um barramento?
4. O que significa eletricamente retirar e forçar um sinal numa linha do barramento?
5. Qual é a diferença entre arbitragem centralizada e arbitragem distribuída?
6. Em que consiste arbitragem por *daisy chaining*?
7. Em que consiste arbitragem por *polling*?
8. Em que consiste arbitragem por requisições independentes?
9. Pode ocorrer contenção na linha BUS GRANT ACKNOWLEDGE? Justifique.

6.4 Características Funcionaise Procedurais

Um barramento pode ser classificado de acordo com o tipo de informação que pode ser transportado por ele:

- Barramento de dados: Cada fio transfere um bit de informação em cada instante de tempo. A quantidade de fios define a largura do barramento. P. ex.: barramento de 8 bits, 16 bits e 32 bits.

- Barramento de endereços: Cada fio transfere um bit do endereço de dados mapeado no espaço de memória da unidade processadora.
- Barramento de controle: cada fio transfere um sinal de controle da transferência de informação entre as unidades.
- Barramento Multiplexado: os sinais no barramento possuem significados distintos em função de tempo.

A quantidade de fios no barramento de controle está intimamente relacionada com o modo de transferência de dados e os protocolos de comunicação suportado. Quanto ao modo de transferência é comum classificar um barramento em

- barramento síncrono ou
- barramento assíncrono.

6.4.1 Barramento Síncrono

Os sinais (de dados, de endereços e de controle) são sequenciados em função de um relógio central, que define janelas de tempo (*time slots*) do mesmo tamanho para cada operação no barramento. Existem duas técnicas básicas para a alocação do intervalo de tempo para as operações, a saber **intervalo dedicado** e **intervalo não-dedicado**.

Por intervalo dedicado entende-se que cada janela de tempo é alocada permanentemente a uma operação, mesmo que esta não seja realizada. De acordo com esta técnica, as janelas de tempo são definidas de forma a encobrir a operação mais lenta. Isto compromete o desempenho do sistema, pois as operações mais rápidas não poderão trabalhar a plena velocidade. Além disso, se não forem necessárias as operações que tem janelas de tempo reservadas, o sistema ficará ocioso prejudicando o desempenho global do sistema.

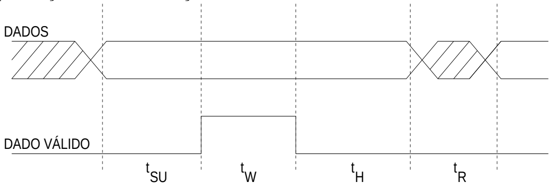
Nas técnicas de intervalo não-dedicado os intervalos de tempo são alocados a uma operação somente se ela for realizada. Isto implica na necessidade de se estabelecer um mecanismo de alocação de intervalos em *hardware* para identificar se as operações vão ocorrer. No caso de 68000 os sinais VPA*/VMA* tem esta função.

6.4.2 Barramento Assíncrono

Neste barramento o instante da transferência não é conhecido a priori. Ele é indicado por sinais de controle próprios. O significado de cada um destes sinais e a relação entre eles definem o **protocolo** de comunicação. O número dos sinais num barramento para este controle define o tipo de comunicação assíncrona, podendo ser comunicação assíncrona controlada por um fio (OWC – *one way controlled*) ou por dois fios (*Req/Ack*).

Controlado por um Fio

É a forma mais simples para transferência assíncrona dos dados entre duas unidades. Existem duas maneiras de se controlar esta transferência: iniciada pela fonte de dados ou pelo receptor de dados. Para que esta forma de comunicação funcione é necessário que haja uma rigorosa definição de temporização da comunicação entre ambas as unidades.



No primeiro caso a presença de dados é indicado pela fonte através do sinal DADO VÁLIDO. Enquanto este sinal estiver em nível ativo, o dado nas linhas é válido e o receptor pode ser sensível à largura do pulso de DADO VÁLIDO t_w (*pulse width*). Se este sinal for também usado para gatilhar os dados no receptor é importante que os dados estejam presentes num intervalo de tempo t_{SU} (*setup time*) antes da ativação de DADO VÁLIDO e que os mesmos permaneçam estáveis no barramento, no mínimo, durante um intervalo de tempo t_H (*hold time*) após a desativação do DADO VÁLIDO. O nome muito utilizado para referir o sinal de controle DADO VÁLIDO neste tipo de transferência é STROBE. Por exemplo, processadores MC68000 utilizam o sinal AS* (*strobe* de endereços) para indicar dados válidos no barramento (de endereços).

As principais desvantagens desta técnica de transferência são:

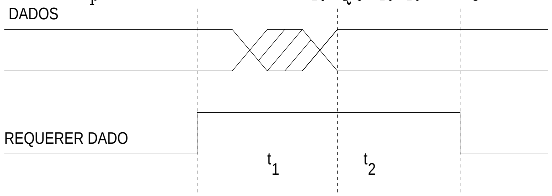
- É muito sensível a ruídos. O sinal de controle pode sofrer interferência em forma de pulso que pode ser interpretado pela outra unidade como uma indicação de transferência válida. Para minimizar este problema, t_w pode ser aumentado. Entretanto, isto compromete a taxa de transferência.

- Nada assegura à fonte que o dado que ele enviou foi recebido pelo receptor. É um tiro no escuro.

As principais vantagens desta técnica são:

- Simplicidade na implementação, requerendo poucos componentes.
- Altas taxas de transferência. Um ciclo completo de transferência requer um tempo
$$t_c \geq t_{SU} + t_w + t_H + t_R,$$
onde o valor de cada um destes tempos dependerá da tecnologia adotada. O tempo t_R corresponde ao tempo necessário para a fonte dos dados recarregar os seus registradores de saída de dados.

Quando a transferência é iniciada pelo receptor de dados, o receptor deve gerar o sinal REQUERER DADO. A fonte de dados, ao receber este sinal, coloca os dados no barramento após um intervalo de tempo igual a t_1. Neste intervalo estão considerados o tempo de propagação pela linha o sinal REQUERER DADOS e o tempo necessário à fonte de dados preparar a transferência. O receptor, então, define o instante no qual ele captura os dados do barramento. O tempo t_2, deve considerar o tempo necessário para que os dados possam propagar entre a fonte e o receptor e o tempo de *setup* da unidade receptora. O dado fica no barramento até que uma nova requisição de dados seja solicitada pelo receptor, iniciando assim uma nova comunicação. Note que este tempo não pode ser menor do que o tempo de *hold* da unidade receptora. No ciclo de leitura, o sinal WE das unidades de memória corresponde ao sinal de controle REQUERER DADO.



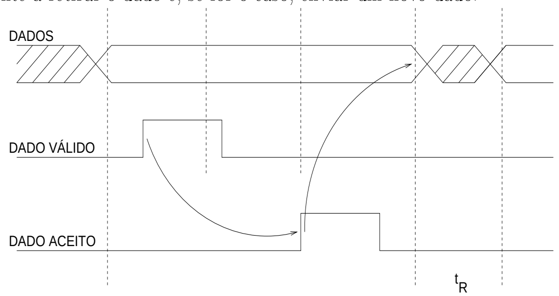
As principais desvantagens são, como na técnica anterior, a susceptibilidade a ruídos e a incerteza da existência de dados no barramento. Nada garante que, após ter sido sinalizado uma requisição de dados, após t_1 os dados estarão no barramento e o que será capturado depois de t_2 é um dado válido. Uma solução é restringir a largura do pulso do sinal REQUERER DADOS a um valor mínimo, comprometendo assim a taxa de transferência. Se a conexão entre as unidades for fisicamente assegurada, esta técnica pode ser utilizada sem incerteza da existência dos dados.

Outra desvantagem é com respeito a taxa de transferência que é normalmente menor que a técnica de controle pela fonte de dados.

Controlado por dois Fios

Quando não existir nenhum interconhecimento da temporização das unidades envolvidas numa transferência, além do sinal gerado pela fonte indicando a presença dos dados, deve ser usado um sinal a mais, vindo do receptor, que indique a recepção. Esta técnica é chamada de comunicação controlada por dois fios. Através destes dois fios é estabelecido um protocolo de comunicação denominado *handshaking*. Como em muitos equipamentos estes dois fios tem os nomes em inglês, *Request* para o sinal gerado pela fonte e *Acknowledge* para o sinal gerado pelo receptor, a transferência é também conhecida por REQ/ACK.

A fonte põe um dado no barramento e ativa o sinal DADO VÁLIDO. O receptor armazena o dado e responde com o sinal DADO ACEITO. Isso leva a fonte a retirar o dado e, se for o caso, enviar um novo dado.



A introdução de mais um sinal de controle não só minimiza a incerteza da comunicação, como tamém permite a operação entre unidades de velocidades

diferentes. O custo disso é a diminuição da taxa de transferência e a necessidade de um circuito de controle mais complexo, pois deve ser considerado um maior número de atrasos. Outro problema é que passam existir dois sinais susceptíveis a ruídos. Além disso, se as duas unidades tiverem velocidades de resposta muito distintas, pode ocorrer **estrangulamento** (*deadlock*) do barramento. Uma solução é **intertravar** os dois sinais: o sinal DADO VÁLIDo é mantido ativo até o sinal DADO ACEITO ser ativado e a desativação do sinal DADO VÁLIDO resulta em desativação do sinal DADO ACEITO e enquanto o sinal DADO ACEITO não for desativado, o sinal DADO VÁLIDO não pode ser ativado novamente.

Exercícios de Revisão

1. Qual é a diferença básica entre um barramento síncrono e um barramento assíncrono?
2. Qual é a diferença básica entre um protocolo de transferência com um sinal de controle (um fio) e um protocolo com dois sinais de controle (dois fios)?
3. O que é um barramento estrangulado?
4. O que é um protocolo de *handshaking* intertravado?

6.5 Padrões de Barramentos

Diversos fabricantes de computadores utilizam os chamados barramentos padronizados. O motivo para isto é principalmente econômico. Por exemplo, se um determinado fabricante de uma undiade processadora adotar um barramento padronizado ele poderá contar com diversos fornecedores de outras placas (de memória, de interface, etc.) e com isso, ele pode manter uma linha de produção mais especializada e trabalhar visando a uma escala de mercado mais ampla.

A especificação de um barramento padrão pode incluir as características mecânicas, elétricas, do protocolo de comunicação e do mecanismo de arbitragem do acesso aos recursos do barramento.

Veremos neste curso dois padrões de barramento: VMEbus e NuBus.

Preparo para Próxima Aula

- Leia atentamente as páginas 805-844 do livro-texto e descreva sucintamente as principais caracterísitcas do VMEBus.
- Leia atentamente as páginas 844-853 do livro-texto e descreva sucintamente as principais caracterísitcas do NuBus.

6.5.1 VMEBus

VMEBus foi proposto inicialmente pela Motorola como uma versão do barramento VERSAbus usando o padrão europeu de placas (EUROCARD). O barramento VME logo foi aceito pelo mercado como aquele que permite um melhor desempenho e que possui a mais completa especificação. Este padrão foi adotado por diversos fabricantes. Entre eles: SIGNETICS, PHILLIPS, MOSTEK e Thomson-EFCIS. O nome VME significa VERSA MODULE EUROPE. Em 1982 o barramento VME foi homologado pelo IEC como sendo o padrão IEC 821 e em 1984 pelo IEEE como sendo o padrão IEEE 1014.

A especificaçào do barramento VME define um sistema de interconexão de unidades de processamento de dados, de armazenamento de dados e de controle de periféricos numa configuraçào fortemente acoplada (*closed coupled*). O sistema foi concebido com os seguintes objetivos:

- prover comunicaçao entre duas unidades no barramento sem perturbar as atividades internas de outras unidades ligadas ao barramento;
- especificar as características elétricas e mecânicas necessárias para projetar unidades que irão comunicar confiavelmente e sem ambiguidade com outras unidades ligadas ao barramento,
- especificar protocolos que definem precisamente a interação entre o barramento VME e as unidades ligadas a ele.
- prover terminologia e definições que descrevem precisamente o protocolo do sistema,

- permitir um amplo campo de opções de projeto de forma que o projetista possa otimizar custos e/ou desempenho sem afetar a compatibilidade do sistema,
- prover um sistema onde o desempenho é limitado mais pela unidade do que limitado pela interface do sistema.

Ele utiliza até dois conectores EUROCARD de 96 pinos. A placa de circuito pode ter dois tamanhos (Fig. 10.35 do livro-texto, p. 808)

- tamanho simples: com um único conector possuindo as dimensões de 100mm × 160mm;
- tamanho duplo com dois conectores com as dimensões de 233,68mm × 160mm (Fig. 10.36 do livro-texto, p. 809).

As características dos seus sinais são definidas explicitamente, o que é permitido e o que não é permitido. Os sinais são compatívies a nível TTL e a imunidade de ruído no nível alto é 0.4 V e no nível baixo, 0.2 V.

Estrutura Básica

Consiste de quatro grupos de linhas de sinais (quatro barramentos) (Tabelas 10.7 e 10.8 do livro-texto, pp. 812 e 813):

- barramento de transferência de dados (DTB - *data transfer bus*) contém as linhas de endereços, de dados e de sinais de controle associados, totalizando 76 linhas;
- barramento de arbitragem contém 14 linhas de sinais necessários para transferência do controle do DTB entre as potenciais unidades-mestres,
- barramento de interrupção contém 10 linhas de sinais que viabilizam o sistema de interrupção vetorizada com 7 níveis de prioridade.
- barramento de utilitários contém 4 linhas de sinais de propósito genérico.

O sistema VME ainda especifica um conjunto de módulos funcionais que se comunicam através das linhas de sinais destes barramentos num *backplane* (Fig. 10.37 do livro-texto, p. 810):

- requerente do DTB (*DTB requester*), capaz de solicitar o controle do barramento de transferência de dados, DTB, e fica na mesma placa de uma unidade-mestre ou processador de interrupção;
- interruptor (*interrupter*), capz de solicitar serviços de uma unidade-mestre através de uma interrupção;
- processador de interrupção (*interrupt handler*), capaz de detectar solicitações de interrupção e iniciar respostas apropriadas;
- árbitro de DTB, capaz de receber as solicitações para uso do DTB e concedê-lo a um dos solicitantes conforme algum critério pré-estabelecido;
- escravo de DTB capaz de responder uma operação de transferência de dados iniciadas por uma unidade-mestre;
- mestre de DTB, capaz de iniciar uma transferência de dados.

Transferência de Dados

DTB é bastante similar ao barramento assíncrono dos processadores da família 68000. Os tipos de transferência no barramento VME devem ser analisados em função do endereço e em função de dados. Com respeito aos endereços, VME possui 3 tipos de transferência:

- endereçamento curto, usando 16 linhas de endereços;
- endereçamento normal, usando 24 linhas de endereços, e
- endereçamento estendido, usando 32 linhas de endereços.

E com respeito aos dados, ele possui 3 tipos de transferência:

- transferência de BYTE,
- transferência de WORD, e
- transferência de LONGWORD.

Uma característica adicional do DTB são as 6 linhas de modificadores de endereço (AM0–AM5) que permite o mestre passar para os escravos informações adicionais como subespaço de endereços, tipo de ciclo e/ou identificação do mestre.

Arbitragem de Barramento

Uma vez que um sistema baseado no padrão VME pode ser configurado para possuir mais de um mestre DTB, deve existir um meio para transferir o controle do DTB entre os mestres de forma sistemática, garantindo que somente um mestre controla o DTB num dado instante. O árbitro no barramento VME é centralizado no *slot* 1 (Fig. 10.42 do livro-texto, p. 823) e o circuito deste árbitro não é especificado pelo padrão. Porém a especificação ANSI/IEEE STD 1014-1987 provê váras sugestões de cirucuitos de arbitragem que evitem problemas de metaestabilidade.

Existem quatro linhas de requisição de barramento correspondendo a diferentes níveis de requisição do barramento (BR0*–BR3*) e três opções de estabelecer prioridades de arbitragem tratadas pelo circuito de arbitragem (inserido no *slot* 1 do barramento):

- PRI (*prioritized* option), onde a prioridade entres os quatro níveis é fixa. A linha BR3* é a mais prioritária e a linha BR0* a menos prioritária.
- RRS (*round robin select*) onde a técnica de controle do acesso nos diversos níveis é circular, isto é, na sequência BR0*–BR3*–BR2*–BR1*–BR0*– ...
- SGL (*single level*), onde somente a linha BR3* é utlizada. As unidades são ligadas em *daisy-chaining* a esta linha, de forma que a unidade que estiver inserida no *slot* mais próximo do *slot* 1 tem prioridade maior.

Fig. 10.46 do livro-texto (p. 829) mostra um circuito de arbitragem que trata os quatro níveis de requisição.

No caso das opções PRI e RRS, além da prioridade tratada pelo árbitro, existe um segundo nível de prioridade quando mais de uma unidade for conectada a uma linha de requisição. Este segundo nível é por *daisy-chaining* (Fig. 10.41 do livro-texto, p. 822). Um circuito de arbitragem por *daisy-chaining* é apresentado na Fig. 10.45 do livro-texto (p. 827).

Fig. 10.44 do livro-texto (p. 826) ilustra a temporização de duas requisições em níveis diferentes (BR1* e BR2*). Note que o sinal BBUSY* e os sinais BGx* são intertravados.

Interrupção

A capacidade de tratamento de interrupções do barramento VME fornece um meio pelo qual um módulo pode requerer a interrupção da atividade do barramento (*interrupter*) e ser servido por um módulo tratador de interrupção (*interrupt handler*). Fig 10.47 do livro-texto (p. 830) ilustra as componentes de um sistema VME que suportam esta capacidade de interrupção.

As transições negativas do sinal de interrupção IACK* geradas por um processador de interrupção são detectadas pelo acionador que está localizado no *slot* 1 e propagadas através dos *slots* interligados através do IACKOUT*–IACKIN* *daisy chain*. A propagação da "mensagem de IACK* ativo", ao chegar no módulo interruptor solicitante do mesmo nível de interrupção, é interrompida. O interruptor coloca então no barramento de dados a sua "identificação" que o processador de interrupção utiliza para invocar a rotina de serviço correta (Fig. 10.48 do livro-texto, p. 831).

VME suporta dois tipos de tratamento de interrupção (Fig. 10.50 do livro-texto, p. 836):

- tratamento simples ou centralizado, onde as requisições podem ser priorizadas em até um máximo de 7 níveis (IRQ1*–IRQ7∗), e
- tratamento distribuído em até um máximo de 7 tratadores.

No segundo caso os tratadores devem requisitar o uso do barramento DTB ao árbitro do sistema (no *slot* 1) antes de atender uma solicitação para evitar a ocorrência da contenção do barramento.

Outros Sinais Utilitários

Existem ainda quatro linhas utilitárias: SYSCLK (relógio do sistema), SYSRESET* (reseta o sistema), SYSFAIL* (indica se há falha no sistema) e ACFAIL* (indica se há falha na alimentação).

6.5.2 NuBus

Diferentemente do VMEbus, NuBus é um barramento projetado para ser independente dos processadores (VMEbus foi concebido com processadores da família 68000 em mente). Ele foi originalmente concebido em MIT (1970) com a finalidade de propor um barramento de alto desempenho e baixo custo para os computadores pessoais e estações de trabalho (*workstations*). A partir

de 1983 ele passa a ser suportado por *Western Digital* e *Texas Instrument* e hoje é a marca registrada de TI – um padrão reconhecido pelo ANSI/IEEE. Entre as companhias que o adotam temos *Apple*.

Comparações com VMEbus

É um barramento síncrono, com linhas de dados e endereços multiplexados, destinado para sistemas de microprocessadores de 32 bits com até 16 dispositivos (ou interfaces); enquanto o VMEbus é um barramento assíncrono com linhas de dados e de endereços separadas. Em consequência, o custo de um NuBus é mais baixo.

Como VMEbus, NuBus suporta multiprocessamento e múltiplos mestres de barramento. Entretanto, diferentemente de VMEbus, não existe um árbitro especial, com a localização fixa no *slot* 1, para a escolha do mestre. O circuito de arbitragem de NuBus é totalmente descentralizado e todos os *slots* tem uma única identificação (por 4 bits) e são potenciais mestres do barramento.

A política de concessão de barramento no NuBus não é por *daisy-chain* como no VMEbus. Portanto, não é necessário *bypass* as conexões de *daisy-chain* nos *slots* vazios.

Outra característica do NuBus é que ele não suporta distintos tipos de transferência de dados (p.ex., ciclo normal de acesso de leitura/escrita, ciclo de processamento de entrada/saída e ciclo de interrupção). Só há um ciclo básico, conhecido como **transação de NuBus**. O ciclo de interrupção de VMEbus é, por exemplo, tratado como ciclo de escrita no NuBus com uso do mecanismo de passagem de mensagens. Em decorrência disso, a complexa estrutura de interrupção é eliminada.

Estrutura Básica

O barramento contém funções essenciais e atende necessidade de sistemas multiprocessadores. Ele consiste de 96 linhas organizadas em seis grupos (Tabela 10.13 do livro-texto, p. 849):

1. 34 linhas para endereços/dados/paridade;
2. 4 linhas para transações de dados;
3. 5 linhas para arbitragem de barramento;

4. 4 linhas para identificação de cada "slot";
5. 4 linhas para "sinais" utilitários; e
6. 45 linhas para alimentação (+5 V, -5.2V, +12 V, -12 V) e terra.

A organização de dados é de menos significativo para o mais significativo (Fig. 10.59 do livro-texto, p. 848). Os sinais do barramento são ativo-baixo.

Transações de Dados

Transação de dados (*data transaction* é um termo adotado pelo padrão NuBus para referir transferência de dados, ou um ciclo de acesso ao barramento (de leitura ou escrita).

Os sinais de controle de transação, START* e ACK*, desempenham funções similares aos sinais de controle do barramento assíncrono do VMEbus. Embora NuBus seja um barramento síncrono (cada ciclo de acesso necessita de, no mínimo, 2 pulsos de relógio, um para endereço e outro para dados), os estados de espera podem ser introduzidos para atrasar a resposta ACK* dos escravos de barramento (Fig. 10.60 do livro-texto, p. 850). A natureza de cada ciclo é definida pelos códigos nas linhas TM0* e TM1*. Tabela 10.14 do livro-texto (p. 851) sintetiza as diferentes possíveis mensagens suportadas por NuBus. A correspondência entre os sinais de NuBus e os de VMEbus para especificar o tamanho de dados transferidos é dada na Tabela 10.15 do livro-texto, p. 852.

Além das 32 linhas para endereços e dados multiplexados, NuBus provê ainda duas linhas para validação de dados: SPV* e SP*. SP* indica a paridade no barramento de endereços/dados e SPV* indica se o bit de paridade é válido ou não.

Arbitragem de Barramento

O controle de arbitragem de barramento é feito através dos sinais ARB0*–ARB3* e RQST*. O potencial mestre de barramento solicita o uso do barramento através da ativação contínua do sinal RQST* até conseguir o seu controle. A política de arbitragem adotada por NuBus é distribuída por *polling*. Portanto, a identificação de cada *slot*, ID0*–ID3*, é fundamental no processo de arbitragem.

Por meio de acionadores com configuração de saída do tipo coletor aberto, cada requerente (*slot*) pode "colocar" a sua identificação, ID0*–ID3*, nas linhas de arbitragem ARB0*–ARB3* e "ler" o nível corrente nestas linhas. Se o requerente detectar um nível de prioridade maior do que o seu, ele retira o seu sinal de requisição. Assim, somente o requerente de maior prioridade permanece acionando as linhas de arbitragem e ele será o escolhido para ser o mestre do barramento (Fig. 10.61 do livro-texto, p. 853).

NuBus apresenta uma estratégia adequada (*fair*) para atender as unidades conectadas a ele. Um *slot*, depois de ter a sua solicitação atendida, só pode solicitar novamente o barramento depois que todas as pendentes forem tratadas. Somente no ciclo conhecido como o ciclo de atenção (*attention cycle*), um mestre de barramento pode manter a posse de um barramento por um tempo maior. Neste caso, os sinais START* e ACK* são mantidos ativos simultaneamente.

Interrupções

NuBus suporta duas formas para um dispositivo "interromper" um processador:

1. através de um ciclo de escrita, escrevendo a mensagem na região de memória monitorada pelo processador (mecanismo de passagem de mensagem); ou
2. através da linha "nonmaster request", NMRQ*, que opera como a linha de interrupação convencional. Esta linha foi inclusa para suportar dispositivos que provêm mecanismos de passagem de mensagem.

Outros Sinais Utilitários

Existem ainda duas linhas utilitárias, RESET* e PFW*. O sinal RESET* é utilizado para inicializar todos os dispositivos do barramento. Ele também pode ser usado em conjunto com PFW* para tratar apropriadamente em casos de falha na alimentação.

Exercícios de Revisão

1. Resolver os exercícios 24 – 31 e 34 do capítulo 10 do livro-texto.

6.6 Auto-avaliação

Após este capítulo, você deve ser capaz de:

- explicar o papel de um barramento num sistema computacional.
- citar as principais especificações de um barramento.
- explicar os efeitos das diferentes configurações de saída dos dispositivos digitais no acionamento de uma linha de barramento.
- estimar o valor dos *pull-up* resistores numa linha acionada pla configuação de saída OC (*open collector*).
- explicar os diferentes modelos elétricos de barramento.
- explicar a importância das terminações corretas em cada linha de um barramento.
- definir a impedância caracterísitca de uma linha sem carga e com carga.
- citar os problemas mais críticos que um projetista deve considerar no projeto de um barramento que suporta "inserção ao vivo".
- citar os problemas mais críticos que um projetista deve considerar no projeto de um circuito de arbitragem.
- classificar as diferentes políticas de arbitragem de um barramento.
- classificar os barramentos segundo os sinais de controle necessários para uma transferência de dados.
- explicar a importância do uso de barramentos padronizados num projeto.
- descrever sucintamente o padrão VMEBus.
- descrever sucintamente o padrão NuBus.

6.7 Lista de Exercícios

1. Resolver o exercício 16 do capítulo 10 do livro-texto.
2. Resolver o exercício 19 do capítulo 10 do livro-texto.
3. Resolver o exercício 22 do capítulo 10 do livro-texto.
4. Resolver o exercício 32 do capítulo 10 do livro-texto.
5. Resolver o exercício 33 do capítulo 10 do livro-texto.
6. Resolver o exercício 34 do capítulo 10 do livro-texto.
7. Resolver o exercício 38 do capítulo 10 do livro-texto.